Mining Optimization Laboratory

Report One – 2008/2009

Directed by Hooman Askari-Nasab

School of Mining and Petroleum Engineering
Department of Civil & Environmental Engineering,
University of Alberta, Edmonton, Alberta, CANADA

Contact information for requests for permission to reproduce or distribute materials available through this report is:

Hooman Askari-Nasab, Ph.D., P.Eng.
Director of Mining Optimization Laboratory
Assistant Professor of Mining Engineering
3-044 Markin/CNRL NREF Building
Department of Civil & Environmental Engineering
University of Alberta, Edmonton, AB,
Canada T6G 2W2
Phone: (780) 492 4053
Fax: (780) 492-0249
Email: hooman@ualberta.ca

Executive Summary

I am pleased to announce the release of the first *Mining Optimization Laboratory (MOL)* annual research report. *MOL* is an industrial affiliates program at the School of Mining and Petroleum Engineering at the University of Alberta. The program has been launched in January 2009. I would like to thank the support of the founding sponsors, Newmont USA Limited and Suncor Energy Inc. in sponsoring this initiative.

It is anticipated that the value generated to sponsors by the outcome of research would create more industrial awareness and excitement to support the research group in long-run with the focus on two major themes (i) risk-based mine planning and design and (ii) simulation optimization of mining systems. The mission of *MOL* research group is to (i) train highly qualified personnel, and (ii) to extend the industry best practice guidelines and standards in the above-mentioned areas. I am sure this report will demonstrate the value generated to our sponsors through research and will give a flavor of the quality of research and deliverables to the potential sponsors.

As promised, research and training has been conducted and documented in two main categories, mine planning and design and simulation optimization of mining systems. This report is a deliverable that would rationalize sponsorship of *MOL* research. The students at the *MOL* are just starting their program and they are going through a training phase by taking courses and working on mine planning projects. However, I am satisfied with the research results documented in this report and foresee a successful research group with focus on far-sighted fundamental research addressing industry applicative needs. This report presents a number of interesting contributions towards developing the major components of a risk-based mine planning framework. Let's review the contributions in Report One by considering some of the main contributors.

Hooman has worked on large-scale long-term open pit production scheduling using mixed integer linear programming (MILP) techniques (papers 101, 102, and 104). The MILP models have the capability of improving the net present value of the production schedules significantly comparing to the traditional methods. An optimal schedule is generated with all technical constraints such as: mining-fleet capacity, shovel limits, processing options, multiple processes, multiple elements, and blending and head-grade constraints for ore and deleterious materials. One of the most important features of the MILP techniques is availability of a measure goodness of the solution compared against the theoretical optimum.

We investigated the shortcomings of the MILP models used for open pit production scheduling, particularly the inability to solve large-scale real-size mining problems. To reduce the size of the open pit production scheduling problem we introduced the concept of mining-cuts into the MILP formulations. Blocks within the same level or mining bench are grouped into clusters based on their attributes, spatial location, rock type, and grade distribution. In this study, we have developed, implemented, and tested four MILP formulations for large-scale open pit production scheduling. The models were implemented in MATLAB and TOMLAB/CPLEX environment. The developed models proved to be able to handle deterministic large-scale mine production problems. A case study was conducted on an iron ore mine with more than 170 thousands blocks within the final pit outline over twenty planning periods. The case studies have proved how effective MILP formulations are in generating high value practical schedules and maximizing the recovery through consistent feed grade to the plant.

Another ongoing research direction followed by ***Hooman*** is mine planning using intelligent open pit production simulator, IOPS (papers 105 and 106). The intelligent agent theoretical framework for real size mine planning was developed as his PhD work based on reinforcement learning algorithms. The intelligent agent framework employs Q-learning algorithm to maximize the net present value of the mining operation and is a powerful tool in generating feasible optimal push-backs. IOPS, is developed and implemented in Java and MATLAB and tested through a mining case study against Whittle software.

Behrang has carried out a very interesting research work on quantifying the uncertainty transferred from geological modeling into mine planning (paper 103). First, sequential Gaussian simulation is used to generate fifty realizations of an oil sands deposit. An optimum final pit limits design is carried out for each SGS realization while fixing all other technical and economic input parameters. Afterwards, the long-term schedule of each final pit shell is generated. Uncertainty in the final pit outline, net present value, production targets, and the head grade are assessed and presented and compared against the kriged and post simulation models. The results show that there is significant uncertainty in the long-term production schedules, even though a buffer stockpile was considered in the study.

Kwame is an assistant professor of mining engineering at the Missouri University of Science and Technology, Rolla, USA. His current research interests include modelling and optimization, and life-cycle assessment of mining systems. He remains an integral part of our research group particularly in mathematical programming and simulation modeling. Kwame presented work on cable shovel kinematics and dynamic modeling combined with formation resistance modeling (paper 203) and a linear programming model of the asphalt mix design problem along with a numerical algorithm to solve the model (paper 205).

Eugene, ***Samira***, and ***Yashar*** have taken very important training steps toward their degrees with taking courses in simulation of industrial and engineering systems, and engineering optimization. Mixed integer programming formulations occasionally create a scattered block extraction order that cannot be implemented in practice. ***Yashar*** has made meaningful contributions in examining different techniques to improve the practical feasibility of scheduling patterns. ***Yashar*** has also documented the steps required to generate a short-term schedule on a mining bench using MIP models with Excel Solver (paper 110). ***Eugene*** and ***Samira*** joined us in January 2009; they have made meaningful contributions in pit optimization and blending considering the short time that they have been with *MOL* (papers 107 and 108). Another training/research activity undertaken by ***Eugene*** and ***Samira,*** was orebody modeling, optimization and pit design of an oil sands deposit in the McMurray formation (papers 111 and 112). They will continue working on geostatistical modeling of oil sands deposit as an integral part of their theses. ***Mohammad***, ***Hesam***, and ***Mahdi*** have joined us as PhD students in September 2009 with industrial engineering background. Thanks for joining us!

September 2009

Hooman Askari

First Annual Research Report of the Mining Optimization Laboratory (MOL) 2008/2009

Table of Contents

Mining Optimization Laboratory (MOL) Researchers / Graduate Students

Following are researchers and students affiliated with Mining Optimization Laboratory in September 2009.

1. **Hooman Askari-Nasab**	Assistant Professor and Director of MOL
2. **Kwame Awuah-Offei**	Assistant Professor of Mining Engineering

---

3. **Mahdi Badiozamani**	PhD Student (new) - 2009/09
4. **Eugene Ben-Awuah**	PhD Student - 2009/01
5. **Hesameddin Eivazy**	PhD Student (new) - 2009/09
6. **Yashar Pourraimian**	PhD Student - 2008/09
7. **Samira Kalantari**	Msc Student - 2009/01
8. **Behrang Koushavand**	PhD Student (CCG) - 2007/09
9. **Mohammad Tabesh**	PhD Student – (new) 2009/09

Mixed integer linear programming formulations for open pit production scheduling

Hooman Askari-Nasab and Kwame Awuah-Offei[1]
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

We have proposed two mixed integer linear programming (MILP) formulations for large-scale long-term open pit production scheduling problem. We developed, implemented, and tested the proposed MILP theoretical frameworks for large-scale open pit production scheduling.

1. Introduction

Historical assessment of mineral project performances has demonstrated the sensitivity of projects' profitability to decisions based upon mine production schedules. The life-of-mine production schedule defines the complex strategy of displacement of ore, waste, overburden, and tailings over the mine life. The objectives of long-term production schedules are to determine the sequence of extraction and displacement of material in order to maximize the future cash flows of mining operations within the existing economic, technical, and environmental constraints. Long-term production schedules lead to definition of reserves and are the backbone of short-term planning and day to day mining operations. The long-term production schedules resolve mine and processing plant capacity and their expansion potential; the production schedule, also defines the management investment strategy. Deviations from optimal plans in mega mining projects will result in enormous financial losses, delayed reclamation, and resource sterilization.

In this study, we have proposed four mixed integer linear programming (MILP) formulations for large-scale long-term open pit production scheduling problem. We developed, implemented, and tested the proposed MILP theoretical frameworks for large-scale open pit production scheduling.

Current production scheduling methods in the literature are not just limited to, but can be divided into three main categories: heuristic methods, applications of artificial intelligence techniques, and operations research methods. Some of these algorithms are embedded into available commercial software packages.

[1] Assistant Professor, Department of Mining & Nuclear Engineering, University of Missouri-Rolla, USA,

One of the heuristic methods used in mine production scheduling was proposed by Gershon (1987). XPAC AutoScheduler (Runge Limited, 1996-2009), a commercial mine scheduling software is developed based on Gershon's (1987) proposed heuristic. Gershon's (1987) algorithm generates cones upward from each block to approximate the shape of a pit and to determine whether or not the block in question could be part of the schedule. A list of exposed blocks and a ranking of those blocks based on what makes it more desirable or less desirable to mine an exposed block at the present time is updated through the algorithm with an index called the positional weight. This weighted function is used to determine the removal sequence.

Another popular heuristic used in strategic mine planning software, such as Whittle (Gemcom Software International, 1998-2008) and NPV Scheduler (Datamine Corporate Limited, 2008) is based on the concept of parametric analysis introduced by Lerchs and Grossmann (1965) (LG). The LG algorithm provides an optimal solution to the final pit outline. There are unlimited numbers of strategies of reaching the final pit, which each has a different discounted cash-flow. The optimal production schedule is the strategy that would maximize the discounted cash-flow and meets all the physical and economical constraints. The parametric analysis generates a series of nested pits based on varying the price of the product (revenue factor) and finding an optimal pit layout using LG algorithm. These nested pits then are used as a guideline to identify clusters of high grade ore and to determine the production schedule. The main disadvantage of heuristic algorithms is that the solution may be far from optimal and in mega mining projects, this is equal to huge financial losses.

Various models based on a combination of artificial intelligence techniques have been developed (Denby and Schofield, 1994; Denby et al., 1996; Tolwinski and Underwood, 1996; Askari-Nasab, 2006; Askari-Nasab et al., 2008; Askari-Nasab and Awuah-Offei, 2009). Tolwinski and Underwood (1996) used a method which combines concepts from dynamic programming, stochastic optimisation, and artificial intelligence with heuristic rules to obtain ultimate pit limit and production planning concurrently. The method works by modelling the development of the mine as a sequence of pits where each pit differs from the previous pit by the removal of blocks. A probability distribution based on the frequency with which particular states occur is used to determine the state changes. Heuristic rules are incorporated to learn these characteristics of the sequence of pits which produce a good, or poor, result. Denby et al. (1996) employed genetic algorithms and simulated annealing by generation of random pit population and assessment of a fitness function to acquire the production schedule and final pit, concurrently. The advantages of their method are flexibility and solution for the ultimate pit limit and production schedule at the same time. The major drawback is that the results are not reproducible and there was no measure of the optimality of the solution.

In a series of publications the authors developed and tested the intelligent agent-based theoretical framework for open pit mine planning (Askari-Nasab et al., 2005; Askari-Nasab, 2006; Askari-Nasab et al., 2007; Askari-Nasab and Szymanski, 2007; Askari-Nasab et al., 2008) comprising algorithms based on reinforcement learning (Sutton and Barto, 1998) and stochastic simulation. This intelligent open pit simulator (IOPS) (Askari-Nasab, 2006) has a component that simulates practical mining push-backs over the mine life. An intelligent agent interacts with the push-back simulator to learn the optimal push-back schedule using reinforcement learning. The intelligent agent-based mine planning simulator, IOPS, was successfully used to determine the optimal push-back schedule of an open pit mine with a geological block model containing 883 200 blocks (Askari-Nasab and Awuah-Offei, 2009). A number of the artificial intelligence techniques, such as IOPS are based on frameworks that theoretically will converge to the optimal solution, given sufficient number of simulation iterations. The main disadvantage however, is that there is no quality measure to solutions provided comparing against the theoretical optimum.

A variety of operations research approaches including linear programming (LP) and mixed integer linear programming (MILP) have been applied to the mine production scheduling problem. The pioneer work of Johnson (1969) used an LP model, which led to the MIP formulations by Gershon (1983) for the production scheduling problem. Mixed integer linear programming mathematical optimization models have the capability to consider multiple ore processors and multiple elements during optimization. This flexibility of mathematical programming models result in production schedules generating significantly higher net present value (NPV) than those generated by the other traditional methods. Every orebody is different, but for a typical open pit long-term scheduling problem, the number of blocks is in the order of a couple of hundred thousands to millions and the number of scheduling periods are twenty and more for a life-of-mine yearly schedule. Evidently, the number of integer and linear decision variables, and the number of constraints formulating a problem of this size would easily exceed the capacity of current state of hardware and commercial mathematical optimization solvers.

Various models based on mixed integer linear programming mathematical optimisation have been used to solve the long-term open-pit scheduling problem (Caccetta and Hill, 2003; Ramazan and Dimitrakopoulos, 2004; Dagdelen and Kawahata, 2007; Boland et al., 2009). The applications of MILP models result in production schedules generating near theoretical optimal net present values. In practice, formulating a real size mine production planning problem by including all the blocks as integer variables will simply exceed the capacity of the current commercial mathematical optimisation solvers. Various methods of aggregation have been used to reduce the number of integer variables that are required to formulate the mine planning problem with MILP techniques. Ramazan and Dimitrakopoulos (2004) illustrated a method to reduce the number of binary integer variables by setting waste blocks as continuous variables instead of integer variables. Ramazan and Dimitrakopoulos (2004) reported a case study on a small single level nickel laterite block model with 2030 blocks over three periods.

Ramazan et al. (2005; 2007) presented an aggregation method based on fundamental tree concepts to reduce the number of decision variables in the MILP formulation. The fundamental tree algorithm has been used in a case study with 38 457 blocks within the final pit limits, Whittle strategic mine planning software (Gemcom Software International, 1998-2008) has been used to break-down the overall problem into four push-backs. Subsequently, the blocks within the push-backs were aggregated into 5512 fundamental trees and scheduled over eight periods using the formulation presented in Ramazan and Dimitrakopoulos (2004). Information about the run-time of the MILP models are not presented in Ramazan (2007); also the break-down of the problem into four push-backs based on the nested pit approach and formulating them as a separate MILP would not generate a global optimum solution to the overall problem. On the other hand the size of the problem of around thirty thousand blocks over eight periods is more a mid-range planning problem rather than a long-term life of mine schedule.

Caccetta and Hill (2003) presented a formulation that used binary integer variables, they developed and implemented a personalized branch-and-cut (Horst and Hoang, 1996) method in C++ using CPLEX (Bixby, 1987-2009) to solve the relaxed LP sub-problems. Boland et al. (2009) have demonstrated an iterative disaggregation approach to using a finer spatial resolution for processing decisions to be made based on the small blocks, while allowing the order of extraction decisions be made at an aggregate level. Boland et al. (2009) reported notable improvements on the convergence time of their algorithm for a model with 96 821 blocks and 125 aggregates over 25 periods. However, combining 96 821 blocks into only 125 aggregates would reduce the freedom of decision variables and the schedule generated could not be considered as an optimal solution in comparison to the case that 96 821 blocks had complete freedom. Moreover, in Boland et al. (2009) there is no representation of the generated schedules in terms of annual ore and waste production, average grade of ore processed, and cross sections and plan views of the schedules to assess the practicality of the solutions from mining operational point of view. Boland et al. (2009) also did not represent enough information on their method of aggregation, they assumed that an aggregation method similar to Ramazan (2007) would be used.

MineMax (Minemax Pty Ltd, 1998-2009) is a commercially available strategic mine scheduling software, which uses an MILP formulation solved by ILOG CPLEX (Bixby, 1987-2009) solver. Given that, MineMax is a commercial software we couldn’t find detailed information about the approach and formulation, but our understanding from the evaluation of the demo tutorial version of MineMax is that as a general strategy it is suggested to initially breakdown the final pit into nested pit shells based on parametric analysis concepts represented by Lerchs and Grossmann (1965). The pit shells define a pit to pit precedence constrained by the minimum and maximum number of benches by which the mining of one specified pit shell is to lag behind the previous one. The other option to define rules for precedence of extraction is either by proportions mined on each bench or by block precedence based on the overall pit slopes. Next each pit shell is formulated as a separate MILP model which can contribute to the overall quantity of mining and processing targets within the grade and precedence constraints; this approach result in MILP formulations for each pit shell with smaller size which will converge faster, but it could not be considered a global optimization of the problem since the pit shells are defined by the parametric analysis first. Another optimization strategy is using sliding windows which are sub-problems that are tackled on a period by period basis.

Blasor (Stone et al., 2007) and Prober (Whittle, 2007) are other proprietary software which tackle the strategic mine production scheduling by an MILP. Current MILP formulations used for open pit production scheduling fail because of: (1) inability to solve large-scale real-size mining problems as global optimization problem, and (2) inability to quantify the geological uncertainty inherent within the problem and, as a result, the associated risk with the mine plans. To overcome these problems the first step is to develop, implement, and test a theoretical framework that is capable of handling real-size mining problems in a global optimization framework. The development of an MILP model that can handle a deterministic large-scale mine production schedule will fulfill this objective.

We have critically reviewed the MILP formulations of the open pit production scheduling problem. We have implemented, and tested two of these MILP formulations. The shortcomings and deficiencies of the formulations were documented. We have proposed two MILP formulations for the long-term open pit production scheduling problem to overcome the shortcomings of the reviewed methods. We have divided the major decision variables into two categories, continuous variables representing the portion of a block that is going to be extracted in each period and binary integer variables controlling the order of extraction of blocks or the precedence of mining-cuts through a dependency directed graph using depth-first-search algorithm. The depth-first-search algorithm component, added a very important practical mining feature to the model. This model allows variable pit slopes to be integrated in the MILP formulation. As a practical constraint our MILP formulation also ensures that the fractional extraction of blocks are not going to be split over more than three periods. We have implemented the optimization formulation in TOMLAB/CPLEX (Holmström, 1989-2009) environment. An iron ore mine intermediate scheduling case study over twelve periods was carried out to verify and validate the models and to illustrate the effectiveness of the generated schedules from mining point of view. The results proved that the models are accurate and efficient and showed enhanced CPU time performance comparing to the reviewed models.

The next section of the paper covers the assumptions, problem definition, and the notations of variables. Section 3 presents four mixed integer linear programming formulations of the problem, while Section 4 presents the numerical modelling. The next section represents the numerical experiments used for verification and validation of the models. Finally, Section 6 presents the conclusions and future work followed by the list of references.

2. Assumptions, problem definition, and notation

We assume that the orebody is represented by a geological block model, which is a three-dimensional array of rectangular or cubical blocks used to model orebodies and other sub-surface structures. Numerical data are used to represent a single attribute of the orebody such as: rock types, densities, grades, elevations, or economic data. The size of the block that is needed for outlining the orebody depends on the shape and size of the ore body, mining bench height, mining method, and mining equipment. Geostatistics provides accurate and reliable estimations or simulation of block attributes at locations where no measurements are available. The most common estimation method used is Kriging (Krige, 1951). However, Kriging results do not capture uncertainty and may be systematically biased. Uncertainty always exists in presence of sparse geological data. Conditional simulation algorithms such as Sequential Gaussian Simulation (Isaaks, 1990) are geostatistical methods used to assess geological uncertainty. The generated realizations are equally probable and represent plausible geological outcomes. We will tackle the deterministic problem in this study so Kriging is the estimation method of choice.

We assume that each block is subdivided into smaller regions identified as parcels. A parcel is part of a block for which the rock-type, tonnage and element content are estimated or simulated by Geostatistical methods. A block may contain zero or more parcels. The total tonnage of the parcels may be the same as the tonnage of the block, or it may be less. If it is less, the difference is called undefined waste, which is waste of unknown rock-type. If a block has no parcels, the total tonnage of the block is undefined waste. Neither the position of a parcel within a block, nor its shape, are defined (Gemcom Software International, 1998-2008). The spatial location of each block is defined by the coordinates of its center; attributes such as rock-type, quantity of valuable and contaminant elements, and block tonnage are estimated and grade of each element of interest is calculated based on the quantity of the attributes and the ore tonnage in that block.

We consider that the geological block model is going to be extracted using open pit mining techniques. We assume that a classical optimum design of final pit limits is carried out; either based on graph theory (Lerchs and Grossmann, 1965; Zhao and Kim, 1992) or network flow algorithm (Johnson and Barnes, 1988; Yegulalp and Arias, 1992). This pit outline represents reserves that would maximize the profit. We have illustrated in Askari-Nasab and Awuah-Offei (2009) that a final pit outline obtained directly by using an optimal long-term scheduling algorithm will result in a pit outline that is a subset of the conventional final pit outline generated by the Lerchs and Grossmann's algorithm (1965). Consequently, we will follow the classical process of open pit long-term scheduling of first, finding the final pit limits and then generating a production schedule within the final pit outline.

The basic problem, in its simplest form, is finding a sequence in which ore and waste blocks should be removed from the predefined open pit outline and their respective destinations, over the life of mine, so that the net present value of the operation is maximized. The production schedule is subject to a variety of physical, technical and economic constraints. The constraints enforce the mining extraction sequence, overall pit slopes, mining, milling, and refining capacities, blending material to meet head grade requirements, minimum mining width, and the number of active mining benches in each production period. The problem presented here involves scheduling of N different ore and waste blocks within the final pit outline over T different periods of extraction.

Blocks within the same level or mining bench are grouped into clusters based on their attributes, spatial location, rock type, and grade distribution. We refer to these clusters of blocks as mining-cuts. Similar to blocks, each mining-cut has coordinates representing the center of the cut and its spatial location. Fig. 1 illustrates a schematic plan view of a mining bench. Blocks are aggregated into mining-cuts.

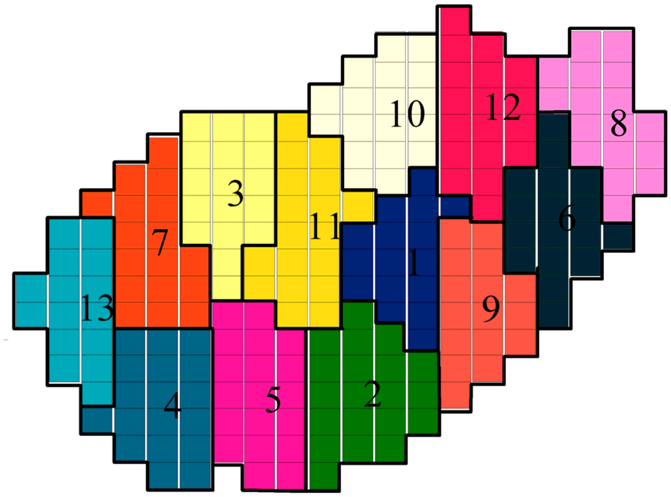


Fig. 1. Schematic plan view of aggregated blocks into mining-cuts on a mining bench.

This grade aggregation methodology will summarize ore data, but will maintain a relevant separation of lithology. Instead of the common approach of estimating grades for the element of interest in blocks, the total amount of the element in the block would be modeled. This tactic will allow to combine the smaller blocks into mining-cuts without sacrificing the accuracy of the estimated values and to model a more realistic equipment movement strategy. The mining-cut clustering algorithm developed uses fuzzy logic clustering (Kaufman and Rousseeuw, 1990) and is not within the scope of this paper and we will disseminate the clustering approach in another publication.

2.1. Notation

We will present four different MILP formulations for the open pit production scheduling problem. The notation of decision variables, parameters, sets, and constraints are as follows:

2.1.1 Sets

$\mathcal{N} = \{1,...,N\}$ set of all the blocks in the model.

$\mathcal{K} = \{1,...,K\}$ set of all the mining-cuts in the model.

$\mathcal{A} = \{1,...,A\}$ set of all directed arcs in the blocks' precedence directed graph denoted by $G_b(\mathcal{N},\mathcal{A})$.

$\mathcal{B} = \{1,...,B\}$ set of all edges in the mining-cuts precedence directed graph denoted by $G_c(\mathcal{K},\mathcal{B})$.

$D(J)$ for each block, n, there is a set $D(J) \subset \mathcal{N}$ which includes all the blocks that must be extracted prior to mining block n to ensure that block n is exposed for mining with safe pit slopes, where J is the total number of blocks in set $D(J)$.

$C(L)$ for each block, n, there is a set $C(L) \subset D(J)$ defining the immediate predecessor blocks that must be extracted prior to extraction of block n, where L is the total number of blocks in set $C(L)$.

$H(S)$ for each mining-cut c_k, there is a set $H(S) \subset \mathcal{K}$ defining the immediate predecessor cuts that must be extracted prior to extracting mining-cut k, where S is the total number of cuts in set $H(S)$.

2.1.2 Indices

A general parameter f can take four indices in the format of $f_{k,n}^{e,t}$. Where:

$t \in \{1,....,T\}$ index for scheduling periods.

$k \in \{1,...,K\}$ index for mining-cuts.

$n \in \{1,...,N\}$ index for blocks.

$e \in \{1,...,E\}$ index for elements of interest in each block.

2.1.3 Parameters

d_n^t d_n^t is the discounted profit generated by extracting block n in period t.

v_n^t the discounted revenue generated by selling the final product within block n in period t minus the extra discounted cost of mining all the material in block n as ore and processing it.

q_n^t the discounted cost of mining all the material in block n as waste.

c_k mining-cut k.

g_n^e, g_k^e average grade of element e in ore portion of block n and average grade of element e in ore portion of mining-cut k.

$gu^{e,t}$ upper bound on acceptable average head grade of element e in period
t.

$gl^{e,t}$ lower bound on acceptable average head grade of element e in period t.

o_n, o_k ore tonnage in block n and ore tonnage in mining-cut k.

w_n, w_k waste tonnage in block n and waste tonnage in mining-cut k.

pu^t upper bound on processing capacity of ore in period t (tonnes).

pl^t lower bound on processing capacity of ore in period t (tonnes).

mu^t upper bound on mining capacity in period t (tonnes).

ml^t lower bound on mining capacity in period t (tonnes).

$r^{e,t}$ processing recovery, is the proportion of element e recovered in time period t.

$p^{e,t}$ price in present value terms obtainable per unit of product (element e).

$cs^{e,t}$ selling cost in present value terms per unit of product (element e).

$cp^{e,t}$ extra cost in present value terms per tonne of ore for mining and processing.

cm^t cost in present value terms of mining a tonne of waste in period t.

2.1.4 Decision Variables

Model 01

$z_n^t \in \{0,1\}$ binary integer variable, equal to 1 if block n is to be mined in period t, otherwise 0.

Model 02

$u_n^t \in [0,1]$ continuous variable, representing the portion of block n to be extracted and processed in period t or mined and treated as waste in period t.

$a_n^t \in \{0,1\}$ binary integer variable controlling the precedence of extraction of blocks. a_n^t is equal to one if extraction of block n has started by or in period t, otherwise it is zero.

Model 03

$x_n^t \in [0,1]$ continuous variable, representing the portion of block n to be extracted as ore and processed in period t.

$y_k^t \in [0,1]$ continuous variable, representing the portion of mining-cut c_k to be mined in period t¸ fraction of y characterizes both ore and waste included in the mining-cut.

$b_k^t \in \{0,1\}$ binary integer variable controlling the precedence of extraction of mining- cuts. b_k^t is equal to one if extraction of mining-cut c_k has started by or in period t, otherwise it is zero.

Model 04

$s_k^t \in [0,1]$ continuous variable, representing the portion of mining-cut c_k to be extracted as ore and processed in period t.

Decision variables y_k^t and b_k^t with the same definition as in Model 03 are used in Model 04 as well.

2.2. Economic block value modeling

The objective functions of the MILP formulations are to maximize the net present value of the mining operation. Hence, we need to define a clear concept of economic block value based on ore parcels which could be mined selectively. The profit from mining a block depends on the value of the block and the costs incurred in mining and processing. The cost of mining a block is a function of its spatial location, which characterizes how deep the block is located relative to the surface and how far it is relative to its final dump. The spatial factor can be applied as a mining cost adjustment factor for each block according to its location to the surface. The discounted profit from block n is equal to the discounted revenue generated by selling the final product contained in block n minus all the discounted costs involved in extracting block n, this is presented by Eqs. (1) and (2).

$$\text{discounted profit = discounted revenue - discounted costs} \tag{1}$$

$$d_n^t = \underbrace{[\sum_{e=1}^{E} o_n \times g_n^e \times r^{e,t} \times (p^{e,t} - cs^{e,t})}_{discounted\ revenues} - \underbrace{\sum_{e=1}^{E} o_n \times cp^{e,t}] - [(o_n + w_n) \times cm^t]}_{discounted\ \cos ts} \tag{2}$$

For simplification purposes we denote:

$$v_n^t = [\sum_{e=1}^{E} o_n \times g_n^e \times r^{e,t} \times (p^{e,t} - cs^{e,t}) - \sum_{e=1}^{E} o_n \times cp^{e,t}] \tag{3}$$

$$q_n^t = (o_n + w_n) \times cm^t \tag{4}$$

3. Mixed integer linear programming models for open pit production scheduling

We present four different formulations for the open pit production scheduling problem, with the objective function to maximize the NPV of the mining operation. It is intuitively apparent that higher NPV's could be achieved by block models with small block sizes and high resolution of the orebody model. The block sizes for production scheduling must be chosen similar to the selective mining size. If the size of the block is not properly defined, the generated schedule will be simulating the mining operation with a selectivity that could not be achieved in practice. We have developed, implemented, tested, and compared four MILP formulations based on different precedence of extraction graphs and extraction and processing selectivity levels. The four MILP formulations are: (i) Model 01- this formulation is similar to Ramazan and Dimitrakopoulos (2004), it only consists of binary integer decision variables, z_n^t, and generates a schedule at block level resolution. We have extended the model by integrating a variable pit slope component into the implementation; (ii) Model 02 – we have proposed this model based on the concepts presented in Caccetta and Hill (2003). The schedule is generated based on a strict temporal sequence of blocks. Caccetta and Hill (2003) formulation only uses binary integer decision variables, which makes the size of branch and cut tree intractable for a large scale problem. The proposed formulation uses continuous variables, u_n^t to model extraction and processing at block level and binary integer decision variables, a_n^t, are used to control precedence of extraction; (iii) Model 03 – this formulation is developed based on the concepts of Boland et al. (2009), processing is controlled at block level with continuous decision variables, x_n^t; where y_k^t, controls the extraction at mining-cut level. Also, the precedence of extraction of blocks is controlled at the mining-cut level by means of binary integer variablesb_k^t. The continuous decision variables ($x_n^t$and y_k^t) lead to fractional block extraction, the proposed model provides control over the maximum number of fractions that each block would take; (iv) Model 04 – this formulation is based on a combination of concepts presented in Models 02 and 03. Extraction, processing, and order of block extraction are controlled at mining-cut level.

3.1. Model 01 – extraction and processing at block level – only binary decision variables

Objective function:

$$\max \sum_{t=1}^{T} \sum_{n=1}^{N} d_n^t \times z_n^t \tag{5}$$

Subject to:

$$gl^{t,e} \le \sum_{n=1}^{N} g_n^e \times o_n \times z_n^t \Big/ \sum_{n=1}^{N} o_n \times z_n^t \le gu^{t,e} \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{6}$$

$$pl^t \le \sum_{n=1}^{N} o_n \times z_n^t \le pu^t \qquad \forall t \in \{1,...,T\} \qquad (7)$$

$$ml^t \le \sum_{n=1}^{N} (o_n + w_n) \times z_n^t \le mu^t \qquad \forall t \in \{1,...,T\} \qquad (8)$$

$$J \times z_n^t - \sum_{j=1}^{J} \sum_{i=1}^{t} z_j^i \le 0 \qquad \forall n \in \{1,...,N\}, \quad t \in \{1,...,T\}, \quad j \in D(J) \quad (9)$$

$$\sum_{t=1}^{T} z_n^t = 1 \qquad \forall n \in \{1,...,N\} \qquad (10)$$

Where Eq. (6) is grade blending constraints; these inequalities ensure that the head grade of the elements of interest and contaminants are within the desired range in each period. There are two equations (upper bound and lower bound) per element per scheduling period in Eq. (6). Eq. (7) is processing capacity constraints; these inequalities ensure that the total ore processed in each period is within the acceptable range of processing plant capacity. There are two equations (upper bound and lower) per period per ore type. Eq. (8) is mining constraints; these inequalities ensure that the total tonnage of material mined (ore, waste, overburden, and undefined waste) in each period is within the acceptable range of mining equipment capacity in that period. There are two equations (upper bound and lower bound) per period. Eq. (9) controls the precedence relationship of block extraction and pit slopes. For each block there is a set $D(J)_n \subset \mathcal{N}$,which includes all the blocks that must be extracted prior to mining block n to ensure that block n is exposed for mining with safe pit slopes, where J is the total number of blocks in set $D(J)_n$. The set $D(J)_1$ for the block labeled as 1 is hatched in Fig. 2. J is the total number of blocks in the set; it is equal to 28 in this case. Fig. 2 also demonstrates how variable slopes could be modeled by means of constructing the set $D(J)$; the block labeled as 5 is added to set $D(J)_1$ to construct a model with flatter slope on the right hand side. There is one equation per block per period for Eq. (9). Depending on the spatial location of the block in the block model the set of $D(J)$ would have different number of blocks. The slope constraints presented by Eq. (9) is the main reason of increase in the number of constraints and the complexity of Model 01 formulation. Finally, Eq. (10) defines reserve constraints; we assume that a final pit limit is superimposed on the block model and we are going to schedule the extraction of all the blocks within the final pit limit or push-back. Eq. (10) ensures that all the blocks within the final pit are going to be extracted once.

3.2. Model 02 - extraction and processing at block level – binary and continuous variables

In this model the mining and processing are at block level resolution; the schedule is controlled by continuous variables so fractional extraction of blocks may occur. The order of block extraction is controlled by binary integer variables at block level.

Objective function:

$$\max \sum_{t=1}^{T} \sum_{n=1}^{N} d_n^t \times u_n^t \qquad (11)$$

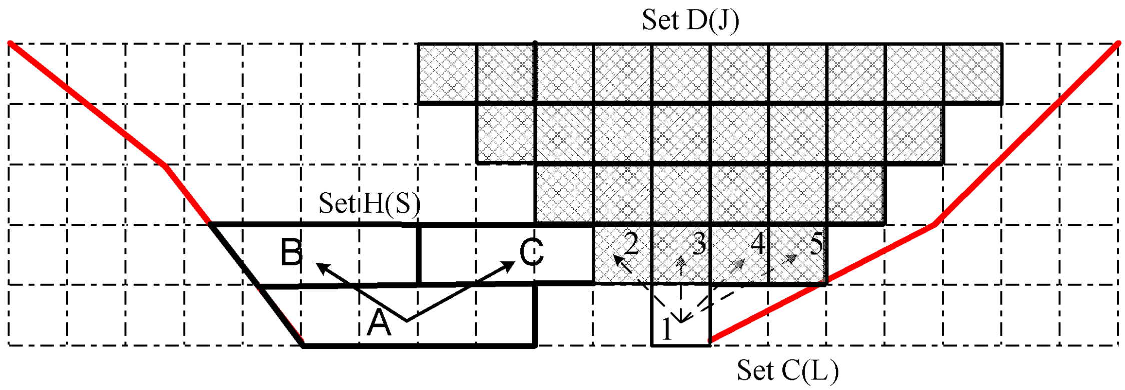


Fig. 2. Precedence of block extraction in the proposed MILP models.

Subject to:

$$gl^{t,e} \le \sum_{n=1}^{N} g_n^e \times o_n \times u_n^t \Big/ \sum_{n=1}^{N} o_n \times u_n^t \le gu^{t,e} \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{12}$$

$$pl^t \le \sum_{n=1}^{N} o_n \times u_n^t \le pu^t \qquad \forall t \in \{1,...,T\} \tag{13}$$

$$ml^t \le \sum_{n=1}^{N} (o_n + w_n) \times u_n^t \le mu^t \qquad \forall t \in \{1,...,T\} \tag{14}$$

$$a_n^t - \sum_{i=1}^{t} u_l^i \le 0 \qquad \forall n \in \{1,...,N\}, \quad t \in \{1,...,T\}, \quad l \in C(L) \tag{15}$$

$$\sum_{i=1}^{t} u_n^i - a_n^t \le 0 \qquad \forall n \in \{1,...,N\}, \quad t \in \{1,...,T\} \tag{16}$$

$$a_n^t - a_n^{t+1} \le 0 \qquad \forall n \in \{1,...,N\}, \quad t \in \{1,...,T-1\} \tag{17}$$

$$\sum_{t=1}^{T} u_n^t = 1 \qquad \forall n \in \{1,...,N\} \tag{18}$$

Eq. (12) controls the grade blending constraints. Eqs. (13) and (14) are processing and mining capacity constraints. Eqs. (15) to (17) control the relationship of block extraction precedence by binary integer variables at block level. Model 02 only requires the set of immediate predecessors' blocks on top of each block to model the order of block extraction relationship. This is presented by set $C(L)$ in Eq. (15). Fig. 2 illustrates the set $C(L)_1$ for the block labeled as 1, the set includes $C(L)_1 = \{2,3,4,5\}$. Where L the number of blocks is equal to 4; compare this to formulation in Model 01 where J was 28. Eq. (18) ensures that the fractions of blocks that are extracted over the scheduling periods are going to sum up to one, which means all the block within the final pit outline are going to be scheduled.

3.3. Model 03 – extraction at mining-cut level and processing at block level – binary and continuous variables

In this model, processing is at block level and extraction is at mining-cut level. The amount of ore processed is controlled by the continuous variable x_n^t, this allows fractional extraction of blocks in different periods. The order of block extraction which is controlled by the directed graph $G_b(\mathcal{N}, \mathcal{A})$ is transferred into the control of the precedence of mining-cuts by means of directed graph $G_c(\mathcal{K}, \mathcal{B})$. The precedence of mining-cuts relationship is modeled via the binary integer variable b_k^t. This is superior to Models 01 and 02, in that, the amount of ore processed and amount of material mined are controlled by two separate variables.

Objective function:

$$\max \sum_{t=1}^{T}\sum_{k=1}^{K}\left(\sum_{n\in c_k} v_n^t \times x_n^t - (\sum_{n\in c_k} q_n^t)\times y_k^t\right) \tag{19}$$

Subject to:

$$gl^{t,e} \le \sum_{n=1}^{N} g_n^e \times o_n \times x_n^t \Big/ \sum_{n=1}^{N} o_n \times x_n^t \le gu^{t,e} \qquad \forall t\in\{1,...,T\},\quad e\in\{1,...,E\} \tag{20}$$

$$pl^t \le \sum_{n=1}^{N} o_n \times x_n^t \le pu^t \qquad \forall t\in\{1,...,T\} \tag{21}$$

$$ml^t \le \sum_{k=1}^{K}\left(\sum_{n\in c_k}(o_n + w_n)\right)\times y_k^t \le mu^t \qquad \forall t\in\{1,...,T\} \tag{22}$$

$$x_n^t \le y_{k,n}^t \qquad \forall n\in\{1,...,N\},\quad n\in c_k,\quad t\in\{1,...,T\} \tag{23}$$

$$b_k^t - \sum_{i=1}^{t} y_s^i \le 0 \qquad \forall k\in\{1,...,K\},\quad t\in\{1,...,T\},\quad s\in H(S) \tag{24}$$

$$\sum_{i=1}^{t} y_k^i - b_k^t \le 0 \qquad \forall k\in\{1,...,K\},\quad t\in\{1,...,T\} \tag{25}$$

$$b_k^t - b_k^{t+1} \le 0 \qquad \forall k\in\{1,...,K\},\quad t\in\{1,...,T-1\} \tag{26}$$

Eqs. (20) to (22) control the grade blending, processing capacity, and mining capacity. In Model 02 grade blending, processing capacity, and mining capacity are modeled by only one decision variable, u_n^t. In Model 03 the extraction and processing of ore is controlled by continuous decision variable x_n^t at block level, where mining is modeled using a continuous variable y_k^t at mining-cut level. This method enables us to have a high resolution solution for selection of ore and processing. The total amount of material mined and the order of extraction is then modeled at mining-cut level y_k^t. This approach reduces the number of binary integer variables in the model drastically. Fig. 2 illustrates set $H(S)=\{B,C\}$, the predecessor mining-cuts that must be extracted prior to extraction of mining-cut A. Eq. (23) represents inequalities that ensure the amount of ore of any block which is processed in any given period is less than or equal to the amount of rock extracted from the mining-cut that the block belongs to in the considered time period. For each mining-cut, k, Eqs. (24) to (26) check the set of immediate predecessor cuts that must be extracted prior to extracting mining-cut, k.

3.4. Model 04 – extraction and processing at mining-cut level – binary and continuous variables

In this model, mining and processing are both at mining-cut level. The blocks are aggregated prior to schedule optimization and the ore processing and mining are controlled by two continuous variables.

Objective function:

$$\max \sum_{t=1}^{T}\sum_{k=1}^{K}(v_k^t \times s_k^t - q_k^t \times y_k^t) \tag{27}$$

$$gl^{t,e} \le \sum_{k=1}^{K} g_k^e \times o_k \times s_k^t \Big/ \sum_{k=1}^{K} o_k \times s_k^t \le gu^{t,e} \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{28}$$

$$pl^t \le \sum_{k=1}^{K} o_k \times s_k^t \le pu^t \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{29}$$

$$ml^t \le \sum_{k=1}^{K} (o_k + w_k) \times y_k^t \le mu^t \qquad \forall t \in \{1,...,T\} \tag{30}$$

$$s_k^t \le y_k^t \qquad \forall k \in \{1,...,K\}, \quad t \in \{1,...,T\} \tag{31}$$

Equations (24) to (26)

Eqs. (28) to (30) control the grade blending, processing capacity, and mining capacity constraints at mining-cut level with fractional extraction from mining-cuts. The extraction from mining-cuts is assumed to be uniform among all the blocks which belong to that mining-cut. Eq. (31) ensures that the amount of ore extracted and processed from any mining-cut in any given period is less than or equal to the amount of rock extracted from that mining-cut. Eqs. (24) to (26) are similar to those demonstrated in Model 03 for precedence of block extraction.

4. Numerical modeling

In most linear optimization problems, the variables of the objective function are continuous in the mathematical sense, with no gaps between real values. To solve such linear programming problems, ILOG CPLEX implements optimizers based on the simplex algorithms (Winston, 1995) (both primal and dual simplex) as well as primal-dual logarithmic barrier algorithms.

Branch and cut is a method of combinatorial optimization for solving integer linear programs. The method is a hybrid of branch and bound and cutting plane methods (Horst and Hoang, 1996). Refer to Wolsey (1998) for a detailed explanation of the branch and cut algorithm, including cutting planes. In recent years there has been significant improvements in mathematical programming optimizers such as ILOG CPLEX (Bixby, 1987-2009). This optimizer uses branch and cut techniques to solve MILP models and it makes the latest theory in optimization of large-scale industrial problems available commercially. In this study we used TOMLAB/CPLEX version 11.2 (Holmström, 1989-2009) as the MILP solver. TOMLAB/CPLEX efficiently integrates the solver package CPLEX (ILOG Inc, 2007) with MATLAB environment (MathWorks Inc., 2007). An important termination criterion that the user can set explicitly in CPLEX is the MILP gap tolerance. We have used the relative MILP gap tolerance, which indicates to CPLEX to stop when an integer feasible solution has been proved to be within the gap tolerance of optimality.

4.1. Size and complexity

One of the major obstacles in using the MILP formulations for mine production scheduling is the sheer size of the problem. The number of blocks, N, in the model is usually between tens of thousands to millions which will lead to a formulation with an objective function with many variables. Moreover, the main physical constraint in open pit mining is the block extraction precedence modeled by binary integer variables. This set of constraints also controls the overall pit slope in different regions. The numbers of blocks that must be extracted prior to mining each block are numerous and will result in formulations with many constraints. Therefore, we are dealing with an MILP formulation with many variables and many constraints.

The most common difficulty with MILPs is the size of the branch and cut tree. The tree becomes so large that insufficient memory remains to solve an LP sub-problem. The number of binary integer variables in the formulations determines the size of the branch and cut tree. As a general strategy in our formulations we aimed at reducing the number of binary integer variables, we also focused on developing formulations that will mainly use continuous optimization techniques rather than discrete optimization. Table 1 shows the number of decision variables and the number of binary integer variables required for the proposed MILP formulations as a function of number of blocks, N, number of mining-cuts, K, and number of scheduling periods, T. The goal has been to reduce the number of binary integer variables in the models by introducing mining-cuts as the means of controlling the precedence of extraction of blocks rather than having one binary integer variable per block.

Table 1 – number of decision variables in the MILP formulations

MODEL	Number of decision variables	Number of integer variables
Model 01	$N \times T$	$N \times T$
Model 02	$2 \times N \times T$	$N \times T$
Model 03	$(2 \times K + N) \times T$	$K \times T$
Model 04	$3 \times K \times T$	$K \times T$

5. Results and discussions

We have developed, implemented, and tested the proposed MILP models presented in section 3 in TOMLAB/CPLEX environment (Holmström, 1989-2009). We compare the performances of the proposed models based on net present value generated, practical mining production constraints, smoothness of the generated schedules, size of the mathematical formulations, the number of integer variables required in formulation, and computational time required for convergence.

All the developed formulations are verified by numerical experiments on a synthetic data set containing 120 blocks and a real mining push-back including 2 598 blocks with seven mining benches. We have also validated Model 03 and Model 04 with an iron ore life-of-mine schedule with 26 000 blocks over a 20 year scheduling horizon. We tested our models on a Quad Core Dell Precision T7400 computer at 3.00 GHz, with 3.25 GB of RAM. Since we aimed at a comparative analysis of the proposed models a relative tolerance of 2% on the gap between the best integer objective and the feasible integer solution was chosen.

Table 2a and b show the numerical results of the tests of MILP models with the data set containing 120 blocks over four periods of extraction. To reach a feasible solution in different models we have used a mining capacity upper bound of 64 to 66 million tonnes per period, whereas the processing capacity varies between 4.9 to 5.3 million tonnes per period. We were forced to set different upper and lower bounds for different models, because the tighter boundaries on some models resulted in infeasible solutions. As expected Model 02 generated the highest NPV; in this formulation processing and mining are both at block level with continuous variables so fractional extraction of blocks are allowed. The variables in Model 02 have the highest resolution among all the other models. Model 01 had the longest runtime with 42.53 seconds as expected; this is due to the numerous numbers of constraints that are generated by Eq. (9). Also, the model is a pure MIP with no continuous variables, the formulation searched 4 432 branch and bound nodes to reach an optimized solution within a 2.25% gap. Ramazan et al. (2005) proposed to reduce the number of binary integer variables required by Model 01 by defining only the ore blocks as binary variables; this approach will reduce the number of binary integer variables but since the formulation assigns blocks to periods of extraction, rather than determining a strict temporal sequence of blocks, still the size of the problem and the runtime even for a small number of blocks is not within a reasonable timeframe and simply this formulation is not a practical tool. In Models 03 and 04 the concept of mining-cuts are introduced and the number of binary integer variables and the computational time required is reduced drastically compared to Model 01 and 02 (>72% and >99.6%, respectively). On the other hand, the NPV of Model 03 and 04 are reduced by 0.72% and 1.11% when compared to Model 02, as in Model 03 and Model 04 the variables are aggregated and have less freedom in the MILP formulations.

Table 2a – Inputs and numerical results for the synthetic data set containing 120 blocks

MODEL	Blocks – Cuts $(N-K)$	Periods (T)	mu^t / ml^t (MT)	pu^t / pl^t (MT)	NPV ($M)
01	120 - 0	4	64 / 0	5.3 / 0	387.88
02	120 - 0	4	64 / 0	4.9 / 0	391.34
03	120 - 21	4	66 / 0	5.0 / 0	388.53
04	120 - 21	4	66 / 0	5.0 / 0	387.00

Table 2b – Inputs and numerical results for the synthetic data set containing 120 blocks

MODEL	Root Node Gap %	CPU TIEM (S)	Coefficient Matrix $A(rows \times col)$	No. decision variables	No. Integer variables	No. of B & B nodes visited
01	2.25	42.53	372 × 480	480	480	4432
02	2.37	0.61	3004 × 960	960	480	38
03	1.29	0.17	920 × 648	648	84	0
04	1.40	0.15	530 × 252	252	84	0

Table 3a and Table 3b show the numerical results of the MILP models for an iron ore push-back data set containing 2 598 blocks over twelve scheduling periods. Fig. 3 and 4 illustrate cross sections of the final pit limits including the orebody and rock type model. The push-back studied includes seven benches of the final pit demonstrated in Fig. 3 and 4 from elevation 1500m to 1590m. The blocks represent a volume of rock equal to 50m×25m×15m. The model contains 155 million tonnes of material with 84 million tonnes of iron ore with an average grade of 73% magnetic weight recovery (MWT%). Sulfur and phosphor are present as deleterious elements and their grades need to be controlled within an acceptable range in the processing plant feed.

Table 3a - Inputs and numerical results for the data set containing 2598 blocks

MODEL	Blocks -Cuts ($N-K$)	Periods (T)	mu^t / ml^t (MT)	pu^t / pl^t (MT)	$gu^{e,t} / gl^{e,t}$ (MWT%)	$gu^{e,t} / gl^{e,t}$ (S% & P%)	NPV ($M)
01	2598 - 0	12	13 / 0	7.15 / 0	No bounds	0 / 1.8 & 0.14	-
02	2598 - 0	12	13 / 0	7.15 / 0	No bounds	0 / 1.8 & 0.14	3011.65
03	2598 - 148	12	13.2 / 0	7.15 / 0	65 / 80	0 / 1.8 & 0.14	2947.68
04a	2598 - 148	12	13.5 / 0	7.15 / 7.0	65 / 80	0 / 1.8 & 0.14	2947.09
04b	2598 - 239	12	13.5 / 0	7.15 / 7.0	65 / 80	0 / 1.8 & 0.14	2985.02
04c	2598 - 436	12	13.5 / 0	7.15 / 7.0	65 / 80	0 / 1.8 & 0.14	2991.55

Table 3b – Inputs and numerical results for the data set containing 2598 blocks

MODEL	Root Node GAP %	CPU TIME (S)	Coefficient Matrix $A(rows \times cols)$	No. Of nonzero elements in A	No. Integer variables	No. of B & B nodes visited
01	No Integer	Solution	-	5 668 788	31 176	-
02	0.05	14 748.32	280 944 × 62 352	2 091 882	31 176	340
03	1.08	1 168.60	43 202 × 34 728	226 774	1 776	70
04a	0.52	26.73	11 232 × 5 325	75 640	1 776	50
04b	0.35	34.37	17 580 × 8 604	115 898	2 868	40
04c	0.58	34.94	30 264 × 15 696	198 082	5 232	30

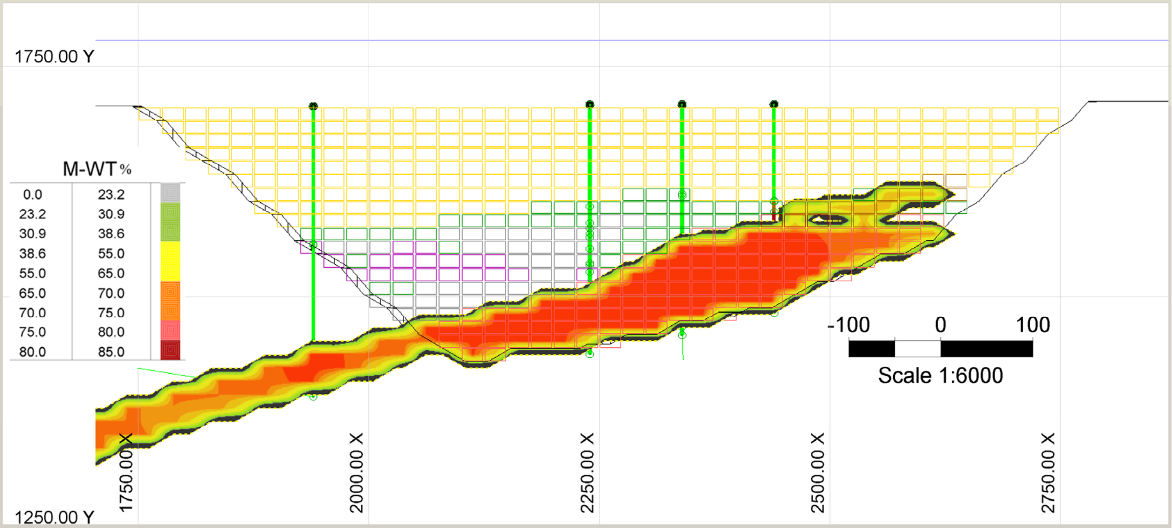


Fig. 3. Cross section 98400 N of the final pit including the orebody and rock type model looking west (meters).

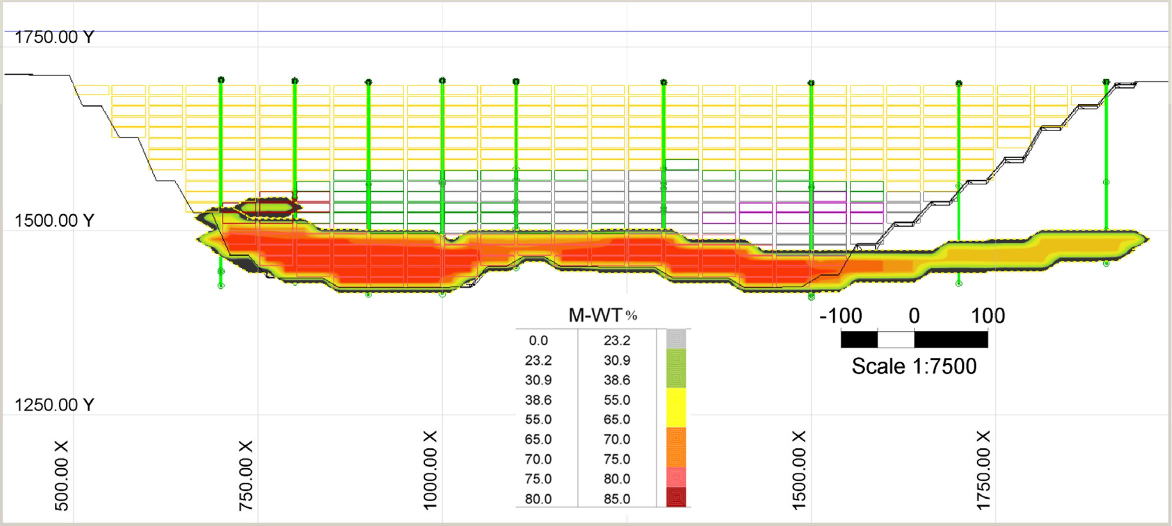


Fig. 4. Cross section 600640 E of the final pit including the orebody and the rock type model looking north (meters).

The maximum allowable average grade for sulfur is 1.8% and for phosphor is 0.14% per period. It is also desirable to keep an average head grade between 65% and 80% of magnetic weight recovery (MWT). Our goal was to generate a schedule with a uniform feed within a range of 7 to 7.15 million tonnes of ore per period. Furthermore, we intended to keep a steady (1.89 to 1.93) stripping ratio over the scheduling horizon, so we chose a maximum mining equipment capacity of 13.5 million tonnes per period. This would ensure that the mining equipment capacity required is not going to fluctuate over time.

To schedule 2598 blocks over 12 scheduling periods, Model 01 generated a huge coefficient matrix with more than 5.6 million nonzero elements and an integer solution for the formulation did not exist (Table 3b). Model 02 generated a coefficient matrix with more than 2 million nonzero elements and ran for over four hours. As expected, Model 02 generated the highest NPV of $3011.65 million of all the models, with a 0.05% gap tolerance. Fig. 5a to 5c show the plan view and cross sections of the generated schedule by Model 02. A closer examination of Fig. 5a to 5c reveals a tight schedule from a practical mining point of view. Fig. 5b shows that in the second period mining occurs on six active benches. This requires considerable equipment movement. Implementation of such a schedule is a challenge in the field, if limited numbers of mining shovels are available.

In Model 03 we have used clustering techniques to aggregate the blocks into 148 mining-cuts, clustering has reduced the number of integer variables to 1 776 from 31 176 in Model 02. Although in Model 03 the processing is at block level, the NPV has dropped to $2947.68 million, a 2.1% reduction (Table 3b). This is because mining occurs at the mining-cut level and has reduced flexibility when compared to Model 02. The run time has been reduced to half an hour from four hours.

Figs. 6a to 6c illustrate the plan view and cross sections of the schedule generated by Model 03. This schedule is more practical from a mining point of view since there are only three active benches in the second period compared to the six active benches of Model 02.

We have investigated the effect of number of mining-cuts on the quality of solutions in terms of NPV and the run-time on three different cases with Model 04. We clustered the 2598 blocks into 148, 239, and 436 mining-cuts using the clustering algorithm. We refer to these as Model 04a, 04b, and 04c, respectively. A model with fewer mining-cuts will have fewer number of integer variables in the MILP formulation and result in reduced run-time, as shown in Table 3b.

Alternatively, models with more mining-cuts imply more freedom for the decision variables and higher NPV would be expected. The NPV for Model 04a with 148 mining-cuts is $2947.09 million, while the NPV for the Model 04c with 436 cuts has increased to $2991.55 million. One of the very important improvements with the Model 04 formulations is the drastic drop in runtime. Compare the four hour runtime for Model 02 to the less than 35 seconds for Model 04. This is a very significant improvement; one must take into account that we are examining a very small model with only 2598 blocks in this study. The increase in the number of blocks would very quickly make Model 02 intractable and the model is not going to converge for larger real size models. Scrutinizing Fig. 7a to 7c shows that the number of active benches for different periods of extraction has decreased to two or three benches. The reduction of number of active benches is because of clustering blocks into mining-cuts. A smaller number of mining-cuts would generate clusters with more blocks. On the other hand, we are constructing the mining-cuts on a bench by bench basis; which tends to generate schedules that expand the mine outlines more horizontally rather than vertically. More research is required to develop a framework that will optimize the number of blocks included in each mining-cut in terms of making a balance in aiming for the highest NPV possible, while the maximum number of active benches included in the schedule would be practical from a mining point of view.

One of the features that make MILP formulations a robust platform for mine planning is not only the NPV maximization but also the control that the mine planner would have on upper and lower bounds of ore and waste production targets.

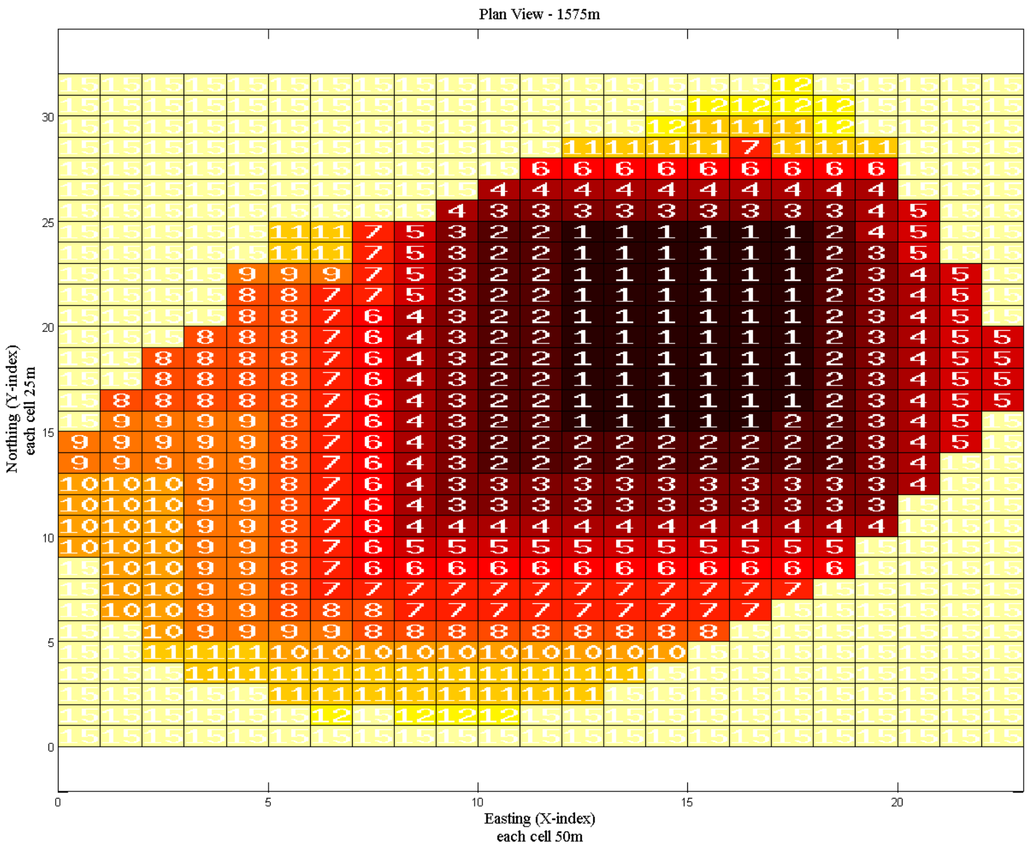


Fig. 5a. Model 02, plan view of bench 1575m.

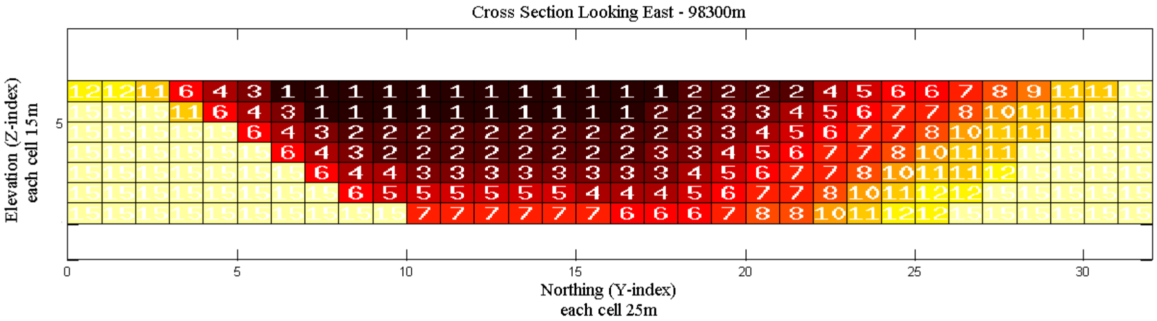


Fig. 5b. Model 02, cross section 98300m looking east.

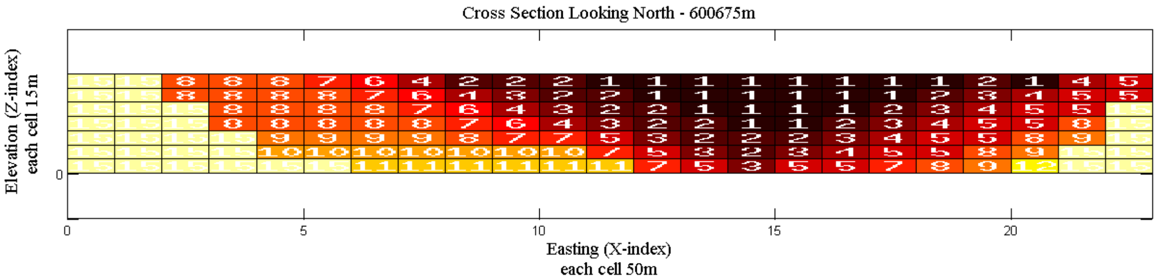


Fig. 5c. Model 02, cross section 600675m looking north.

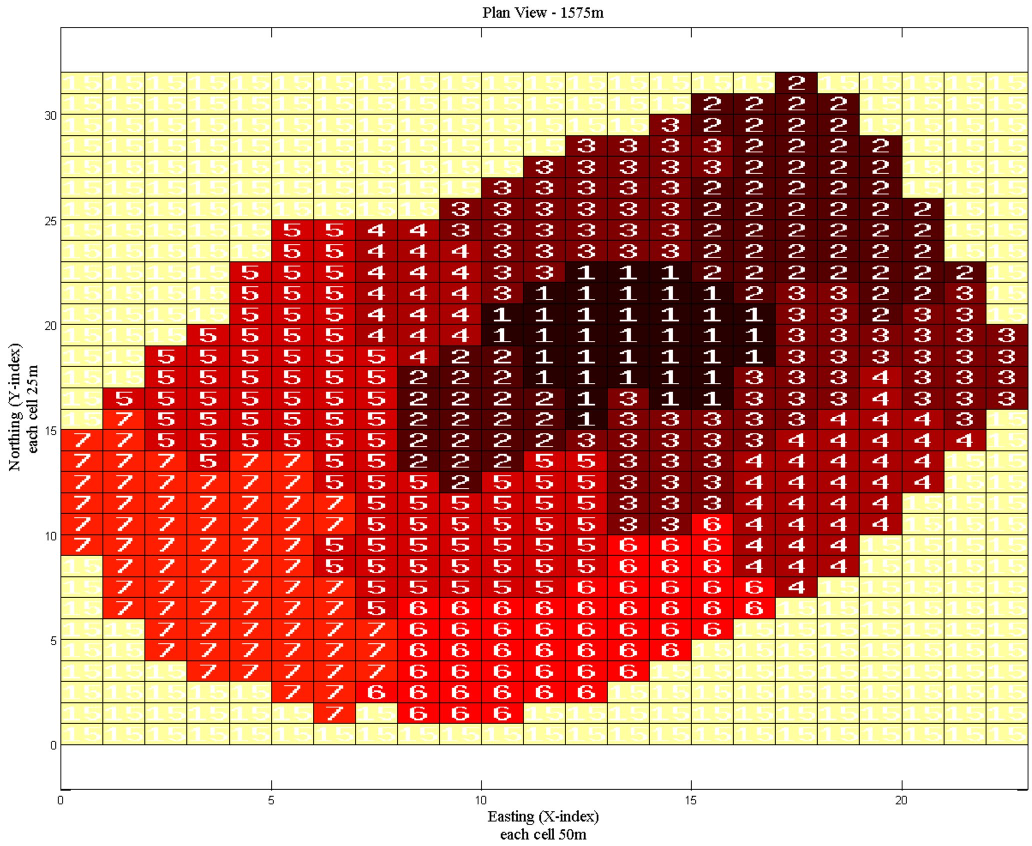


Fig. 6a. Model 03, plan view of bench 1575m.

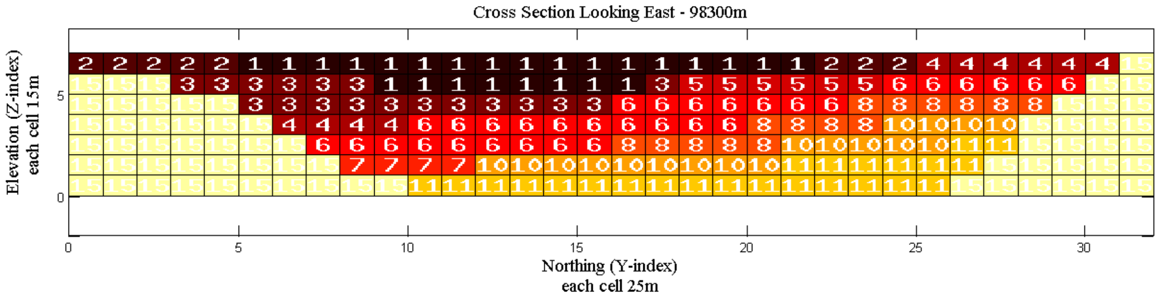


Fig. 6b. Model 03, cross section 98300m looking east.

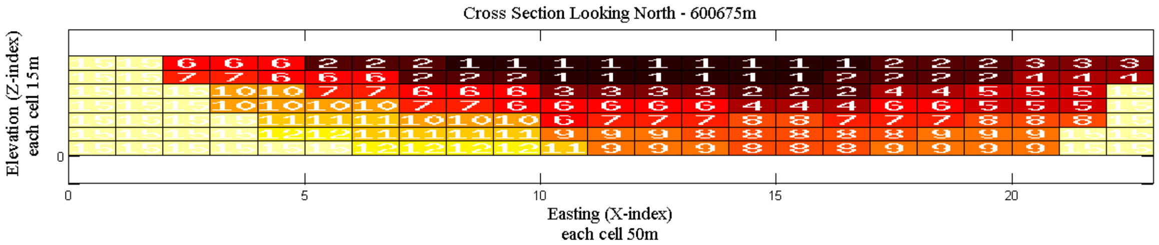


Fig. 6c. Model 03, cross section 600675 looking north.

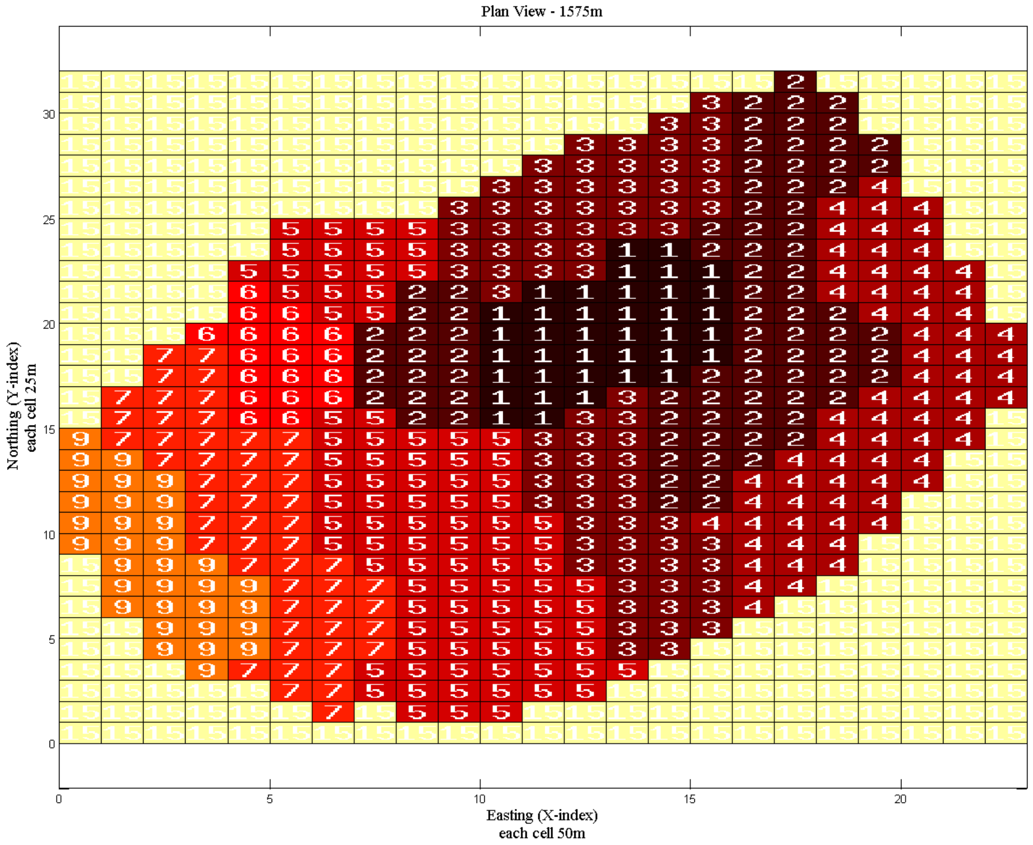


Fig. 7a. Model 04a, plan view of bench 1575m.

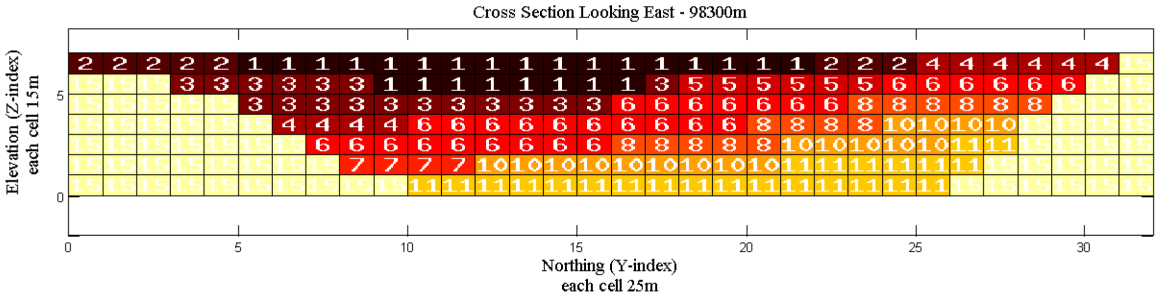


Fig. 7b. Model 04a, cross section 98300m looking east.

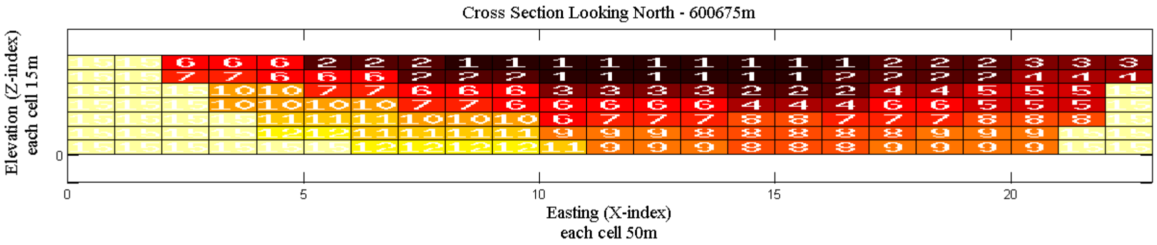


Fig. 7c. Model 04a, cross section 600675m looking north.

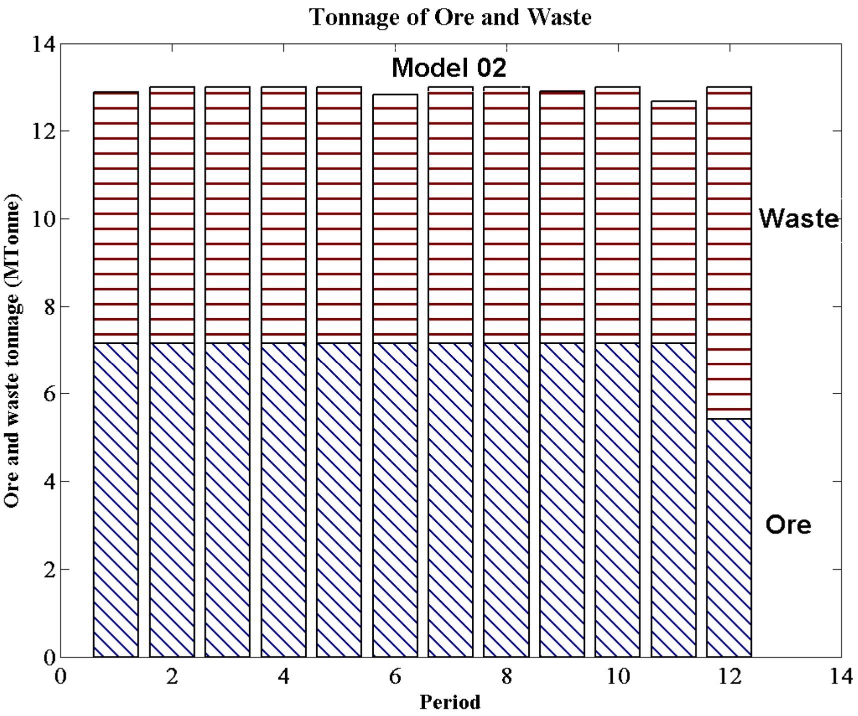


Fig. 8a. Model 02 tonnage of ore and waste per period.

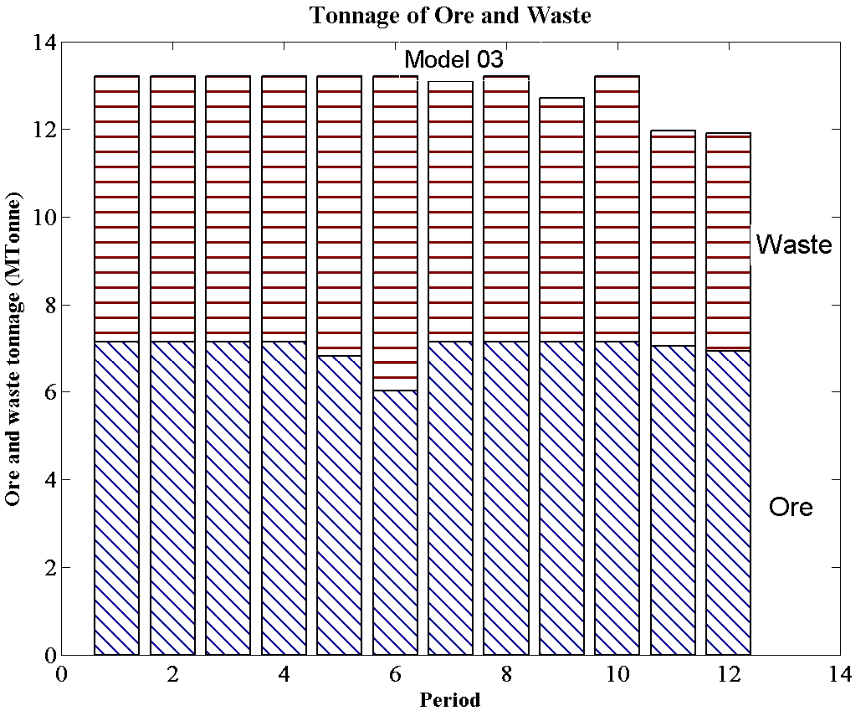


Fig. 8b. Model 03 tonnage of ore and waste per period.

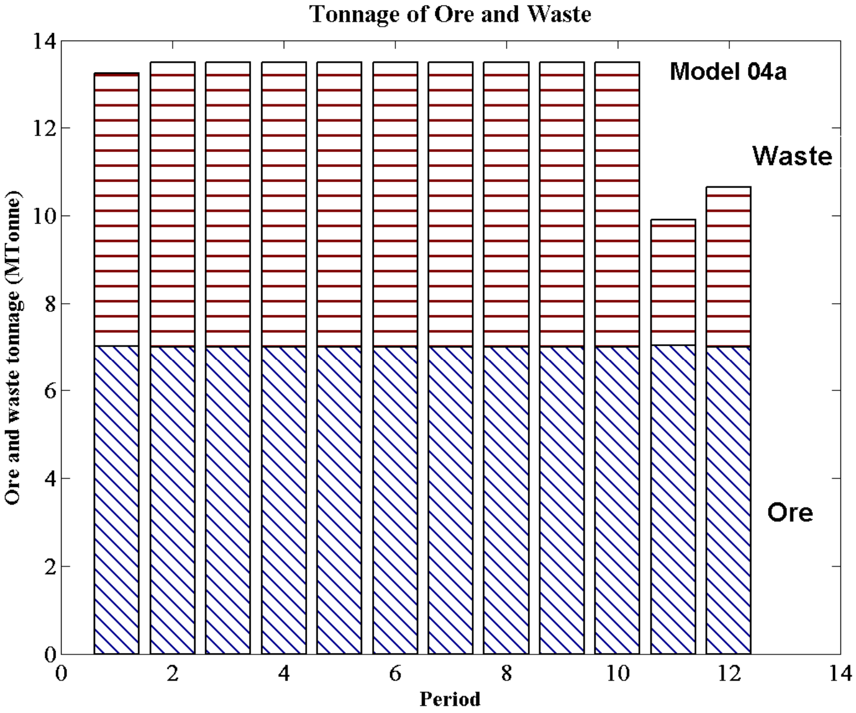


Fig. 8c. Model 04a tonnage of ore and waste per period.

Also, there is an inherent task of blending the run-of- mine materials before concentration. The objective is to mine in such a way that the resulting mix meets the quality and quantity specifications of the processing plant. The blending problem becomes more important as we get more into detailed planning in short/medium range plans.

Fig. 8a to 8c illustrate the yearly tonnage of ore processed, waste mined, and the total tonnage of material mined in each period of production. In Model 02 we set a maximum mining capacity of 13 million tonnes per period and a processing capacity of 7.15 million tonnes per period (Table 3a) with no lower bound on mining and processing capacities. Fig. 8a illustrates the results of Model 02, the generated schedule is smooth with very little fluctuations in the tonnage of feed and stripping ratio. Meanwhile, referring Fig. 9 to 11 for the blending results show that the average grade of MWT, sulphur, and phosphor are within the acceptable range defined in Table 3a (plots are for Models 02, 03, and 04a). A very interesting phenomenon that should be noticed is how in Model 02 (Fig. 9), the MILP high grades for iron ore in the early periods and then the MWT average grade starts to get lower every year. The high grading phenomenon in early periods is completely in accordance with the highest NPV generated by Model 02.

In Model 03 we set a maximum mining capacity of 13.2 million tonnes per period and a processing capacity of 7.15 million tonnes per period (Table 3a) with no lower bound on mining and processing capacities. We were forced to increase the mining upper bounds since the 13 million tonnes upper bound did not generate a feasible integer solution; this is because of introduction of mining-cuts into the model, which reduces the freedom of the variables. Assessment of Fig. 8b shows that the ore feed is not as smooth as Model 02 schedule. Grade blending constraints are all honored except for phosphor in the first period that exceeded to 0.145% rather than 0.14%.

In Model 04a we set a maximum mining capacity of 13.5 million tonnes per period and a processing capacity of 7.15 million tonnes per period (Table 3a) with a lower bound of 7 million tonnes on the processing capacity. All the constraints presented in Table 3a are honored and the ore production schedule is the smoothest among all with an iron ore grade of 65% to 80% average for the MWT.

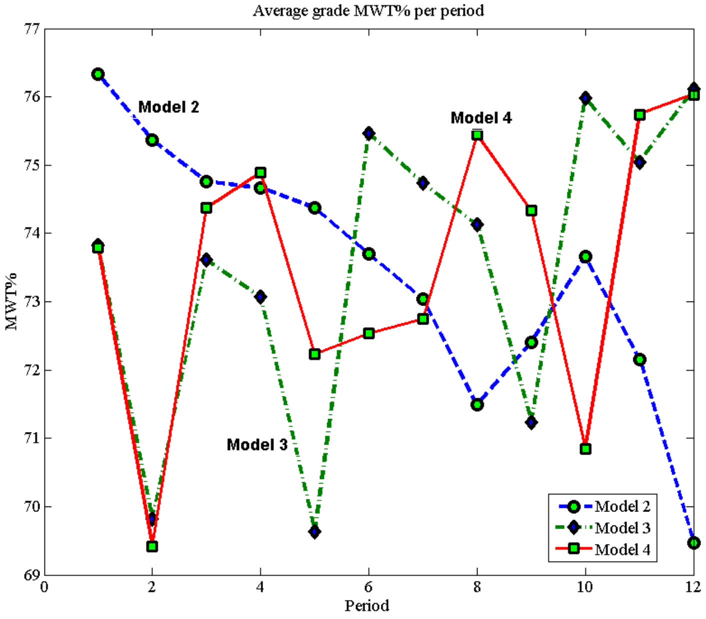


Fig. 9. Average iron ore (MWT%) grade per period.

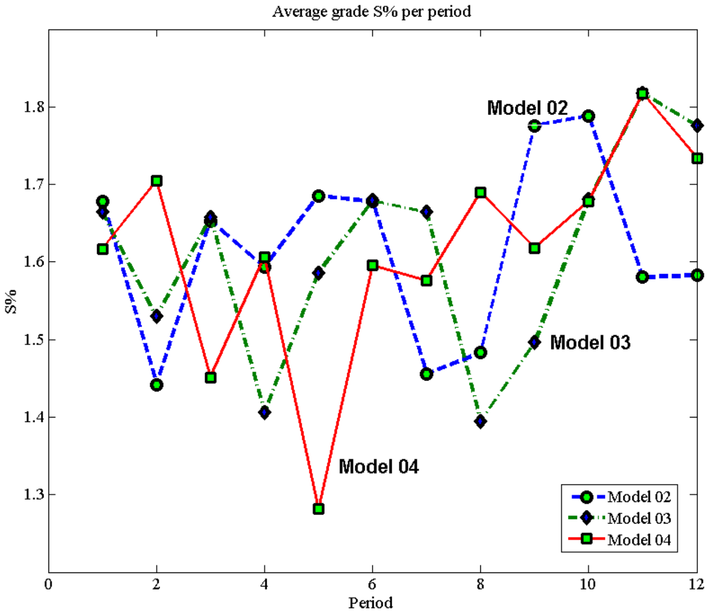


Fig. 10. Average sulphur grade per period.

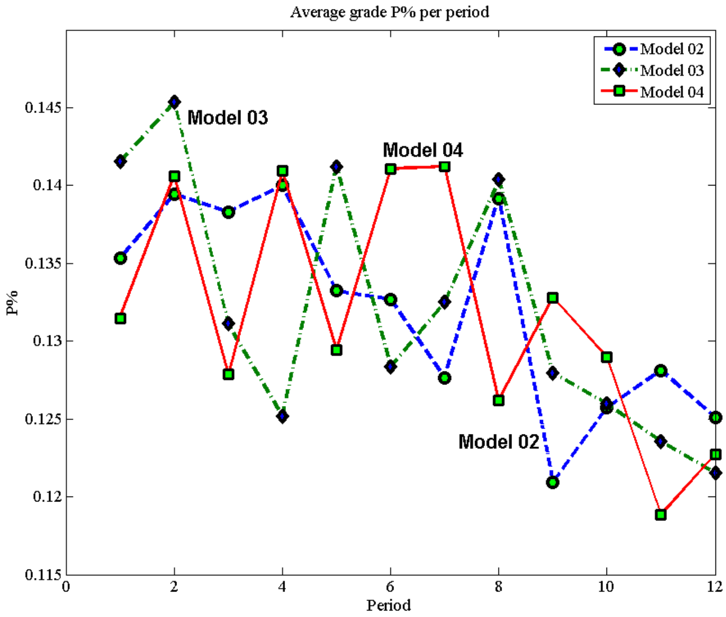


Fig. 11. Average phosphor grade per period.

6. Conclusions and future work

The paper investigated the shortcomings of the current mixed integer linear programming (MILP) models used for open pit production scheduling, particularly the inability to solve large-scale real-size mining problems. In this study, we have developed, implemented, and tested MILP theoretical frameworks for large-scale open pit production scheduling. The developed models proved to be able to handle deterministic large-scale mine production problems.

To reduce the size of the open pit production scheduling problem we introduced the concept of mining-cuts into the MILP formulations. Blocks within the same level or mining bench are grouped into clusters based on their attributes, spatial location, rock type, and grade distribution. Four MILP formulations are presented: Model 01- only consists of binary integer decision variables and generates a schedule at block level resolution; Model 02 – the schedule is generated based on a strict temporal sequence of blocks. The formulation uses continuous variables to model extraction and processing at block level. Binary integer decision variables, are used to control precedence of extraction; Model 03 – processing is controlled at block level with continuous decision variables and the precedence of extraction of blocks is controlled at the mining-cut level by means of binary integer variables; Model 04 –Extraction, processing, and order of block extraction are controlled at mining-cut level. We have implemented the optimization formulations in TOMLAB/CPLEX (Holmström, 1989-2009) environment.

An iron ore mine intermediate scheduling case study over twelve periods was carried out to compare, verify and validate the models. The summary of the comparative analyses revealed that: (i) an MILP formulation at block level resolution (Model 02) is not suitable for long-term scheduling. Although, the block level formulation generates a higher NPV compared to the rest of the models, they would quickly go out of memory and it is almost impossible to generate a life-of-mine schedule for a real-size mine; (ii) block level resolution models (Model 02 and Model 03) are more appropriate for short-range scheduling where the number of blocks are in the order of thousands to ten thousand and the scheduling periods are in the order of ten to twelve periods. These formulations could be used to break-down the long-term yearly schedule into a monthly schedule; (iii) introduction of mining-cuts as the mining units (Models 03 and 04) and as a means to control the precedence of block extraction would drastically reduce the size and runtime of the MILP formulations and the number of binary integer variables required; (iv) MILP formulations based on processing and extraction at mining-cut level (Model 04) leads to practical mathematical tools addressing life-of-mine schedules of large-scale open pit operations which was impossible to solve with the previous formulations. These models provides control over all the mining, processing, and blending constraints, to the mine planner while *maximizing* the NPV; (vi) clustering algorithms and the number of mining-cuts affect the computational efficiency of developed MILP formulations and the generated optimal NPV.

Further focused research is underway to develop and test different clustering techniques that would generate an optimized clustering approach for mining-cuts. Also the next step is to extend the mixed integer linear programming frame work into stochastic mathematical programming domain to address the geological uncertainty issue.

7. References

[1] Askari-Nasab, H., (2006), "Intelligent 3D interactive open pit mine planning and optimization", PhD Thesis Thesis, © University of Alberta, Edmonton, Canada, Pages 167.

[2] Askari-Nasab, H. and Awuah-Offei, K., (2009), "Open pit optimization using discounted economic block value", *Transactions of the Institution of Mining and Metallurgy. Section A, Mining industry*, Vol. 118, 1, pp. 1-12.

[3] Askari-Nasab, H., Frimpong, S., and Awuah-Offei, K., (2005), "Intelligent optimal production scheduling estimator", *in Proceedings of 32nd Application of Computers and Operation Research in the Mineral Industry*, © Taylor & Francis Group, London, Tucson, Arizona, USA, pp. 279-285.

[4] Askari-Nasab, H., Frimpong, S., and Szymanksi, J., (2007), "Modeling open pit dynamics using discrete simulation", *International Journal of Mining, Reclamation and Environment*, Vol. 21, 1, pp. 35- 49.

[5] Askari-Nasab, H., Frimpong, S., and Szymanksi, J., (2008), "Investigating the continuous time open pit dynamics", *The Journal of the South African Institute of Mining and Metallurgy*, Vol. 108, 2, pp. 61-73.

[6] Askari-Nasab, H. and Szymanski, J., (2007), "Open pit production scheduling using reinforcement learning", *in Proceedings of 33rd International Symposium on*

Computer Application in the Minerals Industry (APCOM), © GECAMIN LTDA, Santiago, Chile, pp. 321-326.

[7] Bixby, R. E., (1987-2009), "ILOG CPLEX", ver. 11.0, Sunnyvale, CA , USA: ILOG, Inc.

[8] Boland, N., Dumitrescu, I., Froyland, G., and Gleixner, A. M., (2009), "LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity", *Computers and Operations Research*, Vol. 36, 4, pp. 1064-89.

[9] Caccetta, L. and Hill, S. P., (2003), "An application of branch and cut to open pit mine scheduling", *Journal of Global Optimization*, Vol. 27, November, pp. 349 - 365.

[10] Dagdelen, K. and Kawahata, K., (2007), "Oppurtunities in Multi-Mine Planning through Large Scale Mixed Integer Linear Programming Optimization", *in Proceedings of 33rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 337-342.

[11] Datamine Corporate Limited, (2008), "NPV Scheduler", ver. 4, Beckenham, United Kingdom: Datamine Corporate Limited,.

[12] Denby, B. and Schofield, D., (1994), "Open-pit design and scheduling by use of genetic algorithms", *Transactions of the IMM Section A*, Vol. 103, January - April 1994,, pp. A21-A26.

[13] Denby, B., Schofield, D., and Hunter, G., (1996), "Genetic algorithms for open pit scheduling - extension into 3-dimensions", *in Proceedings of 5th International Symposium on Mine Planning and Equipment Selection*, © A.A.Balkema/Rotterdam/Brookfield, Sao Paulo, Brazil, pp. 177-186.

[14] Gemcom Software International, I., (1998-2008), "Whittle strategic mine planning software", ver. 4.2, Vancouver, B.C.: Gemcom Software International.

[15] Gershon, M., (1987), "Heuristic approaches for mine planning and production scheduling", *Geotechnical and Geological Engineering*, Vol. 5, 1, pp. 1-13.

[16] Gershon, M. E., (1983), "Mine scheduling optimization with mixed integer programming", *Mining Engineering*, Vol. 35, 4, pp. 351-354.

[17] Holmström, K., (1989-2009), "TOMLAB /CPLEX", ver. 11.2, Pullman, WA, USA: Tomlab Optimization.

[18] Horst, R. and Hoang, T., (1996), "Global optimization : deterministic approaches", © Springer, Berlin ; New York, 3rd ed, Pages xviii, 727 p.

[19] ILOG Inc, (2007), "ILOG CPLEX 11.0 User's Manual September", ver. 11.0: ILOG S.A. and ILOG, Inc.

[20] Isaaks, E. H., (1990), "The application of Monte Carlo methods to the analysis of spatially correlated data, Ph.D. thesis," PhD Thesis, © Stanford University, Stanford, CA, USA, Pages 226.

[21] Johnson, T. B., (1969), "Optimum open-pit mine production scheduling", *in Proceedings of Proceedings, 8th International Symposium on Computers and Operations Research*, © Salt Lake City, Utah, USA, pp. 539-562.

[22] Johnson, T. B. and Barnes, R. J., (1988), "Application of the maximal flow algorithm to ultimate pit design", in *Engineering design : better results through operations research methods*, Vol. 8, *Publications in operations research series*, R. R. Levary, Ed. New York, © North-Holland, pp. xv, 713.

[23] Kaufman, L. and Rousseeuw, P. J., (1990), "Finding groups in data : an introduction to cluster analysis", © Wiley, New York, Pages xiv, 342 p.

[24] Krige, D. G., (1951), "A statistical approach to some basic mine valuation and allied problems at the Witwatersrand", MSc Thesis, © University of Witwatersrand, South Africa,

[25] Lerchs, H. and Grossmann, I. F., (1965), "Optimum design of open-pit mines", *The Canadian Mining and Metallurgical Bulletin, Transactions*, Vol. LXVIII, pp. 17-24.

[26] MathWorks Inc., (2007), "MATLAB Software", ver. 7.4 (R2007a): MathWorks, Inc.

[27] Minemax Pty Ltd, (1998-2009), "MineMax Scheduler", ver. 4.2, West Perth, Western Australia: Minemax Pty Ltd.

[28] Ramazan, S., (2007), "Large-scale production scheduling with the fundamental tree algorithm - model, case study and comparisons", *in Proceedings of Orebody Modelling and Strategic Mine Planning*, © The Australian Institute of Mining and Metallurgy, Perth, Western Australia, pp. 121-127.

[29] Ramazan, S., Dagdelen, K., and Johnson, T. B., (2005), "Fundamental tree algorithm in optimising production scheduling for open pit mine design", *Mining Technology : IMM Transactions section A*, Vol. 114, 1, pp. 45-54.

[30] Ramazan, S. and Dimitrakopoulos, R., (2004), "Traditional and new MIP models for production scheduling with in-situ grade variability", *International Journal of Surface Mining, Reclamation & Environment*, Vol. 18, 2, pp. 85-98.

[31] Runge Limited, (1996-2009), "XPAC Autoscheduler", ver. 7.8: Runge Limited.

[32] Stone, P., Froyland, G., Menabde, M., Law, B., Pasyar, R., and Monkhouse, P. H. L., (2007), "Blasor - Blended Iron ore mine planning optimization at Yandi, Western Australia", *in Proceedings of Orebody Modelling and Strategic Mine Planning*, © The Australian Institute of Mining and Metallurgy, Perth, Western Australia,

[33] Sutton, R. S. and Barto, A. G., (1998), "Reinforcement learning, an Introduction", © The MIT Press, Cambridge, Massachusetts, Pages 432.

[34] Tolwinski, B. and Underwood, R., (1996), "A scheduling algorithm for open pit mines", *IMA Journal of Mathematics Applied in Business & Industry*, 7, pp. 247-270.

[35] Whittle, G., (2007), "Global asset optimization", *in Proceedings of Orebody Modelling and Strategic Mine Planning*, © The Australian Institute of Mining and Metallurgy, Perth, Western Australia, pp. 331-336.

[36] Winston, W. L., (1995), "Introduction to mathematical programming : applications and algorithms", © Duxbury Press, Belmont, Ca., 2nd ed, Pages xv, 818, 39 p.

[37] Wolsey, L. A., (1998), "Integer programming", © J. Wiley, New York, Pages xviii, 264 p.

[38] Yegulalp, T. M. and Arias, J. A., (1992), "A fast algorithm to solve the ultimate pit limit problem", *in Proceedings of 23rd APCOM Symposium*, © AIME, Littleton, Colorado, pp. 391-397.

[39] Zhao, Y. and Kim, Y. C., (1992), "A new optimum pit limit design algorithm", *in Proceedings of 23rd APCOM Symposium*, © SME, Littleton, Colorado, University of Arizona, pp. 423-434.

8. Appendix

MATLAB and TOMLAB/CPLEX code and documentation for Model 03.

The code is for processing at block level and mining at mining-cut level.

Numerical modelling of the MILP formulation for open pit production scheduling

Hooman Askari-Nasab and Kwame Awuah-Offei[1]
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

Development of new optimization techniques and uncertainty quantification for long-term mine planning plays a vital role in reducing environmental footprint and financial risk of mining projects. Deviations from optimal plans in mega mining projects will result in huge financial losses, delayed reclamation, and resource sterilization. In this research we developed a mathematical methodology for optimal large-scale open pit mine production scheduling. The deterministic mathematical models developed in this project will pave the way for the uncertainty quantification associated with mine plans by means of stochastic mathematical programming. We have developed a mixed integer linear programming model based on mining-cuts with reduced number of binary integer variables. The numerical modeling techniques are illustrated in details and a mining case study with around twenty thousand blocks over seventeen periods have been scheduled.

1. Introduction

Mixed integer linear programming mathematical optimization have been used by different researchers to tackle the long-term open-pit scheduling problem (2003; Ramazan and Dimitrakopoulos, 2004; Dagdelen and Kawahata, 2007). The MILP models theoretically have the capability to consider diverse mining constraints such as multiple ore processors, multiple material stockpiles, and blending strategies. The applications of MILP models result in production schedules generating near theoretical optimal net present values for mining ventures. The number of binary variables required in formulations presented by Caccetta and Hill (2003) and Ramazan and Dimitrakopoulos (2004) is equal to the number of blocks in the block model multiplied by the total number of scheduling periods. For a typical real size open pit scheduling problem number of blocks is in the order of couple of hundred thousands to millions and the number of scheduling periods usually varies between twenty to thirty years for a life-of-mine schedule. Evidently, a problem of this size is numerically intractable with current state of hardware and commercial optimization solvers. Ramazan and Dimitrakopoulos (2004) presented a method to reduce the number of

[1] Assistant Professor, Department of Mining & Nuclear Engineering, University of Missouri-Rolla, USA,

binary integer variables by setting waste blocks as linear variables. Setting waste blocks as linear variables will cause a block to be extracted in multiple periods, generating a schedule which is not feasible from practical equipment access point of view. Also, notable is work by Dagdelen, who applied the Lagrangian relaxation technique and sub-gradient methods to solve the mine production scheduling MILP problem (Dagdelen and Kawahata, 2007).

Boland et al. (2007) extended the formulation of Caccetta and Hill (2003) based on strict temporal sequence of blocks rather than assigning blocks to periods of extraction. Boland et al. (2007) reduced the number of decision variables by eliminating a number of variables presented in Caccetta and Hill (2003) formulation prior to optimization. This was achieved by combining the block precedence constraints with the production constraints, aggregated over a sequence of time periods. The numerical results illustrated a decrease in computational requirements to obtain the optimal integer solution. Boland et al. (2009) have demonstrated an iterative disaggregation approach to using a finer spatial resolution for processing decisions to be made based on the small blocks, while allowing the order of extraction decisions be made at an aggregate level. Boland et al. (2009) reported notable improvements on the convergence time of their algorithm. Boland et al. (2009) id not present enough information on their method of aggregation and assumed that some aggregation technique already exist.

We have used Boland et al. (2009) general formulation as our starting point of analysis. We divided the major decision variables into two categories, continuous variables representing the portion of a block that is going to be extracted in each period and binary integer variables controlling the order of extraction of blocks through a dependency graph using depth-first-search algorithm. We have implemented our new optimization formulation in TOMLAB/CPLEX environment (Holmström, 1989-2009). The models were verified and validated through synthetic data and a mining case study on an iron ore mine.

1.1. Economic block value modeling

Assumption: a general parameter f can take four indices in the format of $f_{k,n}^{e,t}$. Where:

$t \in \{1,....,T\}$	index for scheduling periods.
$k \in \{1,...,K\}$	index for mining-cuts.
$n \in \{1,...,N\}$	index for blocks.
$e \in \{1,...,E\}$	index for elements of interest in each block.

The objective functions of the MILP formulations are to maximize the net present value of the mining operation. Hence, we need to define a clear concept of economic block value based on ore parcels which could be mined selectively. The profit from mining a block depends on the value of that block and the costs incurred in mining and processing it. The cost of mining a block is a function of its spatial location, which characterizes how deep the block is located relative to the surface and how far it is relative to its final dump. The spatial factor can be applied as a mining cost adjustment factor for each block according to its location to the surface. The discounted profit from block n is equal to the discounted revenue generated by selling the final product contained in block n minus all the discounted costs involved in extracting block n, this is presented by Eqs. (1) and (2).

discounted profit = discounted revenue - discounted costs (1)

$$d_n^t = \underbrace{[\sum_{e=1}^{E} o_n \times g_n^e \times r^{e,t} \times (p^{e,t} - cs^{e,t})}_{discounted\ revenues} - \underbrace{\sum_{e=1}^{E} o_n \times cp^{e,t}] - [(o_n + w_n) \times cm^t]}_{discounted\ \cos ts} \tag{2}$$

Where

- d_n^t is the discounted profit generated by extracting block n in period t,
- o_n is the ore tonnage in block n and ore tonnage in mining-cut k,
- w_n is the waste tonnage in block n,
- g_n^e is the average grade of element e in ore portion of block n,
- $r^{e,t}$ is the processing recovery, which is the proportion of element e recovered in time period t,
- $p^{e,t}$ is the price in present value terms obtainable per unit of product (element e),
- $cs^{e,t}$ is the selling cost in present value terms per unit of product (element e),
- $cp^{e,t}$ is the extra cost in present value terms per tonne of ore for mining and processing,
- cm^t is the cost in present value terms of mining a tonne of waste in period t.

For simplification purposes we denote:

$$v_n^t = [\sum_{e=1}^{E} o_n \times g_n^e \times r^{e,t} \times (p^{e,t} - cs^{e,t}) - \sum_{e=1}^{E} o_n \times cp^{e,t}] \tag{3}$$

$$q_n^t = (o_n + w_n) \times cm^t \tag{4}$$

Where

- v_n^t is the discounted revenue generated by selling the final product within block n in period t minus the extra discounted cost of mining all the material in block n as ore and processing it; and
- q_n^t is the discounted cost of mining all the material in block n as waste.

2. Mixed integer linear programming model for open pit production scheduling

We present two different formulations for the open pit production scheduling problem, with the objective function to maximize the NPV of the mining operation. We extended our models based on concepts presented in Boland et al. (2009) as the starting point of our research.

2.1. Extraction at mining-cut level and processing at block level

In the proposed model processing is at block level and extraction is at mining-cut level. The amount of ore processed is controlled by the continuous variable x_n^t, and the amount of material mined is controlled by the continuous variable y_k^t. Using continuous decision variables allows fractional extraction of blocks in different periods. $b_k^t \in \{0,1\}$, is the binary integer variable controlling the precedence of extraction of mining- cuts. $b_k^t \in \{0,1\}$ is equal to one if extraction of mining-cut has started by or in period t, otherwise it is zero.

Objective function:

$$\max \sum_{t=1}^{T}\sum_{k=1}^{K}\left(\sum_{n\in c_k} v_n^t \times x_n^t - (\sum_{n\in c_k} q_n^t)\times y_k^t\right) \tag{5}$$

Where

- T is the maximum number of scheduling periods, where $\mathcal{T} = \{1,...,T\}$ is the set of all the scheduling time periods in the model,
- K is the total number of mining-cuts to be scheduled, where $\mathcal{K} = \{1,...,K\}$ is the set of all the mining-cuts in the model,
- c_k represents mining-cut k,
- $x_n^t \in [0,1]$ is a continuous decision variable, representing the portion of block n to be extracted as ore and processed in period t,
- $y_k^t \in [0,1]$ is a continuous decision variable, representing the portion of mining-cut c_k to be mined in period t, fraction of y characterizes both ore and waste included in the mining-cut.

It should be mentioned that in the objective function given by Eq. (5), mining is controlled at the mining-cut level, whereas the processing is at the higher resolution of block level. The objective function is subject to the following constraints.

Mining capacity constraints:

$$\sum_{k=1}^{K}\left(\sum_{n\in c_k}(o_n + w_n)\right)\times y_k^t \le mu^t \qquad \forall t \in \{1,...,T\} \tag{6}$$

$$\sum_{k=1}^{K}\left(\sum_{n\in c_k}(o_n + w_n)\right)\times y_k^t \ge ml^t \qquad \forall t \in \{1,...,T\} \tag{7}$$

Where

- mu^t is the upper bound on mining capacity in period t (tonnes),
- ml^t is the lower bound on mining capacity in period t (tonnes).

Eq. (6) controls that the total amount of ore and waste mined in each period to be within the targeted maximum mining capacity of equipment. The constraints are controlled by the continuous variable y_k^t at the mining-cut level. Eq. (7) controls the minimum amount of material that needs to be mined; Eq. (7) is useful in achieving a constant stripping ratio over the mine life. A production schedule with an invariable stripping ratio would have significant savings potential by ensuring that fleet size required is matched to targets for material movement. The decision of the proper production rate which leads to the boundaries on mining capacity is an important stage of the production scheduling of open pit mines. Different scenarios of annual ore production rates must be examined and the one with highest NPV and uniform mill feed must be chosen. The mining capacity boundaries are function of the ore reserve, overall stripping ratio, designed processing capacity, targeted mine-life, and the capital investment available for purchasing equipment. As it is illustrated by Eqs. (6) and (7), the upper and lower bounds of mining capacity could vary by scheduling periods, this flexibility allows the designer to target on replacing the fleet with different mining capacities at different stages of mine-life. The shortage of equipment in specific periods could be compensated with contract mining. Eqs. (6) and (7) will generate one constraints per period.

Processing capacity constraints:

$$\sum_{n=1}^{N} o_n \times x_n^t \le pu^t \qquad \forall t \in \{1,...,T\} \tag{8}$$

$$\sum_{n=1}^{N} o_n \times x_n^t \ge pl^t \qquad \forall t \in \{1,...,T\} \tag{9}$$

Where

- N is the number of blocks in the block model, where $\mathcal{N} = \{1,...,N\}$ is a set of all the blocks in the model,
- pu^t is the upper bound on processing capacity of ore in period t (tonnes),
- pl^t is the lower bound on processing capacity of ore in period t (tonnes).

Eqs. (8) and (9) represent inequality constraints controlling the mill feed or processing capacity; These constraints assist the mine planners in achieving an overall mine-to-mill integration by providing a uniform feed throughout the mine-life. Constraints (8) and (9) are at block level, which means the decisions are made based upon the tonnage of ore above the cut-off grade within individual blocks. In practice, the processing capacity constraints must be set within a tight upper and lower bounds to provide a uniform feed to the mill. Depending on the shape of the orebody and distribution of ore grades in the orebody, these constraints could not be honored under some circumstances, which will lead to an infeasible problem. Pre-stripping could be achieved by setting the upper and lower bounds of processing capacity constraints equal to zero for the desired periods; this would enforce the optimizer to only mine waste blocks in the early periods that would open up the orebody for later mining. Eqs. (8) and (9) will generate one constraints per period per ore type.

Grade blending constraints:

$$\sum_{n=1}^{N} g_{n}^{e} \times o_n \times x_n^t \Big/ \sum_{n=1}^{N} o_n \times x_n^t \le gu^{e,t} \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{10}$$

$$\sum_{n=1}^{N} g_{n}^{e} \times o_n \times x_n^t \Big/ \sum_{n=1}^{N} o_n \times x_n^t \ge gl^{e,t} \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{11}$$

Where

- g_n^e is the average grade of element e in ore portion of block n, where $\mathcal{E} = \{1,...,E\}$ is the set of all the elements of interest in the model,
- $gu^{e,t}$, is the upper bound on acceptable average head grade of element e in period t,
- $gl^{e,t}$, is the lower bound on acceptable average head grade of element e in period t.

Production scheduling is concerned with the inherent task of blending the run-of-mine materials before processing. The objective is to mine in such a way that the resulting mix meets the quality specifications of the processing plant. The blending problem becomes more important as the design moves towards mid-range to short-range planning, where the planner is concerned with reducing the grade variability. Constraints (10) and (11) are at block level and there would be one equation per element per scheduling period for upper and lower bound.

Ore processed and material mined constraints:

$$x_n^t \le y_{k,n}^t \qquad \forall n \in \{1,...,N\}, \quad n \in c_k, \quad t \in \{1,...,T\} \tag{12}$$

Where x_n^t is the portion of block n to be extracted as ore and processed in period t, and $y_{k,n}^t$ is representing the portion of mining-cut c_k to be mined in period t, fraction of y characterizes both ore and waste included in the mining-cut. Eq. (12) demonstrates inequalities that ensure the amount of ore of any block which is processed in any given period is less than or equal to the amount of rock extracted from the mining-cut that the block belongs to in any given scheduling period. A very important assumption in the formulation is that each mining-cut is extracted homogeneously; this means that $y_{k,n}^t$ illustrates the fraction of mining-cut k to be extracted in time period t, all the blocks within the cut, $n \in c_k$ are extracted with the same proportion of $y_{k,n}^t$. This assumption generates production schedules that mimic the real mining operation in the sense that it would minimize the jumping movement of the equipment from one point to another. Eq. (12) generates one equation per block per period.

Precedence of mining-cuts extraction and slope constraints

$$b_k^t - \sum_{i=1}^{t} y_s^i \le 0 \qquad \forall k \in \{1,...,K\}, \quad t \in \{1,...,T\}, \quad s \in H(S) \tag{13}$$

$$\sum_{i=1}^{t} y_k^i - b_k^t \le 0 \qquad \forall k \in \{1,...,K\}, \quad t \in \{1,...,T\} \tag{14}$$

$$b_k^t - b_k^{t+1} \le 0 \qquad \forall k \in \{1,...,K\}, \quad t \in \{1,...,T-1\} \qquad (15)$$

Where

- $b_k^t \in \{0,1\}$ is a binary integer decision variable controlling the precedence of extraction of mining- cuts. b_k^t is equal to one if extraction of mining-cut c_k has started by or in period t, otherwise it is zero,
- $H(S)$ is a set $H(S) \subset \mathcal{K}$ for each mining-cut c_k, defining the immediate predecessor cuts that must be extracted prior to extracting mining-cut k, where S is the total number of cuts in set $H(S)$.

For each mining-cut k Eqs. (13) to (15) check the set of immediate predecessor cuts that must be extracted prior to mining-cut k. This precedence relationship ensures that all the blocks above the current mining-cut are extracted prior to extraction of mining-cut. As it could be deduced from Eq. (15), the formulation is based on the temporal sequence of extraction rather than checking for all the periods. For Eqs. (13) to (15), there would be one equation per mining-cut per period.

For each block n there is a set $C(L) \subset \mathcal{N}$, which includes all the blocks that must be extracted prior to mining block n to ensure that block n is exposed for mining with the desired overall pit slopes, where J is the total number of blocks in set $C(L)$. We will use a directed graph to model the precedence of extraction between blocks. We defined a directed graph $G_b(\mathcal{N}, \mathcal{A})$ by the set of vertices, $\mathcal{N}$ (blocks); connected by ordered pairs of elements called arcs, $\mathcal{A}$.

During the clustering of blocks into mining-cuts another directed graph at mining-cut level is constructed, which captures the precedence relationship of mining-cuts. This directed graph is denoted by $G_c(\mathcal{K}, \mathcal{B})$ where $\mathcal{B} = \{1,...,B\}$ is the set of all edges in the mining-cuts precedence directed graph. The directed graph $G_c(\mathcal{K}, \mathcal{B})$ is constructed in a way that while satisfying the order of extraction at mining-cut level, it would also satisfy the relationships defined by the graph $G_b(\mathcal{N}, \mathcal{A})$ at block level. This approach of defining two directed graphs at mining-cut and block level enables us to model variable pit slopes with small acceptable slope errors in the different regions of the open pit. In other words, mining is controlled at the mining-cut level but the slopes are modeled at the block level.

2.2. Alternative MILP formulation

Eqs. (5) to (15) represent the MILP formulation for long-term open pit production scheduling. The proposed formulation requires $(2 \times K + N) \times T$ number of decision variables, where $K \times T$ of these variables are binary integers. One of the major obstacles in using the MILP formulations for mine production scheduling is the sheer size of the problem. The number of blocks, N, in the model is usually between tens of thousands to millions; the numerous number of blocks within the model will lead to a formulation with an objective function with many variables. Moreover, the main physical constraint in open pit mining is the relationship of precedence of extraction of blocks modeled by binary integer variables. The most common difficulty with MILPs is size of the branch and cut tree; the tree becomes so large that insufficient memory remains to solve an LP sub-problem. The number of binary integer variables in the formulations determines the size of the branch and cut tree. As a general strategy in our formulations we aimed at reducing the number of binary integer variables, we also focused on developing formulations that will mainly use continuous optimization techniques rather than discrete optimization. We have reduced the number of the binary integer variables to $K \times T$, where to some extent we have control over the number of mining-cuts K , during the clustering process.

We investigated the effect of using continuous decision variables (x_n^t *and* y_k^t), which leads to fractional block extraction on the quality and practicality of the generated schedules. There is a possibility that block $n \in c_k$ get extracted over multiple periods. Our computational experiments on different orebody models using the formulation presented in section 2.1 revealed that the blocks' fractions are usually scheduled over consecutive periods and in the worst case examined, some blocks were extracted over three periods. We should also emphasize again that blocks are uniformly extracted as part of mining-cuts that means in the worst case observed a mining-cut is extracted over three periods, which is not impractical from mining point of view. The total tonnage of ore processed in the MILP formulation presented in section 2.1 is related to how mining and processing capacities are set in accordance with the ore reserve total tonnage. There is the possibility that quantities of ore above the cut-off grade would not get processed due to the processing capacity limitations; it is feasible to overcome the abovementioned problems by adding reserve and maximum number of fractions constraints to the MILP formulation presented in section 2.1.

Maximum number of fractions and reserve constraints

$$\sum_{t=1}^{T} x_n^t = 1 \qquad \forall n \in \{1,...,N\} \qquad (16)$$

$$\sum_{t=1}^{T} y_k^t = 1 \qquad \forall k \in \{1,...,K\} \qquad (17)$$

$$\sum_{t=1}^{T} u_k^t \le m \qquad \forall k \in \{1,...,K\} \qquad (18)$$

$$\sum_{t=1}^{T} u_k^t \times y_k^t = 1 \qquad \forall k \in \{1,...,K\} \qquad (19)$$

Where

- $u_k^t \in \{0,1\}$ is a binary integer decision variable equal to one if mining-cut c_k is scheduled to be extracted in period t , otherwise zero,
- m is an integer number representing the maximum number of fractions that mining-cuts are allowed to be extracted over.

Equality constraints presented by Eq. (16) would ensure that all the ore within the predefined pit limits or the targeted push-back would be processed during the scheduling. Eq. (16) adds one constraint per block. Eq. (17) would ensure that all the material within the predefined pit outline is going to be mined; this would add one constraint per mining-cut. Eq. (18) and (19) guarantee that the maximum number of fractions of mining-cuts in the solution for y_k^t is not going to exceed m. For large-scale models with many numbers of scheduling periods m could be set equal to two or three fractions maximum. Eq. (19) introduces non-linear constraints into the MILP formulation. The modified model by adding Eqs. (16) to (19) even after linearization, would frame a more complicated MILP formulation comparing to the MILP formulations without Eqs. (16) to (19).

2.3. Extraction and processing at mining-cut level

The formulation of the MILP presented in sections 2.1 and 2.2 has reduced the required numbers of binary integer variables for controlling the precedence relationships drastically. Nevertheless, for large-scale production scheduling models with millions of blocks and decades of mine life the size of the LP sub-problems represented by the x_n^t continuous decision variables would be intractable with the current state of the optimization technology. To be able to overcome this obstacle, we present an MILP formulation which both mining and processing are at the mining-cut level. This approach makes it possible to formulate a tractable MILP model for even very large-scale open pit mines with millions of blocks over decades of mine life. We introduce a continuous processing decision variable at mining-cut level. The blocks are aggregated prior to optimization into mining-cuts with clustering algorithms. the MILP formulation of the model is as follows:

Objective function:

$$\max \sum_{t=1}^{T}\sum_{k=1}^{K}(v_k^t \times s_k^t - q_k^t \times y_k^t) \tag{20}$$

Subject to:

$$gl^{t,e} \le \sum_{k=1}^{K} g_k^e \times o_k \times s_k^t \Big/ \sum_{k=1}^{K} o_k \times s_k^t \le gu^{t,e} \qquad \forall t \in \{1,...,T\},\quad e \in \{1,...,E\} \tag{21}$$

$$pl^t \le \sum_{k=1}^{K} o_k \times s_k^t \le pu^t \qquad \forall t \in \{1,...,T\},\quad e \in \{1,...,E\} \tag{22}$$

$$ml^t \le \sum_{k=1}^{K} (o_k + w_k) \times y_k^t \le mu^t \qquad \forall t \in \{1,...,T\} \tag{23}$$

$$s_k^t \le y_k^t \qquad \forall k \in \{1,...,K\},\quad t \in \{1,...,T\} \tag{24}$$

Eqs. (13) to (15).

Where

- $s_k^t \in [0,1]$ is a continuous variable, representing the portion of mining-cut c_k to be extracted as ore and processed in period t,
- o_k is the ore tonnage in mining-cut k,
- w_k is the waste in mining-cut k.

Constraints similar to what is presented by Eqs. (16) to (19) for reserve constraints and maximum number of mining-cuts is optional in this formulation as well.

3. Numerical modeling

In most linear optimization problems, the variables of the objective function are continuous in the mathematical sense, with no gaps between real values. To solve such linear programming problems, ILOG CPLEX implements optimizers based on the simplex algorithms (Winston, 1995) (both primal and dual simplex) as well as primal-dual logarithmic barrier algorithms.

Branch and cut is a method of combinatorial optimization for solving integer linear programs. The method is a hybrid of branch and bound and cutting plane methods (Horst and Hoang, 1996). Refer to Wolsey (1998) for a detailed explanation of branch and cut algorithm, including cutting planes. In recent years there has been significant improvements in mathematical programming optimizers such as ILOG CPLEX (Bixby, 1987-2009). This optimizer uses branch and cut techniques to solve MILP models and it is closing the gap between theory and practice in optimization of large-scale industrial problems. In this study we used TOMLAB/CPLEX version 11.2 (Holmström, 1989-2009) as the MILP solver. TOMLAB/CPLEX efficiently integrates the solver package CPLEX (ILOG Inc, 2007) with MATLAB environment (MathWorks Inc., 2007).

An important termination criterion that the user can set explicitly in CPLEX is the MILP gap tolerance. We have used the relative MILP gap tolerance, which indicates to CPLEX to stop when an integer feasible solution has been proved to be within the gap% of optimality.

3.1. General formulation

The general formulation in TOMLAB for a mixed integer linear programming problem is in the form of:

$$\min_{z} f(z) = \mathbf{c}^T.\mathbf{z} \tag{25}$$

Subject to:

$$\mathbf{z}_l \leq \mathbf{z} \leq \mathbf{z}_u \tag{26}$$

$$\mathbf{b}_l \leq \mathbf{A}.\mathbf{z} \leq \mathbf{b}_u \tag{27}$$

Where

- $\mathbf{c}$ is a $j \times 1$ vector, the objective function coefficient, and $\mathbf{c}^T$ represents the transpose of $\mathbf{c}$.
- $\mathbf{z}$ is a $j \times 1$ vector, elements of vector $\mathbf{z}$ are the decision variables of the MILP formulation.
- $\mathbf{z}_l$ *and* $\mathbf{z}_u$ are $j \times 1$ vectors, defining the lower and upper bounds on the decision variables.
- $\mathbf{A}$: is a $i \times j$ coefficient matrix, representing the constraints of the MILP formulations.

- $\mathbf{b}_l$ *and* $\mathbf{b}_u$: are $j \times 1$ vectors, defining the lower and upper boundary conditions. Equality constraints are defined by setting the lower bounds equal to the upper bounds, $\mathbf{b}_l = \mathbf{b}_u$.

3.2. Objective function

We will formulate the general MILP model presented in section 2.1 with the maximum number of fractions and reserve constraints demonstrated by Eqs. (16) to (19). The objective of open pit production scheduling problem given by Eq. (5) is to maximize the NPV of the operation. The general format of the MILP formulation in TOMLAB given by Eq. (25) is to minimize the objective function. Therefore the objective function coefficient vector, $\mathbf{c}$, which is a $(4K+N)T \times 1$ vector given by Eq. (28) should be multiplied by a negative sign, as the result Eq. (25) would change to $\min_{z} f(z) = -\mathbf{c}^T.\mathbf{z}$. To simplify the notation, we will use the vertical matrix concatenation operator, ';'. This operator constructs a matrix or vector by concatenating the matrices or vectors along the vertical dimension of the matrix or vector. The objective function coefficient vector, $\mathbf{c}$ is a vector of size $(4K+N)T \times 1$ given by Eq. (28).

$$\mathbf{c}^{(4K+N)T\times 1} = \left[\mathbf{v};\ \mathbf{q};\ \mathbf{0};\ \mathbf{0};\ \mathbf{0}\right] \tag{28}$$

Where

- $\mathbf{v}$ is an $NT \times 1$ vector holding the discounted values defined by Eq. (3) where N is the maximum number of blocks in the model and T is the number of scheduling periods,
- $\mathbf{q}$ is a $KT \times 1$ vector holding the discounted mining costs defined by Eq. (4) where K is the number of mining-cuts in the model and T is the number of scheduling periods,
- $\mathbf{0}$ is $KT \times 1$ zero vector with all elements equal to zero,

The coefficients in the objective function and in the constraints matrix have different units and dissimilar order of magnitude; it is necessary to transform the objective function and the constraints coefficient matrix to unitless matrices and vectors. To do so we normalize the vectors and matrices by dividing them by a norm of its multipliers vectors. Let us define $\overline{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$ and $\overline{\mathbf{q}} = \mathbf{q}/\|\mathbf{q}\|$, where $\|\mathbf{q}\|$ and $\|\mathbf{v}\|$ are norm of $\mathbf{q}$ and $\mathbf{v}$. Therefore, the coefficient vector is going to be in the form of $\mathbf{c} = \left[\overline{\mathbf{v}};\ \overline{\mathbf{q}};\ \mathbf{0};\ \mathbf{0};\ \mathbf{0}\right]$. We will use the notation '$\overline{\ }$' to illustrate a normalized vector in the rest of the paper.

Eq. (29) illustrates how the decision variables vector, $\mathbf{z}$ is constructed. $\mathbf{z}$ is a $(4K+N)T \times 1$ vector.

$$\mathbf{z}^{(4K+N)T\times 1} = [\mathbf{x};\ \mathbf{y};\ \mathbf{b};\ \mathbf{u};\ \mathbf{uy}] \tag{29}$$

Where

- $\mathbf{x}$ is an $NT \times 1$ vector with the continuous decision variables, $x_n^t \in [0,1]$ as elements, representing the portion of block n to be extracted as ore and processed in period t,
- $\mathbf{y}$ is a $KT \times 1$ vector with the continuous decision variables, $y_k^t \in [0,1]$ as elements, representing the portion of mining-cut c_k to be mined in period t,
- $\mathbf{b}$ is a $KT \times 1$ vector holding the binary integer decision variables, $b_k^t \in \{0,1\}$, these decision variables control the precedence of extraction of mining- cuts,
- $\mathbf{u}$ is $KT \times 1$ vector holding the binary integer variables $u_k^t \in \{0,1\}$; u_k^t is equal to one if mining-cut c_k is scheduled to be extracted in period t, otherwise zero, u_k^t is defined in Eq. (18),
- $\mathbf{uy}$ is $KT \times 1$ vector holding the continuous variables $uy_k^t \in \{0,1\}$, defining the outcome of $u_k^t \times y_k^t$ defined by Eq. (19).

3.3. Constraints

In this section, we will develop the numerical models for the equality and inequality constraints represented by Eqs. (6) to (19).

Mining capacity constraints:

Eqs. (6) and (7) represents the mining capacity constraints, the numerical model is represented by Eq. (30), where $\mathbf{A}_1$ is a $2T \times (4K+N)T$ coefficient matrix and $\mathbf{b}_1$ is a $2T \times 1$ boundary condition vector.

$$\mathbf{A}_1.\mathbf{z} \le \mathbf{b}_1 \tag{30}$$

$$\mathbf{A}_1^{2T\times(4K+N)T} = \begin{bmatrix} \mathbf{0}_1 & \overline{\mathbf{A}}_m & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_1 & -\overline{\mathbf{A}}_m & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \end{bmatrix} \tag{31}$$

$$\mathbf{b}_1^{2T\times1} = \begin{bmatrix} \overline{\mathbf{m}}_u \\ -\overline{\mathbf{m}}_l \end{bmatrix} \tag{32}$$

Where

- $\mathbf{A}_m$ is a $T \times KT$ matrix with elements holding the total tonnage of material in each mining-cut in each period,
- $\mathbf{m}_u$ is a $T \times 1$ vector of mining capacity upper bounds as defined in Eq. (6),
- $\mathbf{m}_l$ is a $T \times 1$ vector of mining capacity lower bounds as defined in Eq. (7),
- $\mathbf{0}_1$ is a $T \times NT$ zero matrix,
- $\mathbf{0}_2$ is a $T \times KT$ zero matrix,

Processing capacity constraints:

Eqs. (8) and (9) represents the processing capacity constraints, the numerical model is represented by Eq. (33), where $\mathbf{A}_2$ is a $2T \times (4K+N)T$ coefficient matrix and $\mathbf{b}_2$ is a $2T \times 1$ boundary condition vector.

$$\mathbf{A}_2.\mathbf{z} \le \mathbf{b}_2 \tag{33}$$

$$\mathbf{A}_2^{2T\times(4K+N)T} = \begin{bmatrix} \overline{\mathbf{A}}_p & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ -\overline{\mathbf{A}}_p & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \end{bmatrix} \tag{34}$$

$$\mathbf{b}_2^{2T\times1} = \begin{bmatrix} \overline{\mathbf{p}}_u \\ -\overline{\mathbf{p}}_l \end{bmatrix} \tag{35}$$

Where

- $\mathbf{A}_p$ is a $T \times NT$ matrix with elements holding the total tonnage of ore in each block in each period,
- $\mathbf{p}_u$ is a $T \times 1$ vector of processing capacity upper bounds as defined in Eq. (8),
- $\mathbf{p}_l$ is a $T \times 1$ vector of processing capacity lower bounds as defined in Eq. (9).

Grade blending constraints:

Eqs. (10) and (11) represent the grade blending constraints, the numerical model is represented by Eq. (36), where $\mathbf{A}_3$ is a $2ET \times (4K+N)T$ coefficient matrix and $\mathbf{b}_3$ is a $2ET \times 1$ boundary condition vector.

$$\mathbf{A}_3.\mathbf{z} \le \mathbf{b}_3 \tag{36}$$

$$\mathbf{A}_3^{2ET\times(4K+N)T} = \begin{bmatrix} \overline{\mathbf{A}}_g & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\overline{\mathbf{A}}_g & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{37}$$

$$\mathbf{b}_3^{2ET\times1} = \begin{bmatrix} \overline{\mathbf{g}}_u \\ -\overline{\mathbf{g}}_l \end{bmatrix} \tag{38}$$

Where

- $\mathbf{A}_g$ is an $ET \times NT$ matrix of average grade of each element of interest in each block in each period,
- $\mathbf{g}_u$ is an $ET \times 1$ vector of average grade upper bounds on acceptable average head grade of the elements of interest as defined in Eq. (10),
- $\mathbf{g}_l$ is a $ET \times 1$ vector of average grade lower bounds on acceptable average head grade of the elements of interest as defined in Eq. (11),
- $\mathbf{0}_3$ is an $ET \times KT$ zero matrix.

Ore processed and material mined constraints:

Eq. (12) represents the ore processed and material mined constraints, the numerical model is represented by Eq. (39), where $\mathbf{A}_4$ is a $NT \times (4K+N)T$ coefficient matrix and $\mathbf{b}_4$ is an $NT \times 1$ zero boundary condition vector. Eq. (39) inequalities ensure that the amount of ore of any block which is processed in any given period is less than or equal to the amount of rock extracted from the mining-cut that the block belongs to in any given scheduling period.

$$\mathbf{A}_4.\mathbf{z} \le \mathbf{b}_4 \tag{39}$$

$$\mathbf{A}_4^{NT\times(4K+N)T} = \begin{bmatrix} \mathbf{A}_x & \mathbf{A}_y & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 \end{bmatrix} \tag{40}$$

Where

- $\mathbf{A}_x$ is a $NT \times NT$ matrix with an element of 1 for each block in each period,
- $\mathbf{A}_y$ is a $NT \times KT$ matrix with an element of -1 per mining-cut for each block in each period,
- $\mathbf{0}_4$ is a $NT \times KT$ zero matrix.

Precedence of mining-cuts extraction and slope constraints

Eqs. (13) to (15) represent the precedence of mining-cuts extraction and slope constraints, the numerical model is represented by Eq. (41). We will present the construction of slope constraints matrix, Eq. (41), with an illustrative example.

$$\mathbf{A}_5.\mathbf{z} \le \mathbf{b}_5 \tag{41}$$

Let's consider a set of mining-cuts to be scheduled. For the sake of discussion we assume that the model consist of five mining-cuts (Fig. 1); the immediate predecessor cuts are labeled with directed-arcs pointing from the parent to the child node. A directed graph constructs the precedence relationship between mining-cuts, the directed graph tags the mining-cuts that must be extracted prior to extracting each mining-cut k. This set is denoted by $H(S) \subset \mathcal{K}$, where S is the total number of mining-cuts in H(S).

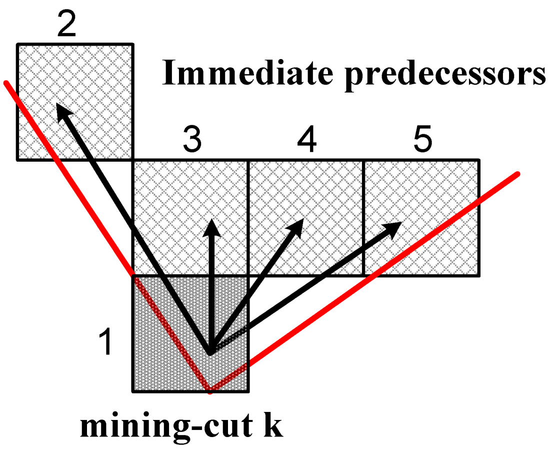


Fig. 1. Schematic view of a mining cut and its predecessors cuts.

We start with constructing the required matrices defining Eq. (13). We assume that the five mining-cuts ($K = 5$) illustrated in Fig. 1, are scheduled to be extracted over four periods ($T = 4$). The immediate predecessors' set $H(S) = \{2,3,4,5\}$; this set represents the mining-cuts that must be extracted prior to extraction of mining-cut labeled as one.

We define $\mathbf{b}1_{vec}$ and $\mathbf{y}1_{vec}$ as $1 \times K$ vectors in Eqs. (42) and (43), these vectors are subcomponents used in assembling the matrices required to model Eq. (13). $\mathbf{b}1_{vec}$ and $\mathbf{y}1_{vec}$ are used to assemble matrices $\mathbf{b}1_{mat}$ and $\mathbf{y}1_{mat}$ as presented in Eqs. (44) and (45) for each mining-cut; where $\mathbf{0}_5$ is a $1 \times K$ vector of zeros. Then, matrices $\mathbf{A}_{sy1}$ and $\mathbf{A}_{sb1}$, which are $KT \times KT$ matrices are constructed for all the mining-cuts in the model, where $\mathcal{K} = \{1,...,K\}$, this concatenation is demonstrated by Eqs. (46) and (47).

$$\mathbf{b}1_{vec}^{1\times K} = [1 \quad 0 \quad 0 \quad 0 \quad 0] \tag{42}$$

$$\mathbf{y}1_{vec}^{1\times K} = [0 \quad -1 \quad -1 \quad -1 \quad -1] \tag{43}$$

$$\mathbf{b}1_{mat}^{T\times KT} = \begin{pmatrix} \mathbf{b}1_{vec} & \cdots & \mathbf{0}_5 \\ \vdots & \ddots & \vdots \\ \mathbf{0}_5 & \cdots & \mathbf{b}1_{vec} \end{pmatrix} \tag{44}$$

$$\mathbf{y}1_{mat}^{T\times KT} = \begin{pmatrix} \mathbf{y}1_{vec} & \cdots & \mathbf{0}_5 \\ \vdots & \ddots & \vdots \\ \mathbf{y}1_{vec} & \cdots & \mathbf{y}1_{vec} \end{pmatrix} \tag{45}$$

$$\mathbf{A}_{sy1}^{KT\times KT} = \left(\mathbf{y}1_{mat1}; \quad \mathbf{y}1_{mat2}; \; \cdots\cdots \; ; \mathbf{y}1_{matK}\right) \tag{46}$$

$$\mathbf{A}_{sb1}^{KT\times KT} = \left(\mathbf{b}1_{mat1}; \quad \mathbf{b}1_{mat2}; \; \cdots\cdots \; ; \mathbf{b}1_{matK}\right) \tag{47}$$

Subsequently, we will construct the matrices required to capture Eq. (14). We define $\mathbf{b}2_{vec}$ and $\mathbf{y}2_{vec}$ as $1 \times K$ vectors in Eqs. (48) and (49), these vectors are subcomponents used in assembling the matrices required to model Eq. (14). $\mathbf{b}2_{vec}$ and $\mathbf{y}2_{vec}$ vectors are used to assemble matrices $\mathbf{b}2_{mat}$ and $\mathbf{y}2_{mat}$ as presented in Eqs. (50) and (51), for each mining-cut. Afterward, matrices $\mathbf{A}_{sy2}$ and $\mathbf{A}_{sb2}$, which are $KT \times KT$ matrices are constructed for all the mining-cuts in the model, this concatenation is demonstrated by Eqs. (52) and (53).

$$\mathbf{b}2_{vec}^{1\times K} = [-1 \quad 0 \quad 0 \quad 0 \quad 0] \tag{48}$$

$$\mathbf{y}2_{vec}^{1\times K} = [1 \quad 0 \quad 0 \quad 0 \quad 0] \tag{49}$$

$$\mathbf{b}2_{mat}^{T\times KT} = \begin{pmatrix} \mathbf{b}2_{vec} & \cdots & \mathbf{0}_5 \\ \vdots & \ddots & \vdots \\ \mathbf{0}_5 & \cdots & \mathbf{b}2_{vec} \end{pmatrix} \tag{50}$$

$$\mathbf{y2}_{mat}^{T\times KT} = \begin{pmatrix} \mathbf{y}2_{vec} & \cdots & \mathbf{0}_5 \\ \vdots & \ddots & \vdots \\ \mathbf{y}2_{vec} & \cdots & \mathbf{y}2_{vec} \end{pmatrix} \tag{51}$$

$$\mathbf{A}_{sy2}^{KT\times KT} = \left(\mathbf{y}2_{mat1};\quad \mathbf{y}2_{mat2};\ \cdots\cdots\ ;\quad \mathbf{y}2_{matK}\right) \tag{52}$$

$$\mathbf{A}_{sb2}^{KT\times KT} = \left(\mathbf{b}2_{mat1};\quad \mathbf{b}2_{mat2};\ \cdots\cdots\ ;\quad \mathbf{b}2_{matK}\right) \tag{53}$$

Next, we will construct the matrices required to capture Eq. (15). We define $\mathbf{b}3_{vec}$ and $\mathbf{b4}_{vec}$ as $1\times K$ vectors in Eqs. (54) and (55), these vectors are subcomponents used in assembling the matrices required to model Eq. (15). $\mathbf{b}3_{vec}$ and $\mathbf{b4}_{vec}$ vectors are used to assemble matrix $\mathbf{b}3_{mat}$ as presented in Eq. (56), for each mining-cut. Matrix $\mathbf{A}_{sb3}$, which is $K(T-1)\times KT$ matrix is constructed for all the mining-cuts in the model, this concatenation is demonstrated by Eq. (57).

$$\mathbf{b}3_{vec}^{1\times K} = [1 \quad 0 \quad 0 \quad 0 \quad 0] \tag{54}$$

$$\mathbf{b}4_{vec}^{1\times K} = [-1 \quad 0 \quad 0 \quad 0 \quad 0] \tag{55}$$

$$\mathbf{b}3_{mat}^{(T-1)\times KT} = \begin{pmatrix} \mathbf{b3}_{vec} & \mathbf{b}4_{vec} \cdots & \mathbf{0}_5 \\ \vdots \ \ddots & \ddots & \vdots \\ \mathbf{0}_5 \ \ \ldots & \mathbf{b3}_{vec} & \mathbf{b}4_{vec} \end{pmatrix} \tag{56}$$

$$\mathbf{A}_{sb3}^{K(T-1)\times KT} = \left(\mathbf{b}3_{mat1};\quad \mathbf{b}3_{mat2};\ \cdots\cdots\ ;\quad \mathbf{b}3_{matK}\right) \tag{57}$$

Now we can construct the matrix $\mathbf{A}_5$ in Eq. (58), the inequality constraints is represented in Eq.(41), where

- $\mathbf{A}_5$ is a $[2KT + K(T-1)]\times(4K+N)T$ coefficient matrix,
- $\mathbf{b}_5$ is a $[2KT + K(T-1)]\times 1$ zero boundary condition vector,
- $\mathbf{0}_6$ is a $KT\times NT$ zero matrix,
- $\mathbf{0}_7$ is a $KT\times KT$ zero matrix,
- $\mathbf{0}_8$ is a $K(T-1)\times NT$ zero matrix,
- $\mathbf{0}_9$ is a $K(T-1)\times KT$ zero matrix.

$$\mathbf{A}_5^{[2KT+K(T-1)]\times(4K+N)T} = \begin{bmatrix} \mathbf{0}_6 & \mathbf{A}_{sy1} & \mathbf{A}_{sb1} & \mathbf{0}_7 & \mathbf{0}_7 \\ \mathbf{0}_6 & \mathbf{A}_{sy2} & \mathbf{A}_{sb2} & \mathbf{0}_7 & \mathbf{0}_7 \\ \mathbf{0}_8 & \mathbf{0}_9 & \mathbf{A}_{sb3} & \mathbf{0}_9 & \mathbf{0}_9 \end{bmatrix} \tag{58}$$

Maximum number of fractions and reserve constraints

The numerical model for the maximum number of mining fractions and reserve constraints are represented by Eqs. (16) to (19), where $\mathbf{A}_6$ is a $(N+3K)\times(4K+N)T$ coefficient matrix and $\mathbf{b}_6$ is an $(N+3K)\times 1$ boundary condition vector.

$$\mathbf{A}_6.\mathbf{z} \le \mathbf{b}_6 \tag{59}$$

$$\mathbf{A}_6^{(N+3K)\times(4K+N)T} = \begin{bmatrix} \mathbf{A}_{rx} & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_1 & \mathbf{A}_{ry} & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_1 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{A}_u & \mathbf{0}_2 \\ \mathbf{0}_1 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{A}_{uy} \end{bmatrix} \tag{60}$$

$$\mathbf{b}_6^{(N+3K)\times 1} = \left[\ \mathbf{1}_1; \quad \mathbf{1}_2; \quad \mathbf{m}; \quad \mathbf{1}_2 \right] \tag{61}$$

Where

- $\mathbf{1}_1$ is a $N\times 1$ vector with all elements equal to one.
- $\mathbf{1}_2$ is a $K\times 1$ vector with all elements equal to one.
- $\mathbf{A}_{rx}$ is a $N\times NT$ matrix with elements of one for each block in each period, these equality constraints add one constraints per block as defined by Eq. (16).
- $\mathbf{A}_{ry}$ is a $K\times KT$ matrix with elements of one for each mining-cut in each period, these equality constraints add one constraints per mining-cut as defined by Eq. (17).
- $\mathbf{A}_u$ is a $K\times KT$ matrix with elements of one for the periods that a mining-cut is scheduled to be extracted as defined by Eq. (18).
- $\mathbf{m}$ is a $K\times 1$ vector with elements equal to the number of maximum fractions that one mining-cut is allowed to be scheduled as defined by Eq. (18).
- $\mathbf{A}_{uy}$ is a $K\times KT$ matrix.

The equality constraints, in Eqs. (16), (17) and (19) are defined by setting the lower bounds equal to the upper bounds. Finally, we concatenate all the matrices and vectors representing the constraints and bounds into the coefficient matrix, $\mathbf{A}$ with the size of $[(3K+2E+N+4)T+N+2K]\times[(4K+N)T]$ and one boundary condition vector, **b,** with the size of $(4K+N)T\times 1$. The concatenation is represented by Eqs. (62) and (63).

$$\mathbf{A}^{[(3K+2E+N+4)T+N+2K]\times[(4K+N)T]} = \left(\mathbf{A}_1; \quad \mathbf{A}_2; \quad \mathbf{A}_3; \quad \mathbf{A}_4; \quad \mathbf{A}_5; \quad \mathbf{A}_6\right) \tag{62}$$

$$\mathbf{b}^{(4K+N)T\times 1} = \left(\mathbf{b}_1; \quad \mathbf{b}_2; \quad \mathbf{b}_3; \quad \mathbf{b}_4; \quad \mathbf{b}_5; \quad \mathbf{b}_6\right) \tag{63}$$

4. Numerical experiments and mining case study

A case study of scheduling an iron ore deposit was carried out to verify and validate the models. The total number of blocks within the final pit limit is 19,492. We used the fuzzy logic clustering algorithm to aggregate the blocks into 599 mining-cuts. Fig. 2 illustrates aggregating blocks into mining-cuts using fuzzy logic clustering, blocks are spatially grouped together based on rock-type and grade distribution, units in meters. Three types of ore; top magnetite, oxide, and bottom magnetite are classified in the deposit. The block model contains the estimated magnetic weight recovery (MWT%) of iron ore and the contaminants are phosphor (P%) and sulphur (S%). The blocks in the geological model represent a volume of rock equal to $25m \times 25m \times 15m$.

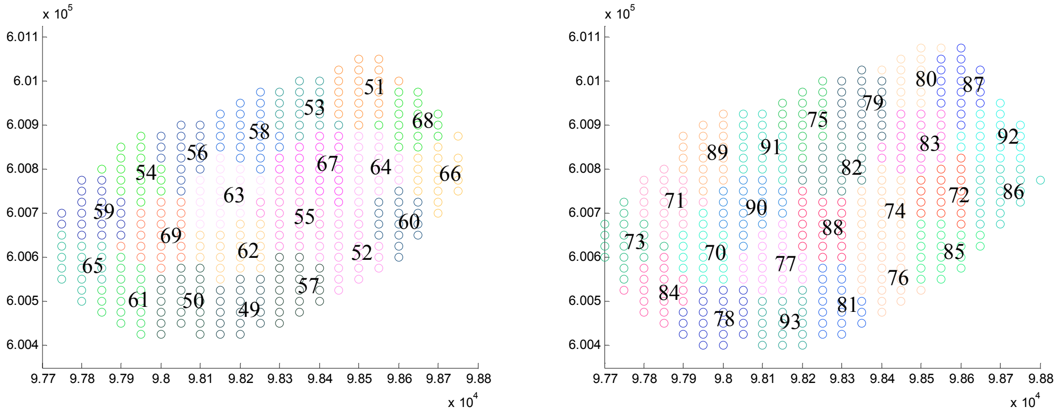


Fig. 2 Aggregating blocks into mining-cuts using fuzzy logic clustering, blocks are spatially grouped together based on rock-type and grade distribution, units in meters.

Table 1 summarizes the information related to the case study. The pit includes 427.33 Mt of rock where 116.29 Mt is ore with an average magnetic weight recovery of grade of 72.9%. Initially a capacity of 30 Mt /year was considered as the upper bound of mining. The objective function aimed to maximize the net present value with a discount rate of 10% per period. TOMLAB/CPLEX was used for implementation and solving the MILP formulation. 19,492 blocks were scheduled over 17 periods this made a coefficient matrix, A of a size $A(63,474 \times 30,549)$ with 510,094 nonzero elements. The CPLEX solver was set to find a solution within 3% gap of the theoretical optimal solution.

Table 2 illustrates the settings for two optimization runs, in the first one we used the default CPLEX settings. The upper and lower bound for MWT, S, and P are defined. A ten Mt/yr processing capacity was set for this case. In the second test we used four years of pre-stripping with setting the upper bound of the processing plant equal to zero. The processing capacity was ramped up gradually to ten Mt/yr by year nine.

Table 1. Final pit and production scheduling information.

Description	Value	Description	Value
Number of blocks	19,492	Minimum mining width (m)	150
Number of mining-cuts	599	Number of periods (years)	17
Total tonnage of rock (Mt)	427.33	A(rows × columns)	63,474×30,549
Total ore tonnage (Mt)	116.29	No. Of nonzero elements in A	510,094
Total tonnage of recovered Fe (Mt)	76.33	Number of decision variables	30,549
Average grade of MWT%	72.9%	Number of integer variables	10,183
Mining capacity (Mt/year)	30		

Table 2. Inputs and numerical results for the data set containing 2598 blocks.

Settings	Processing capacity periods - pu^t / pl^t (Mt)	Grade blending $gu^{e,t}$ / $gl^{e,t}$ (%)	NPV ($M)	Root node gap %	CPU time (S)
1– default	1to 17 - 10/0	$0 \le S \le 1.8$ $0 \le P \le 0.14$ $55 \le MWT \le 85$	2,315.29	2.3	15,574
2- with probing	1 to 4 - 0/0 5 to 6 - 6/5 7 to 8 - 8/7 9 to 17- 10/0	$0 \le S \le 1.8$ $0 \le P \le 0.14$ $60 \le MWT \le 85$	2,141.60	2.02	7,957

As it was expected the net present value in the second case dropped because of the tighter bounds imposed on the model. 114 million tonnes of ore was processed out of 116.29 million tonnes available.

Fig. 3 illustrates the plan view of bench 1567m, the orebody is outlined with the final pit and the waste blocks are represented by their rock type color profile. Fig. 4 illustrates the generated schedule on bench 1567m with the order of extraction color profiled. Fig. 5 illustrates the schedule at the block level on bench 1567m. Fig. 6 to 9 represent the schedule on cross section 98300m looking east and cross section 600840 looking north.

Fig. 10 and 11 show the yearly schedule of the ore and waste production along with the average grade of MWT, S, and P through out the mine life.

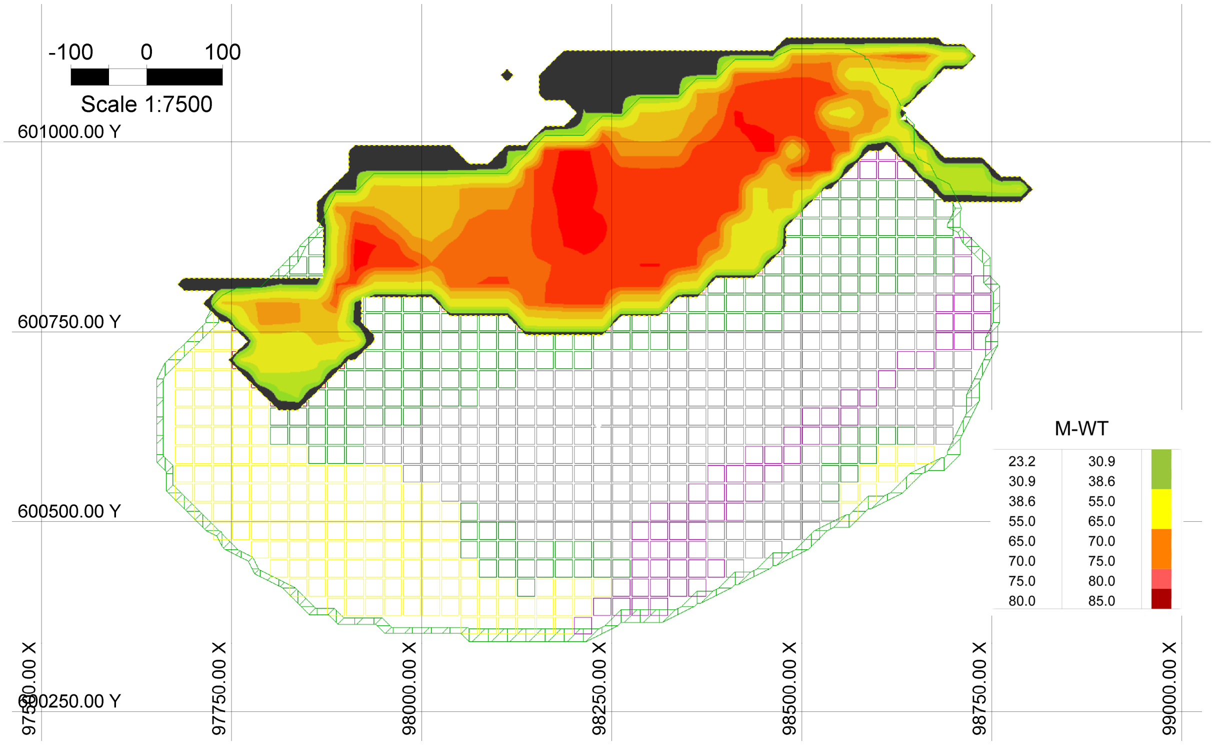


Fig. 3. Plan view of bench 1567m, the orebody is outlined, with the MWT grade; the green line shows the outline of final pit and the waste blocks are represented by their rock type color profile.

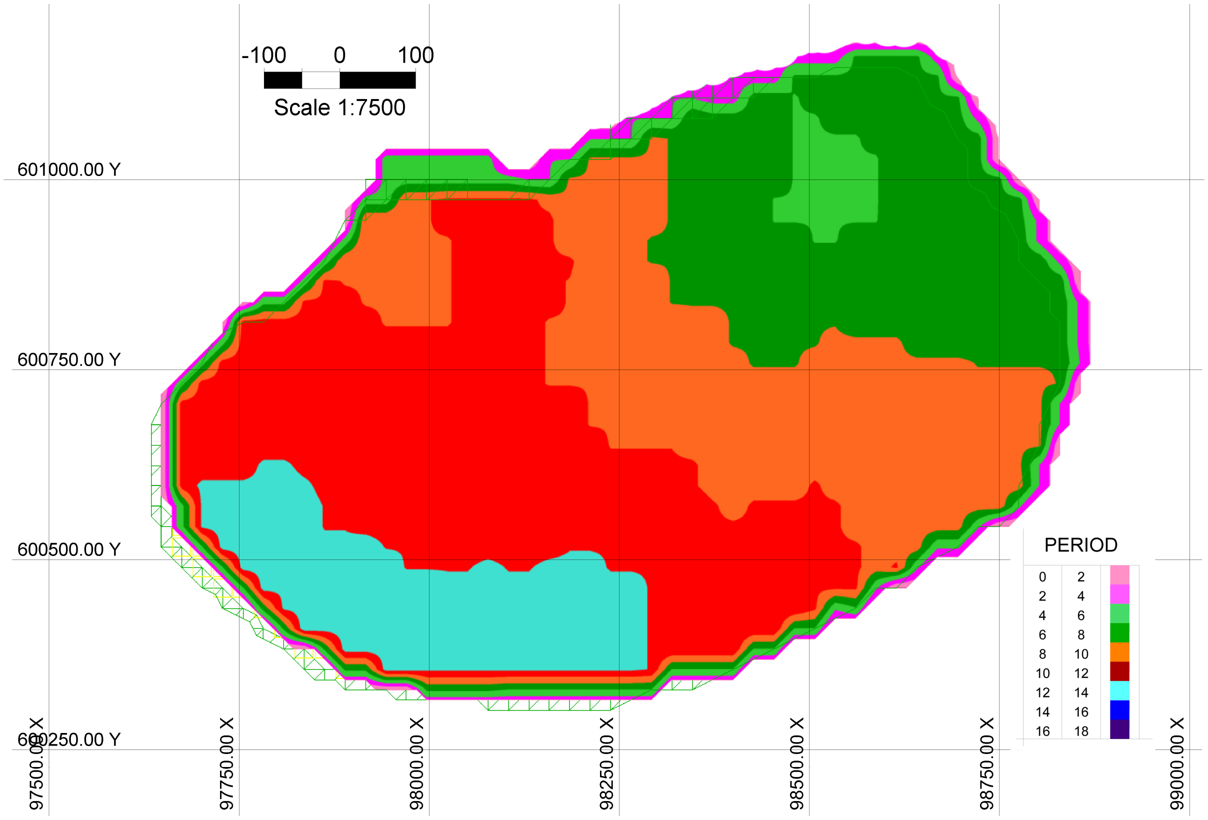


Fig. 4. Plan view of bench 1567m schedule, the extraction periods are colored sequentially.

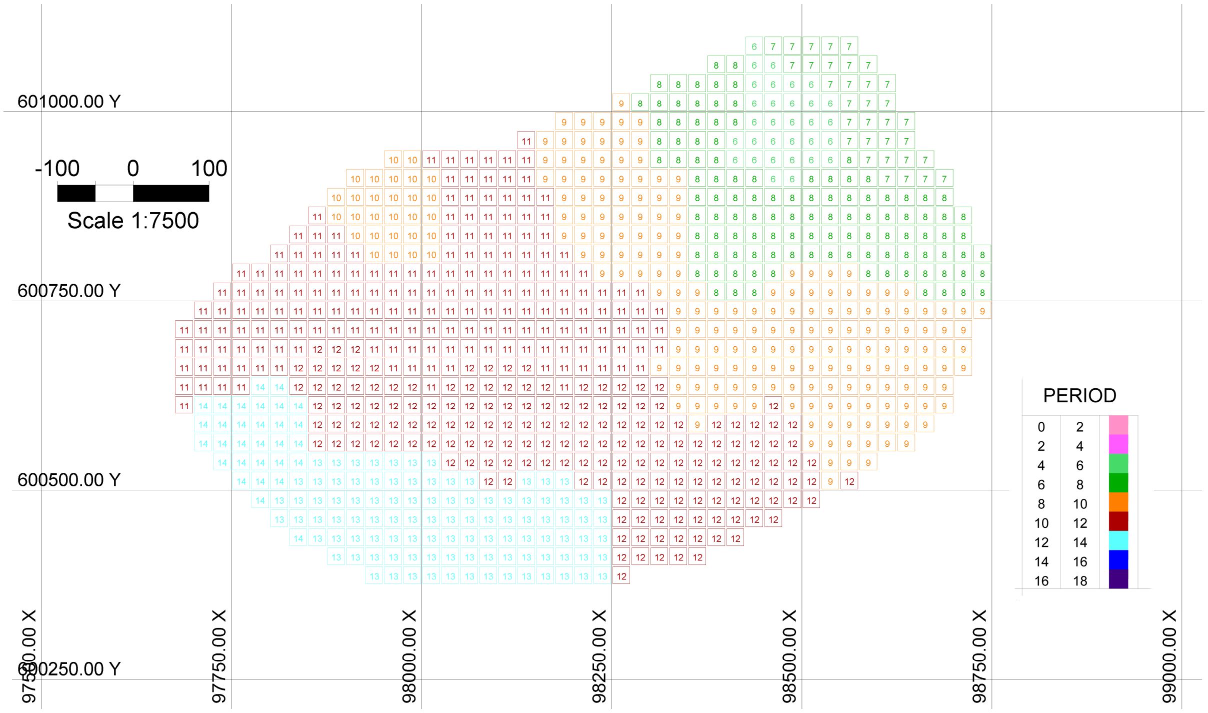


Fig. 5. Plan view of bench 1567m schedule, the extraction periods are defined at the block level, units in meters.

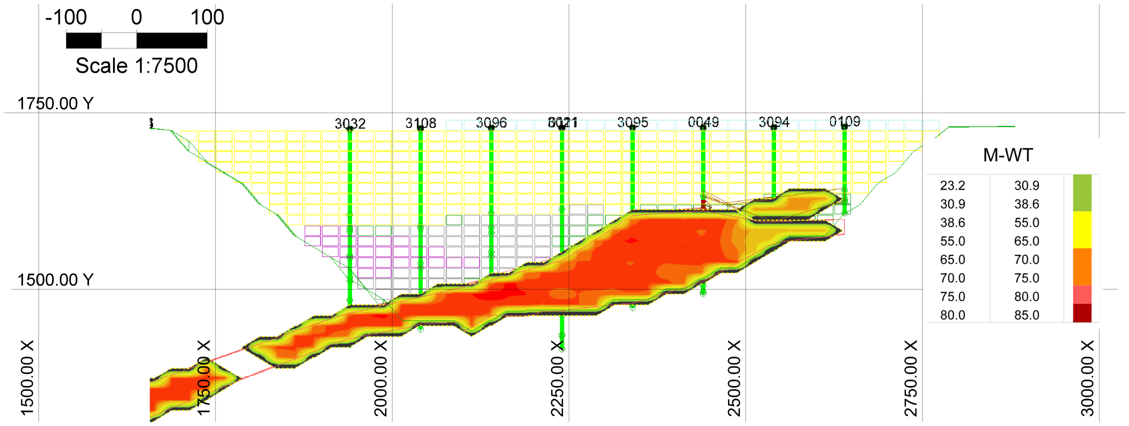


Fig. 6. Cross section 98300m looking east, the orebody is outlined, with the MWT grade, the waste blocks are represented by their rock type color, units in meters.

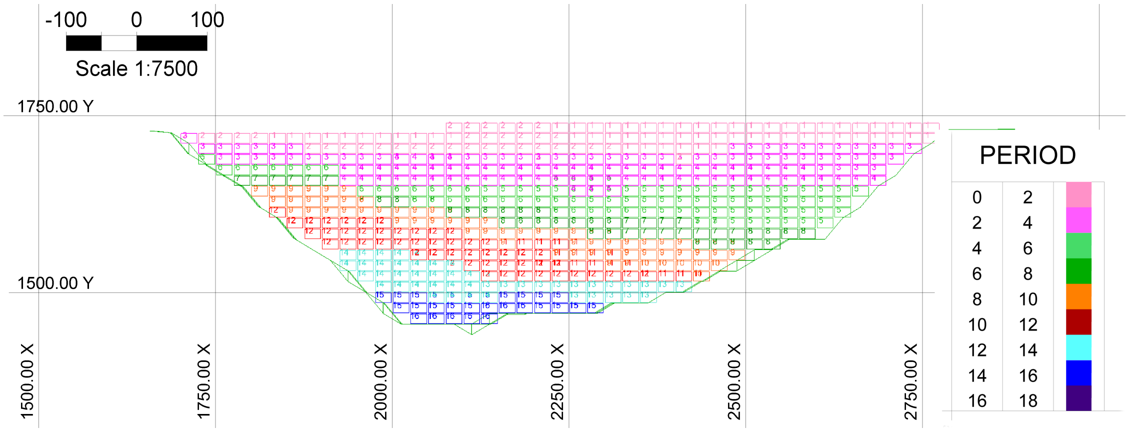


Fig. 7. Schedule of cross section 98300m looking east, units in meters.

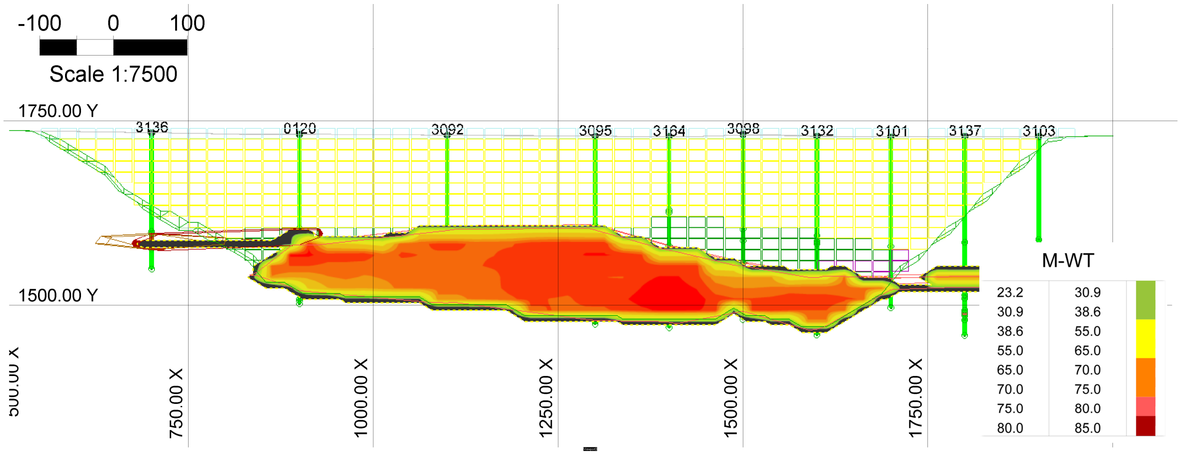


Fig. 8. Cross section 600840m looking north, the orebody is outlined, with the MWT grade, the green line shows the outline of final pit and the waste blocks are represented by their rock type, units in meters.

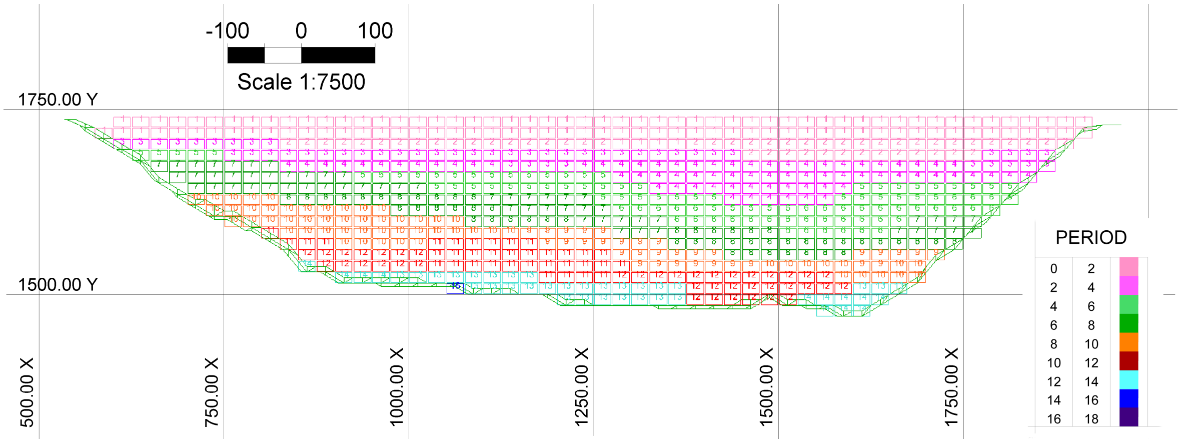


Fig. 9. Schedule of cross section 600840m looking north, extraction periods are color profiled, units in meters.

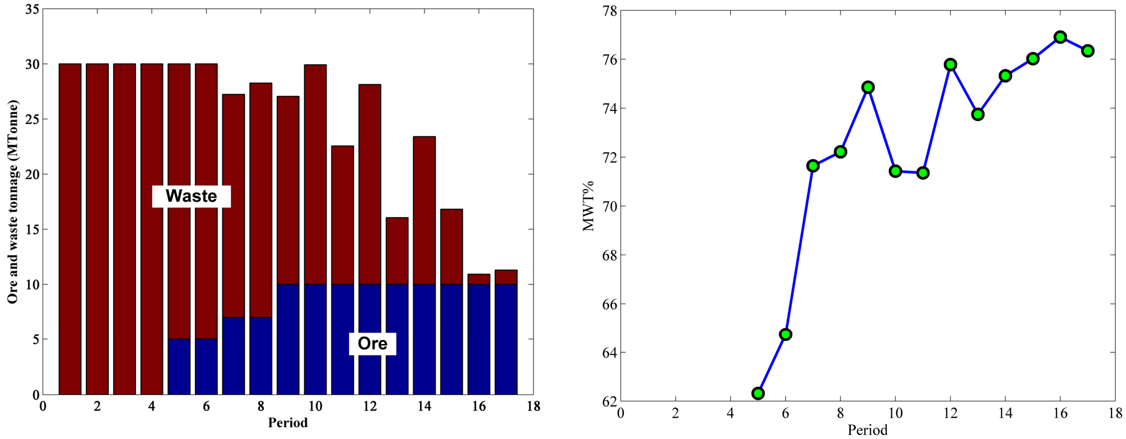


Fig. 10. Ore and waste tonnage schedule (left-a), Average grade MWT% per period (right-b), period in years.

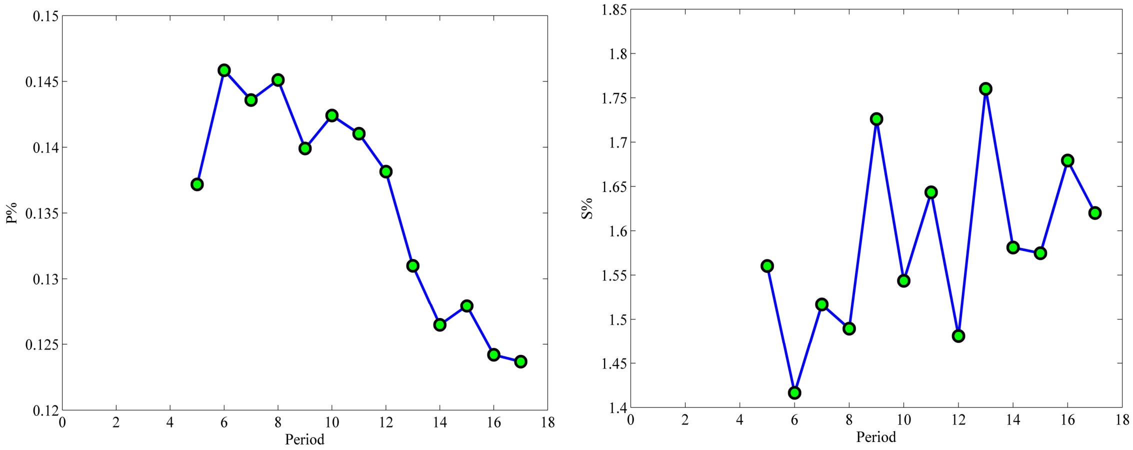


Fig. 11. Average grade P% per period (left-a), average grade S% per period (right-b), periods in years.

5. Conclusions

The applications of the MILP model developed in this study showed that it has the capability of generating production schedules within a close gap to the theoretical optimal net present values for mining operations. The balance between the number of mining-cuts and the total number of blocks in the model is very important. Research is underway to test the developed models on large-scale open pit problems with up to two hundred thousand blocks within the final pit over thirty years of mine-life. In the future we will focus on reformulating the problem with reduced number of integer variables and complexity and tackling the geological uncertainty within the mine planning domain.

6. References

[1] Bixby, R. E., (1987-2009), "ILOG CPLEX", ver. 11.0, Sunnyvale, CA , USA: ILOG, Inc.

[2] Boland, N., Dumitrescu, I., Froyland, G., and Gleixner, A. M., (2009), "LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity", *Computers and Operations Research*, Vol. 36, 4, pp. 1064-89.

[3] Boland, N., Fricke, C., and Froyland, G., (2007), "A strengthened formulation for the open pit mine production scheduling problem", Optimization Online, © the Mathematical Programming Society and by the Optimization Technology Center, Retrieved Jan. 9, 2009 from: http://www.optimization-online.org/DB_HTML/2007/03/1624.html

[4] Caccetta, L. and Hill, S. P., (2003), "An application of branch and cut to open pit mine scheduling", *Journal of Global Optimization*, Vol. 27, November, pp. 349 - 365.

[5] Dagdelen, K. and Kawahata, K., (2007), "Oppurtunities in Multi-Mine Planning through Large Scale Mixed Integer Linear Programming Optimization", *in Proceedings of 33rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 337-342.

[6] Holmström, K., (1989-2009), "TOMLAB /CPLEX - v11.2", ver. Pullman, WA, USA: Tomlab Optimization.

[7] Horst, R. and Hoang, T., (1996), "Global optimization : deterministic approaches", © Springer, Berlin ; New York, 3rd ed, Pages xviii, 727 p.

[8] ILOG Inc, (2007), "ILOG CPLEX 11.0 User's Manual September", ver. 11.0: ILOG S.A. and ILOG, Inc.

[9] MathWorks Inc., (2007), "MATLAB 7.4 (R2007a) Software", MathWorks, Inc.

[10] Ramazan, S. and Dimitrakopoulos, R., (2004), "Traditional and new MIP models for production scheduling with in-situ grade variability", *International Journal of Surface Mining, Reclamation & Environment*, Vol. 18, 2, pp. 85-98.

[11] Winston, W. L., (1995), "Introduction to mathematical programming : applications and algorithms", © Duxbury Press, Belmont, Ca., 2nd ed, Pages xv, 818, 39 p.

[12] Wolsey, L. A., (1998), "Integer programming", © J. Wiley, New York, Pages xviii, 264 p.

Transfer of geological uncertainty into mine planning

Behrang Koushavand[1] and Hooman Askari-Nasab
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

Uncertainty is always presented in presence of sparse geological data. Conditional simulation algorithms such as Sequential Gaussian Simulation (SGS) and Sequential Indicator Simulation (SIS) are geostatistical methods used to assess geological uncertainty. The generated realizations are equally probable and represent plausible geological outcomes. Long-term mine planning and the management of future cash flows are vital for surface mining operations. Traditionally the long-term mine plans are generated based on an estimated input geological block model. The most common estimation method used in industry is Kriging; however, Kriging results do not capture uncertainty and may be systematically biased. Mine plans that are generated based on one input block model fail to quantify the geological uncertainty and its impact on the future cash flows and production targets. A method is presented to transfer geological uncertainty into mine planning. First, Sequential Gaussian Simulation is used to generate fifty realizations of an oil sands deposit. An optimum final pit limits design is carried out for each SGS realization while fixing all other technical and economic input parameters. Afterwards, the long-term schedule of each final pit shell is generated. Uncertainty in the final pit outline, net present value, production targets, and the head grade are assessed and presented. The results show that there is significant uncertainty in the long-term production schedules. In addition, the long-term schedule based on one particular simulated ore body model is not optimal for other simulated geological models. The mine planning procedure is not a linear process and the mine plan generated based on the Kriging estimate is not the expected result from all of the simulated realizations. The probability of each block being extracted in each planning period and the probability that the block would be treated as ore or waste in the respective period are calculated and can be used to assist in long range mine planning.

1. Introduction

Mining planning is a process to find a feasible block extraction schedule which maximizes net present value (NPV) and is one of the critical processes in the mining engineering. Also there are some technical, financial and environmental constraints that should be considered. The uncertainty of ore grade may cause some shortfalls at the designed production and discrepancies between planning expectations and actual production. Vallee (2000) reported that 60% of the mines surveyed had 70% less production than designed capacity in the early years. Others (e.g., Rossi and Parker, 1994) reported shortfalls against

[1] PhD student Centre for Computational Geostatistics

predictions of mine production in later stages of production. Traditional production scheduling methods which do not consider the risk of not meeting production targets caused by grade variability, cannot produce optimal results. Rendu 2002 and Osanloo et al 2007 have explained the uncertainty assessment and algorithms in long-term mine planning. The negative effects of grade uncertainty in optimizing open pit mine design are articulated in Smith and Dimitrakopoulos (1999), Dimitrakopoulos et al. (2002) and Godoy and Dimitrakopoulos (2004) . Most of these methods are not efficient because they assess risk in a schedule, and do not produce optimal scheduling solutions in the presence of uncertainty. In addition, these efforts do not consider multi-element deposits with complex ore quality constraints, such as nickel laterites, iron ore or magnesium deposits. Furthermore, dealing with orebody uncertainty needs to consider issues of equipment access and mobility in the related “stochastic” optimization formulations. Dimitrakopoulos and Ramazan (2003) present a risk-based production-scheduling formulation for complex, multi-element deposits, which was based on expected block grades and probabilities of grades being above required cutoffs. They use jointly simulated models to get these values, but these values come from local distribution without considering jointly uncertainty and at multi-Gaussian framework, kriging can fully characterized local distribution. Also probability of extraction is not equal to probability of grades being above required cutoffs because in real life, a block may not be extracted even if it has grade above cutoff. This may happens because of capacity of mill. Therefore using these values as uncertainty assessment parameters would not be appropriate.

2. Methodology

The concepts and results will be illustrated through a case study corresponding to an oil sands deposit in Fort McMurray, Alberta. In order to quantify the geological uncertainty transferred into the mine plans the following steps have been followed:

2.1. Geostatistical modeling

There are geostatistical simulation methods that are widely used to assess uncertainty in GEO-data. All of these realizations are equal probable and can be consider as a plausible representative of geological complexity. Choosing one or some of these realizations will not be objective to fair uncertainty assessment. It is important to use sufficient number of realizations to get a robust mine plan. We have followed the steps presented by Leuangthong et al. (2004) in geostatistical modeling of oil sand deposits using GSLIB (Deutsch and Journel, 1998) software catalog to create conditional simulated realizations. The name of the programs used in the following paragraphs all refer to GSLIB programs (Deutsch and Journel, 1998). Stages presented by Leuangthong et al. (2004) are:

1. Analyze of correlation structure. This investigates whether a transformation of the vertical coordinate system is required, in order to determine the true continuity structure of the deposit. Determination of the correct grid is dependent on the correlation grid that yields the maximum horizontal continuity.
2. Decluster drillhole data distribution. The relevant statistics must be deemed representative of the deposit prior to modeling. Any or a combination of cell,

nearest neighbour and/or declustering by kriging weights may be employed to determine the summary statistics that are representative of the field.

We used the DECLUS program to get declustering weights whereby values in areas/cells with more data receive less weight than those in sparsely sampled areas. The DECLUS program provides an algorithm for determining 3D declustering weights in cases where the clusters are known to be clustered preferentially in either high or low valued-areas.

3. Model spatial continuity of the bitumen grade using variograms, which is the common spatial measure of continuity that shows the variability of grades with distance.

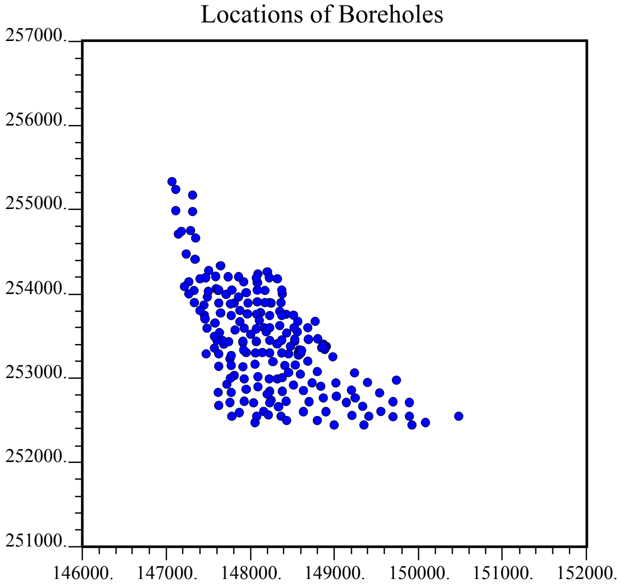


Figure 1. Location map of boreholes, units in meters.

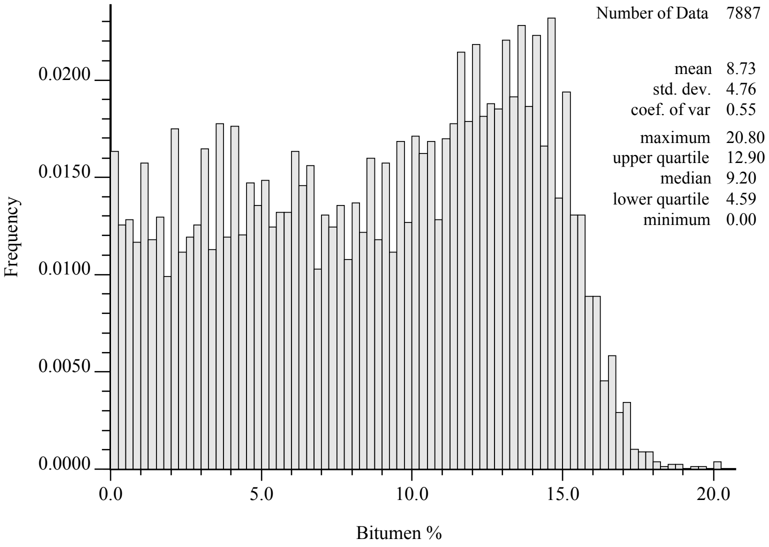


Figure 2. Histogram of bitumen grade (%mass).

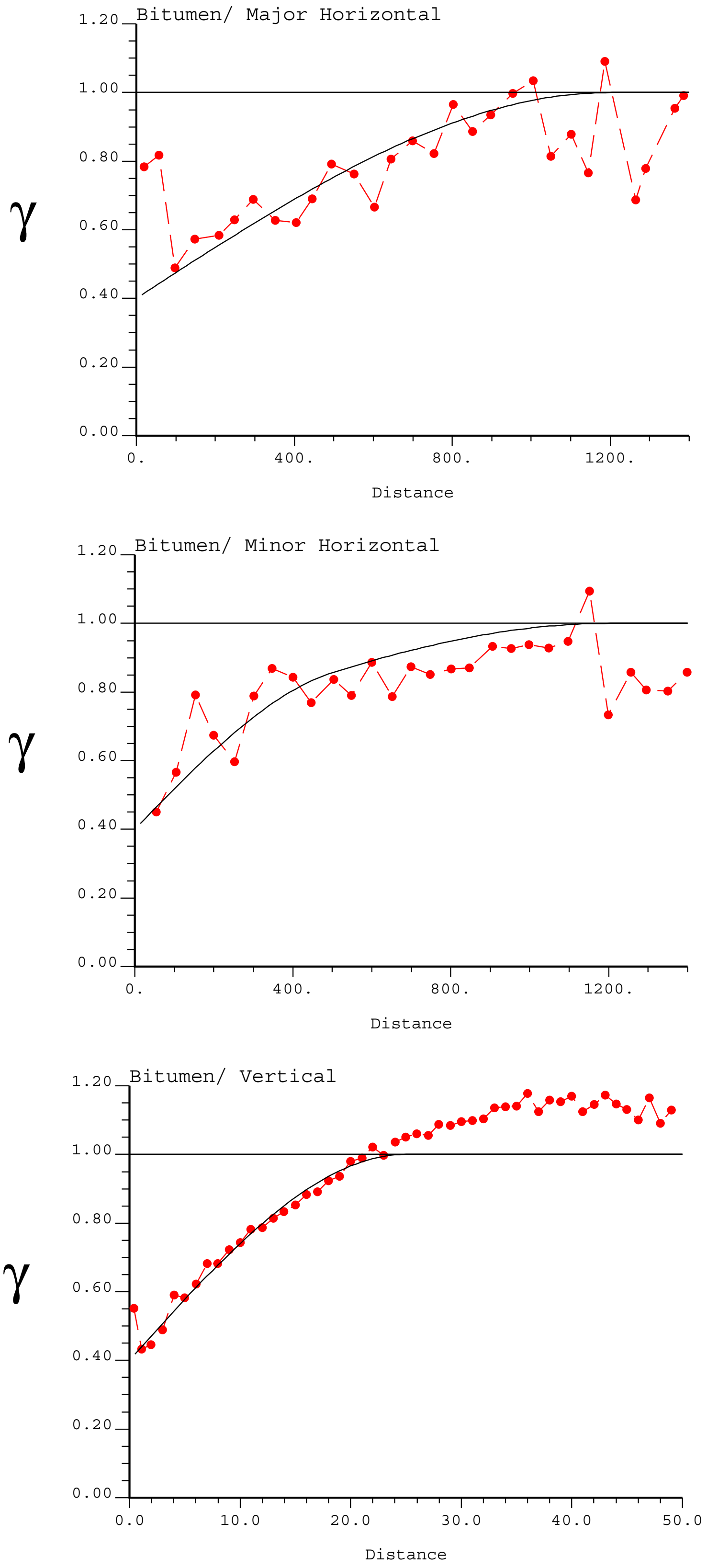


Figure 3. Experimental directional variograms (red dots) and the fitted variogram models (dash lines), distance units in meters.

We transferred data into normal scores using the NSCORE program; afterwards we calculated the directional experimental variograms using the GAMV program. VMODEL program was used to fit an anisotropic variogram with two nested spherical structures. The Azimuths of major and minor directions were 50 and 140 degrees. Figure 1 shows the location map of boreholes and Figure 2 shows histogram of bitumen grades in mass percent. Figure 3 shows the experimental and fitted variogram model in major, minor and vertical directions. The block defined bellow is chosen to model the domain.

4. Perform estimation and cross validation using kriging as checks against simulation results. Cross validation using kriging provides a quality control check on the estimation (and also simulation) parameters.

We used a large block with the origin coordinates of X = 146000m, Y = 251000m, and Z=190m to model the domain. The small blocks represent a volume of rock equal to 50 m×50 m×10 m. The model contains 331,200 blocks that makes a model framework with dimensions of 120X ×120Y×23Z.

We used Ordinary Kriging to estimate the bitumen grade at each block location. Kriging is widely accepted in the mining industry and yields a map of the large Figure 3 expected value of conditional distribution at each location, E-type result is the mean value of simulation values at each location. Both of Ordinary Kriging and E-type estimation were done at the original unit. The shape of variogram and histograms were not produced with these two methods.

Figure 4 illustrates the plan view of the Kriged model at the elevation 290m. Also, Figure 5 shows the E-type estimation for fifty realizations. There are more blocks below the 6% bitumen cutoff represented by blue color in the kriged model (igure) compared against the E-type model (Figure 5).

5. Generate multiple realizations of bitumen grade using Sequential Gaussian Simulation (SGS) (Isaaks, 1990), this method is the means of constructing uncertainty models of bitumen grades. SGS is by far one of the most commonly applied geostatistical simulation algorithm applied in the natural resources sector. It has been extensively validated and provides a measure of local and global uncertainty, which is not afforded from Kriging.

We used the SGSIM program to simulate fifty conditional realizations using the variogram model presented in Figure 3. Figure 6 illustrate the plan view of realization thirteen (L13) at 290m elevation. Among all the realizations L13 had the minimum amount of ore tonnage and the minimum amount of recovered bitumen; we refer to L13 as the worst case. Figure 7 shows the plan view of realization twenty nine (L29) at 290m elevation. Among all the realizations L29 had the maximum amount of ore and the maximum amount of recovered bitumen; we refer to L29 as the best case.

6. Check simulation results against the input data and compare results against the kriged models. We checked the quality of geo-model by histogram and variogram reproduction.

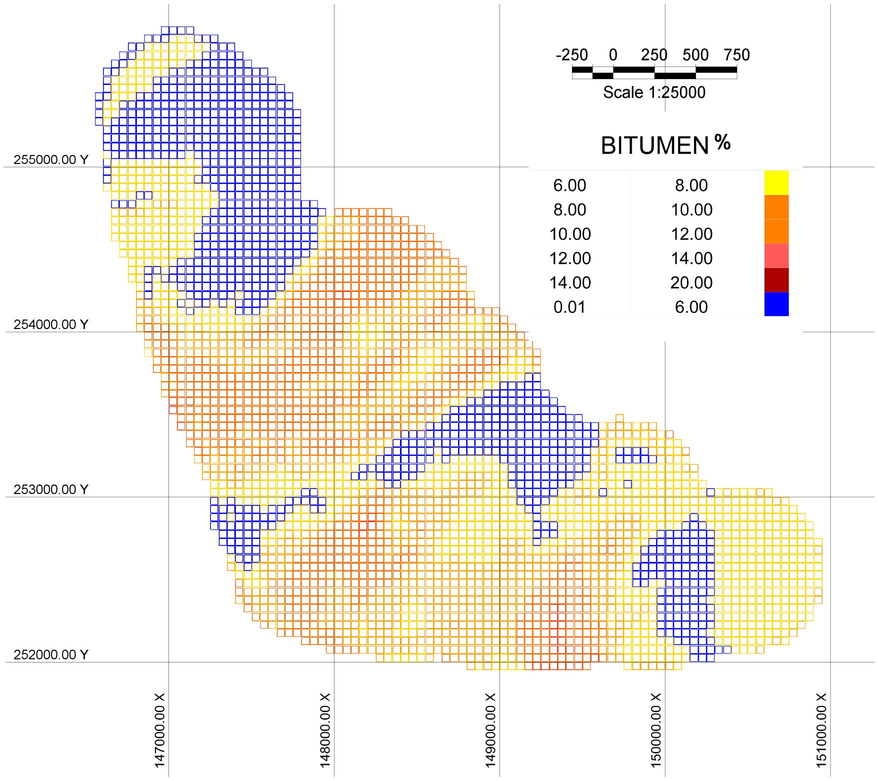


Figure 4. Plan view of the kriged model at 290m elevation.

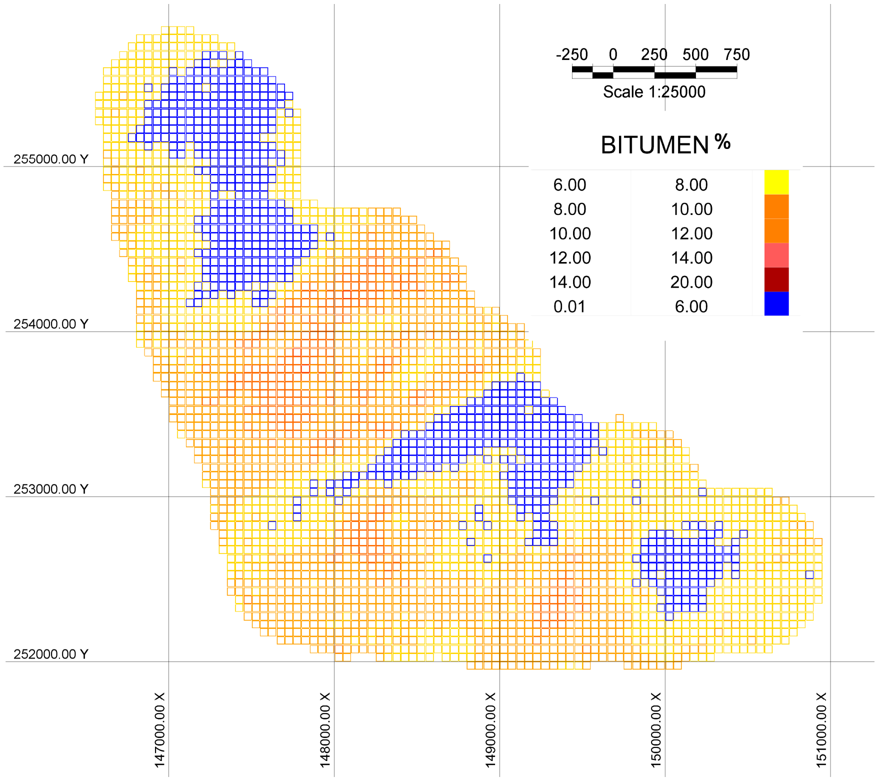


Figure 5. Plan view of the E-type model at 290m elevation.

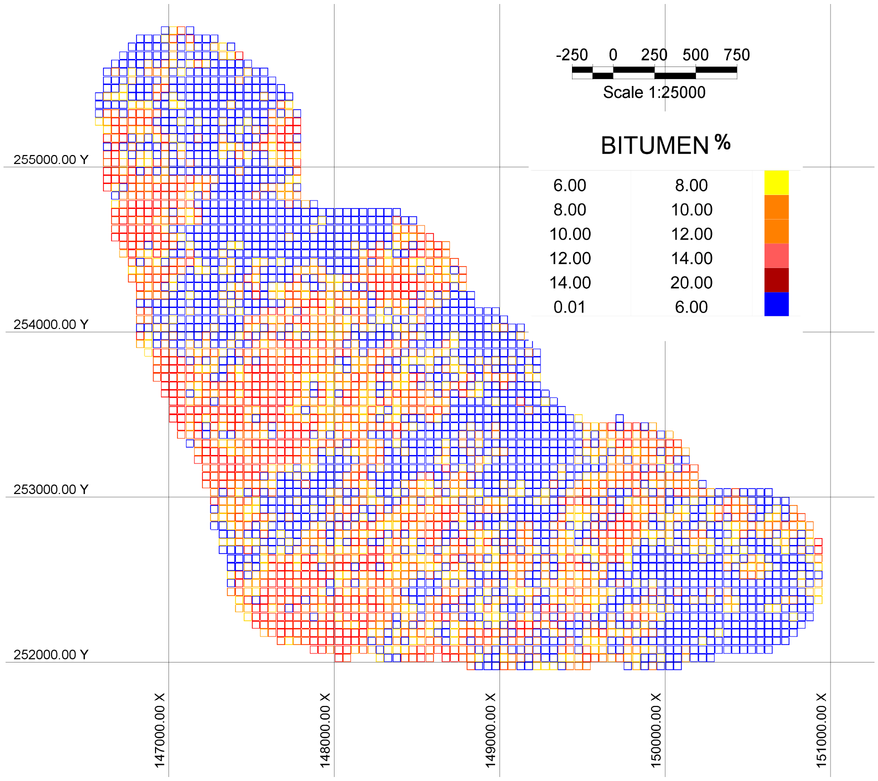


Figure 6. Plan view of the worst case realization (R13) at 290m elevation.

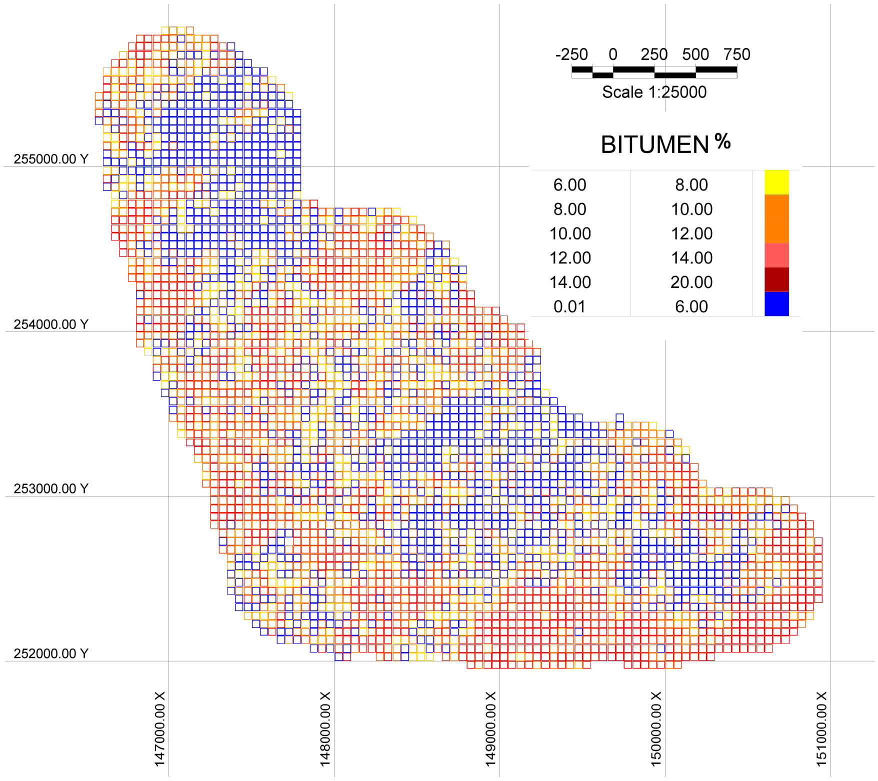


Figure 7. Plan view of the best case realization (R29) at 290m elevation.

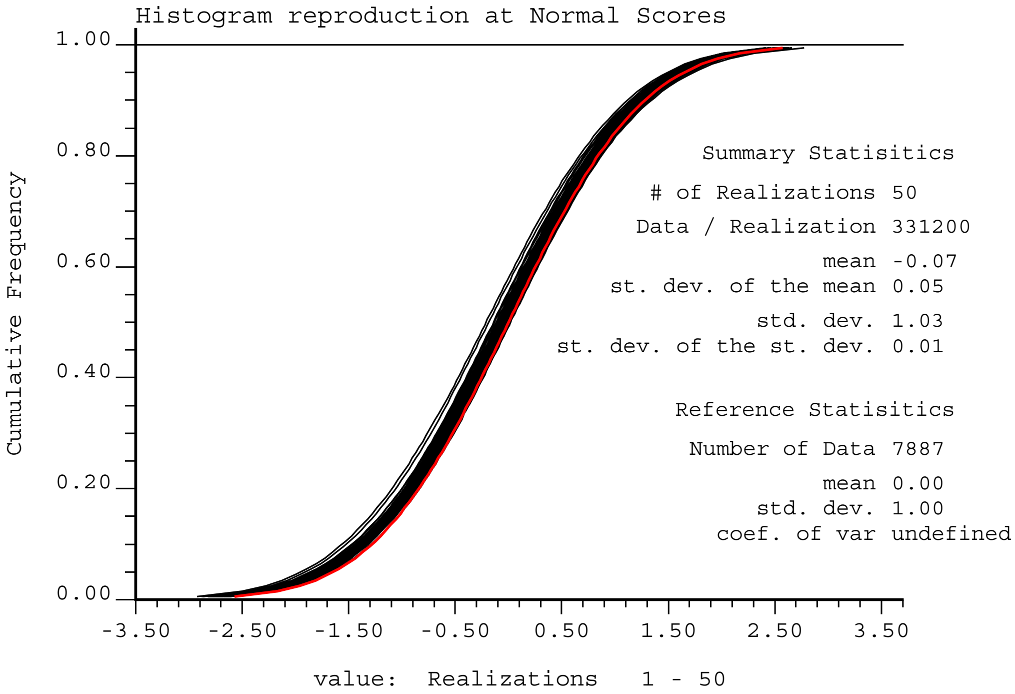


Figure 8a. Histogram reproduction of simulation realizations (black lines) and reference distribution (red line) at normal scores.

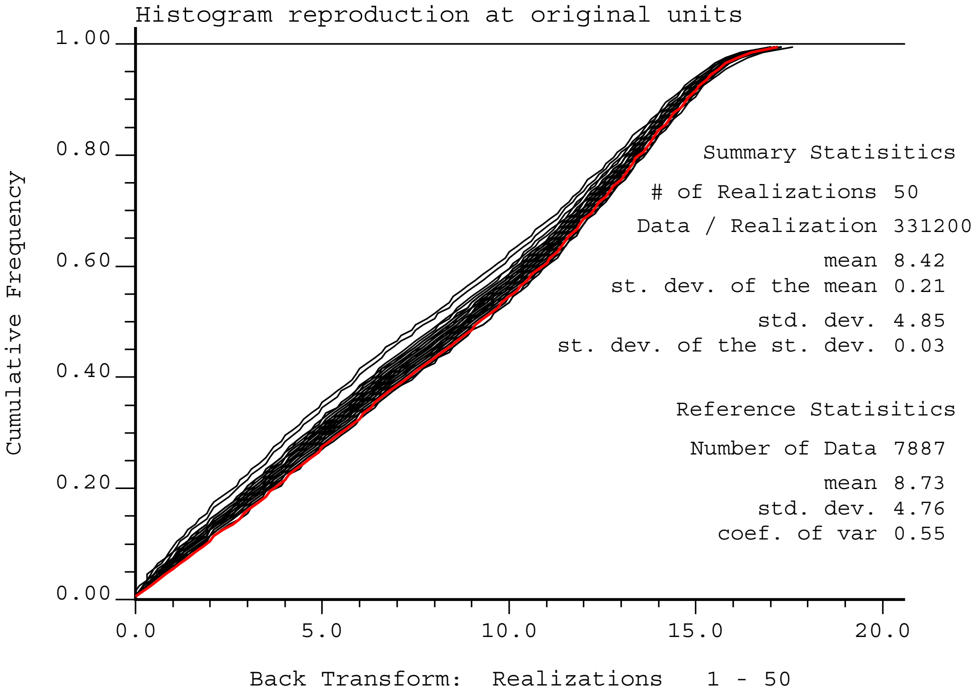


Figure 8b. Histogram reproduction of simulation realizations (black lines) and reference distribution (red line) at original units.

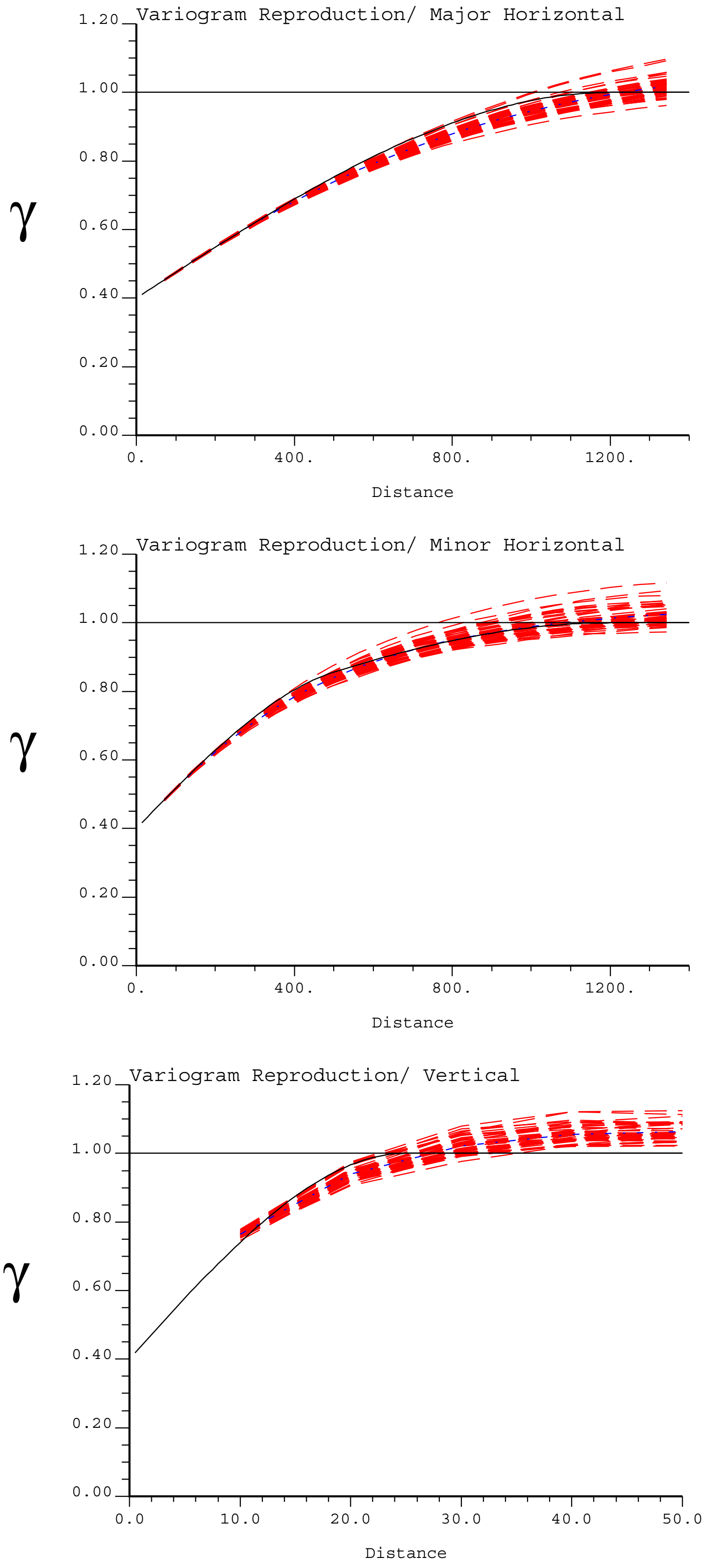


Figure 9. Variogram reproduction of simulation realizations (red dash lines) and reference variogram model (black line).

Figure 8 shows the values at normal scores and the original units. Black lines are cumulative distribution of each realization and the red line is the reference distribution. Figure 9 shows the variogram reproduction at horizontal major and minor and vertical directions. SGSIM program has reproduced both histogram and variogram very well.

2.2. Optimal final pit outline design

The final pit limit design was carried out based on the industry standard Lerchs and Grossmann algorithm (Lerchs and Grossmann, 1965) using the Whittle strategic mine planning software (Gemcom Software International, 1998-2008). The kriged, E-type, and fifty SGS realization were imported into the Whittle software.

The ultimate pit limit design was carried out based on the Syncrude's costs in CAN$/bbl of sweet blend for the third quarter of 2008 (Jaremko, 2009). Price of oil was considered US $45 with an exchange rate of 1.25:1 equal to CAN $56.25/bbl SSB for the same time period. We assumed that every two tonnes of oil sands with an average grade of 10% mass will produce one bbl of sweet blend, which is approximately 200 kg. We also assumed a density of 2.1 tonne/m^3 for oil sands, and a density of 1.2 to 1.5 tonne/m^3 for waste material, including clay and sand. Table 1 summarizes the costs used for pit limit design. The mining costs of $12.18 is per tonne of oil sands ore, we assumed a stripping ratio of 1.8:1 for our model this would lead to a cost of $4.6/tonne of material removed (ore and waste).

Table 1. Summary of costs used in pit limit design.

Description	Value	Description	Value
Mining Costs (CAN $/ bbl SSB)	24.35	Mining Costs (CAN $/tonne)	12.18
Upgrading Costs (CAN $/ bbl SSB)	10.05	Upgrading Costs (CAN $/tonne)	5.025
Others (CAN $/bbl SSB)	1.5	Others (CAN $/tonne) 0.75	0.75
Total Costs (CAN $/ bbl SSB)	35.9	Total Costs (CAN $/ tonne)	17.28

Table 2. Final pit and mine planning parameters.

Description	Value	Description	Value
Cutoff grade (%mass bitumen)	6	Processing limit (M tonne/year)	30
Mining recovery fraction	0.88	Mining limit (M tonne/year)	70
Mining dilution factor	1	Stockpile limit (M tonne)	70
Processing recovery factor	0.95	Overall slope (degrees)	20
Minimum mining width (m)	150	Pre-stripping (years)	4

Table 2 shows the pit design factors used in the study. Thirty nine pitshells were generated using 49 fixed revenue factors ranging between 0.3 to 2.5. The number of pitshells was reduced to 7 after applying the minimum mining width of 150 meters for the final pit and the intermediate pits. Table 3 summarizes the amount of material in the final pit limit for the kriged block model with a 6% cutoff grade. The minimum slope error, the average lope error and the maximum slope error respectively are: 0.0 degrees, 0.2 degrees, 0.4 degrees.

The final pit limits was designed for E-type model and all the fifty simulation realizations with the exact same input variables.

Table 3. Materials in the final pit.

Description	Value
Total tonnage of material (M tonne)	1077.44
Tonnage of ore (M tonne)	423.62
Tonnage of material below cutoff (M tonne)	137.06
Tonnage of waste (M tonne)	516.76
Bitumen recovered (M tonne)	40.30
Stripping ratio (waste:ore)	1.54

2.3. Production scheduling

The kriged model was the basis for production scheduling, we aimed at keeping a uniform processing feed through out the mine life, to achieve this goal we defined four pushbacks based on pitshells number 1, 2, 4, and 7. A fixed lead of three benches was defined, where the lead is the number of benches by which the mining of a specified pushback is ahead of the next one. Four years of pre-stripping was considered to make sure enough space and ore would be available throughout the mine life. Since we wanted to study the effect of geological uncertainty on the production schedule, a buffer stockpile with the capacity of 70 million tonne was included to minimize the effect of geological uncertainty on achieving the production target of 30 million tonnes of ore per year. Although, stockpiles are not used in real oil sands mining operations.

Figure 10a illustrates the kriged block model schedule, in the first four years ore is extracted but the processing plant is not available until year four, the extracted ore in the first four years is sent to the stockpile with the maximum capacity of 70 million tonnes. The total amount of ore within the final pit limit is 424 and 430 million tonnes with an average grade of 10% and 9.3% for the kriged model and the E-type model respectively. Figure 10b shows the schedule for the E-type model, the exact same settings and strategy of the kriged model is followed in this case as well. In both case of the kriged and the E-type models the mine-life is sixteen years but an extra two and half years is used to process the material from the stockpile.

Figure 11a shows the production schedule for realization L13, referred to as the worst case scenario with 341 million tonnes of ore in the final pit and with an average grade of 11%. L13 had the minimum average grade and stripping ratio (1.67:1). The production target in years 5 to 9 are not satisfied with L13 and the mine-life is shorter compared to the kriged model. Figure 11b illustrates the production schedule for realization L29, referred to as the best case scenario with 400 million tonnes of ore reserve with the highest average grade of 11.87% among all the simulation realizations. L29 stripping ratio of 2:1 was the highest among all realizations. Figure 12a and 12b illustrate the yearly production schedule for the pits with minimum and maximum amount of material within the pit limits. Table 4 summarizes the material movement for the kriged, E-type, and simulation results.

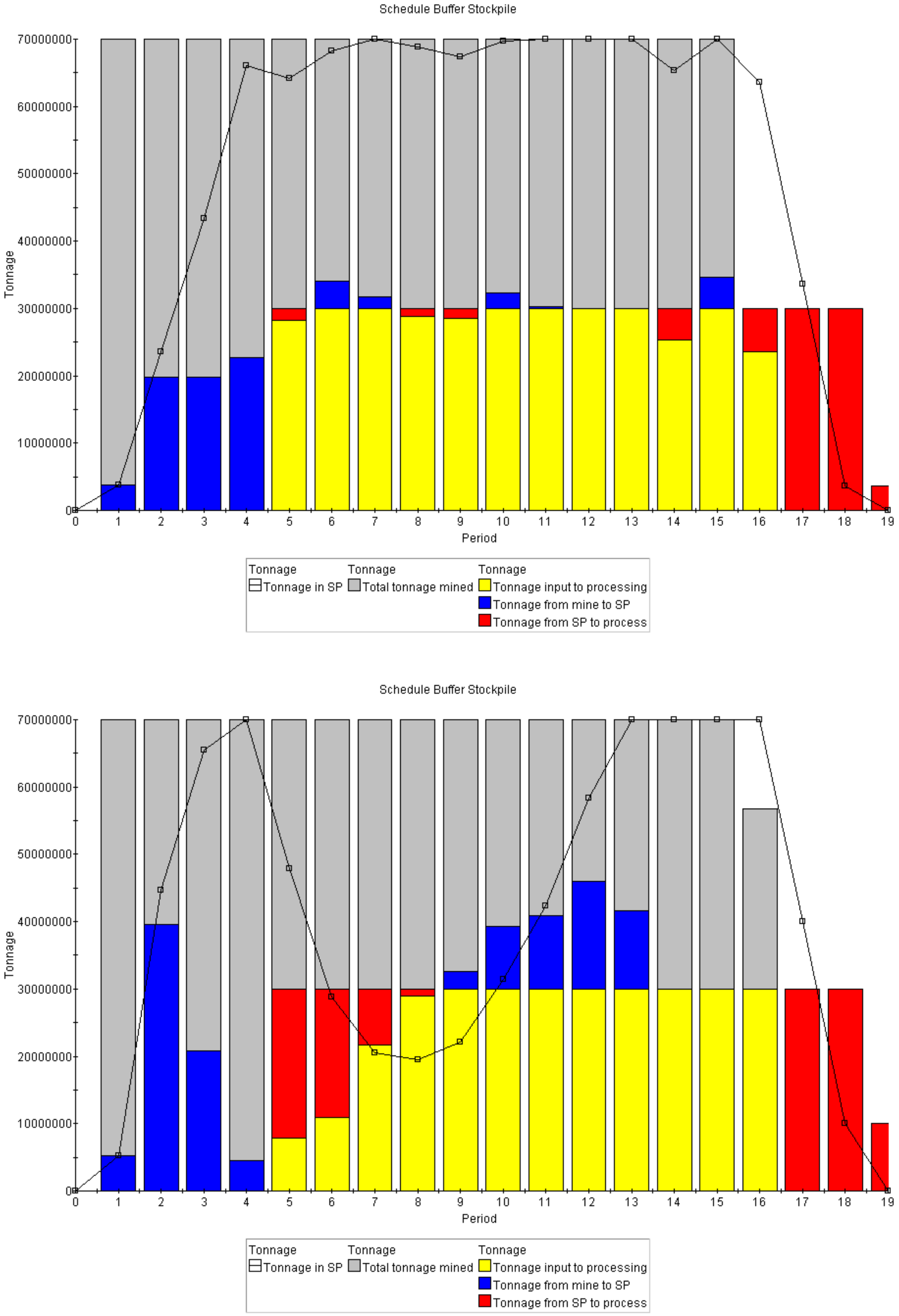


Figure 10. Schedules of Kriged values (a-top) and E-type values (b-bottom).

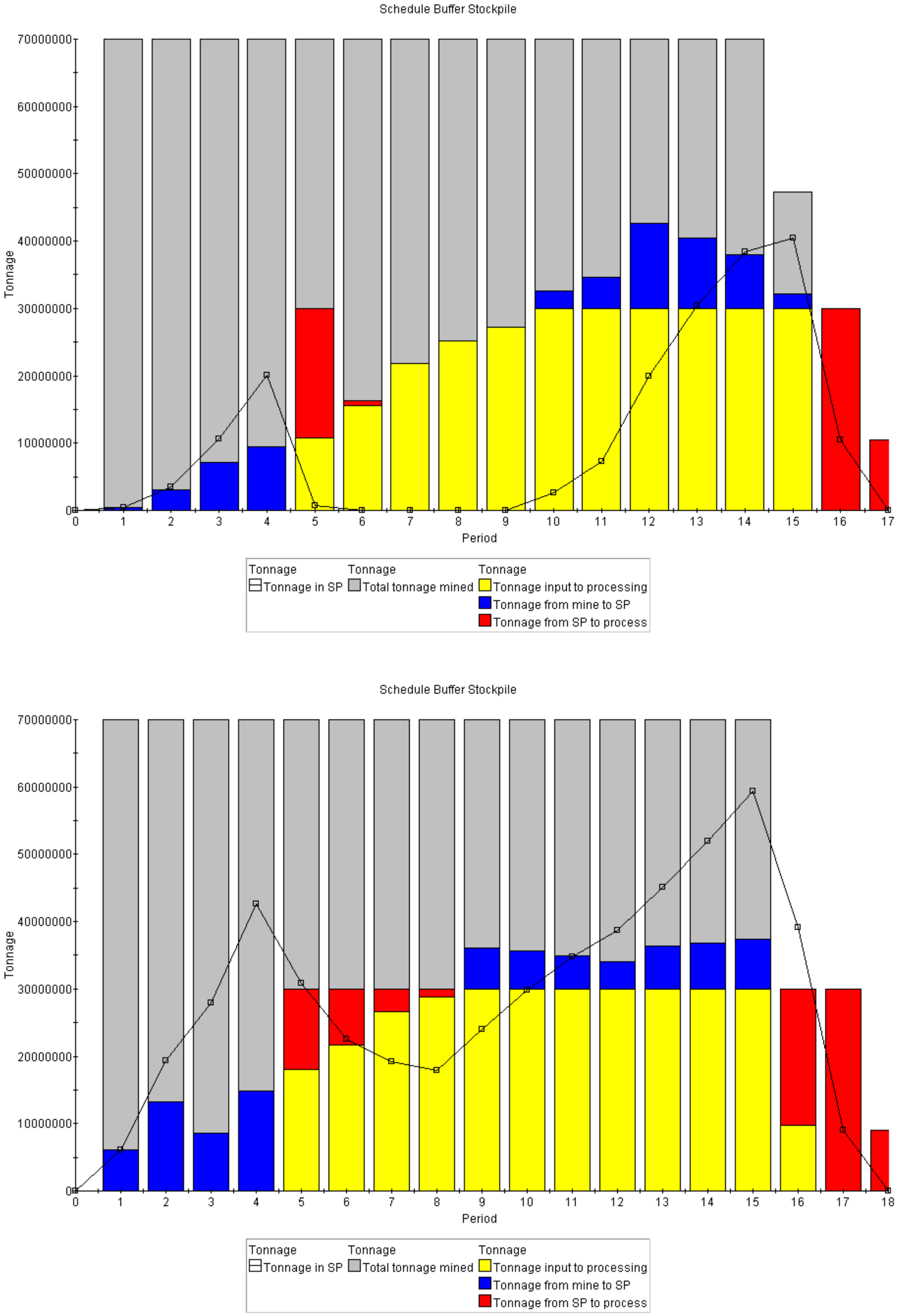
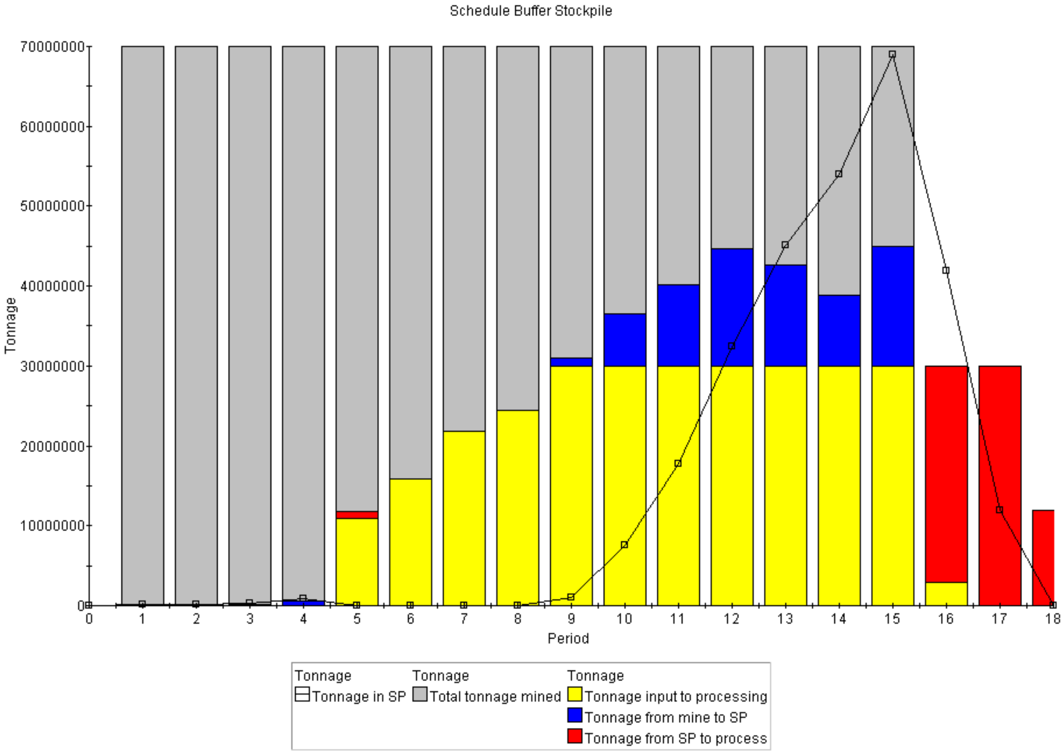


Figure 11**.** Schedules of Minimum Ore realization 13 (a-top) and maximum ore realization 29 (b-bottom).



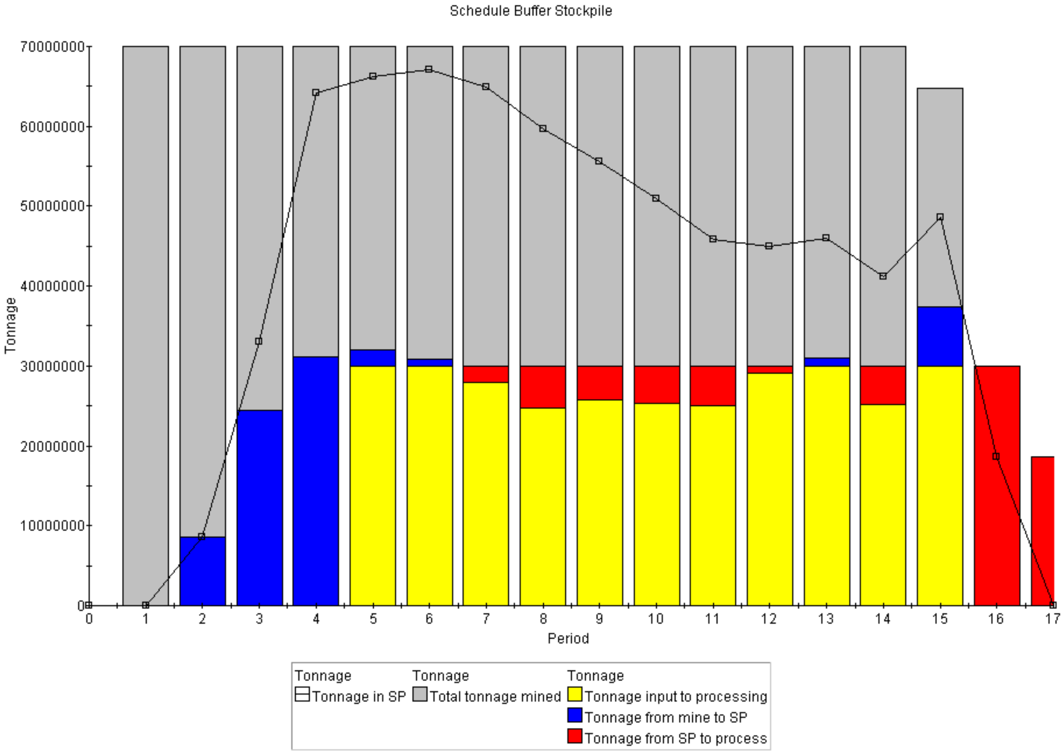


Figure 12. Schedules of minimum (a-top) maximum (b-bottom) NPV, realizations 8 and 19 respectively.

Table 4. Summary of material movement and performance measures for the kriged, E-type and simulation realizations.

		Min	Max	Mean	P50	Krig	E-type
Material Movement	Ore (M tonne)	341 (L=13)	400 (L=29)	372	371	424	430
	Waste(reject) (M tonne)	136 (L=29)	181 (L=30)	160	161	137	148
	Waste(other) (M tonne)	493 (L=31)	532 (L=43)	514	515	517	529
	Total Tonnage (M tonne)	1,002 (L=31)	1,077 (L=43)	1,046	1,047	1,078	1,107
	Stripping ratio	1.67 (L=29)	2.01 (L=13)	1.81	1.815	1.54	1.57
Product Bitumen	Input (M tonne)	37.83 (L=13)	46.66 (L=29)	42.59	42.39	42.43	40.05
	Recovered (M tonne)	35.94 (L=13)	44.33 (L=29)	40.46	40.28	40.31	38.04
	Input grade (%mass)	11.09 (L=13)	11.87 (L=19)	11.44	11.42	10.02	9.31
Measures	NPV (million $)	1,109 (L=8)	2,101 (L=19)	1,792	1,807	1,525	1,202
	Life (year)	15.85 (L=4)	17.79 (L=50)	16.63	16.65	18.12	18.33

3. Discussion

Conditional simulations enable us to provide a set of production scenarios which capture and asses the uncertainty in the final pit outline, net present value, production targets, and the head grade. The probability of each block being extracted in each planning period and the probability that the block would be treated as ore or waste in the respective period could be calculated. In this section, we present and discuss the histograms that quantify the overall uncertainty on the mine production schedule output parameters by means of conditional simulations of the grades. More precisely, the histograms and box plots for the following production schedule output variables are presented:

1. Tonnage of material within the final pit limits.
2. Overall stripping ratio.
3. Total tonnage of ore.
4. Average grade of ore.
5. Tonnage of bitumen produced.
6. Number of barrels of bitumen produced.
7. Net present value.
8. Mine life.

If we follow the production schedule generated based on the kriged model, ideally every block model generated based on the simulation realizations constitutes a plausible scenario with different outcomes. We now have an image of the uncertainty on the production schedule and the deviations from the preset targets.

Figure 13a illustrates total tonnage of material within the final pit limits, the krig model (the black line) has 1078 million tonnes which is more than the maximum amount recorded for all the simulation realizations. The inter quartile limits are 1039 and 1077 million tonnes. Although the kriged pit has more material than all the simulation models, its stripping ratio of 1.54 is less than 1.67 the minimum stripping ratio of simulation realizations (Figure 13b).

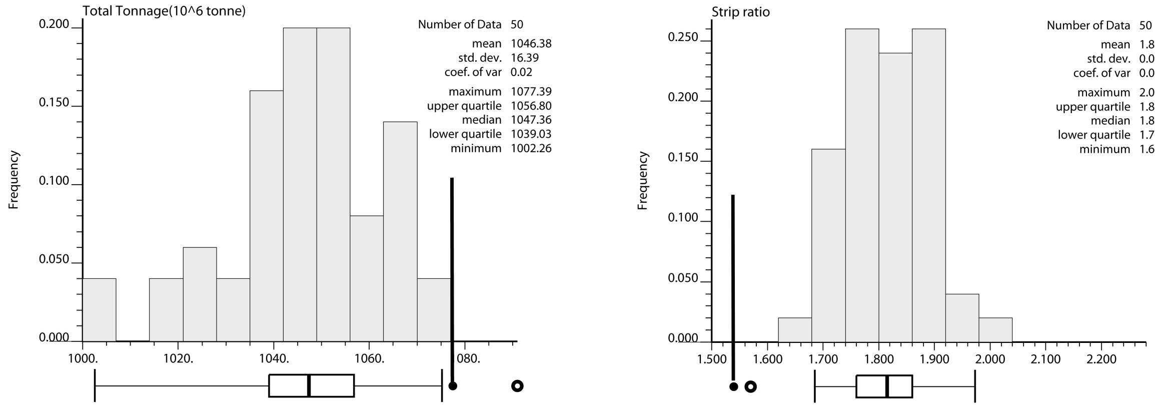


Figure 13. Histograms and box plots of total tonnage of material within the final pits (a-left), and overall stripping ratio (b-right). E-type result by circle and kriged result by black line.

The total tonnage of ore within the kriged model is 424 million tonnes with an average grade of 10% bitumen at a cutoff grade of 6% (Figure 14), where the average total tonnage of ore within the simulation realizations is 372 million tonnes with an average grade of 11.44% (Figure 14 and Table 4). The maximum amount of 400 million tonnes of ore is for realization twenty nine. The lower quartile and upper quartile for the simulation realizations are 361 and 382 million tonnes respectively. The more ore within the kriged and E-type models and assuming the processing plant as the bottleneck of the operation, would result in a longer mine life for the kriged and E-type models. The inter-quartile of the average grade of realizations is between 11.38 and 11.53%, which reflects a very small variance in for the estimated grades by simulation.

Figure 15 illustrates the yearly average head grade for the kriged model (black line), the E-type model (blue dashed line), and the simulation realizations (red dashed lines). As anticipated from the results of Figure 14 the average head grade for the kriged and E-type models are less than the simulation realizations (Figure 15), the head grade variance in the early years of production are higher as it is shown in the box plot in Figure 16, this variance reduces for the later years and almost a uniform head grade could be expected for years 11 to 15.

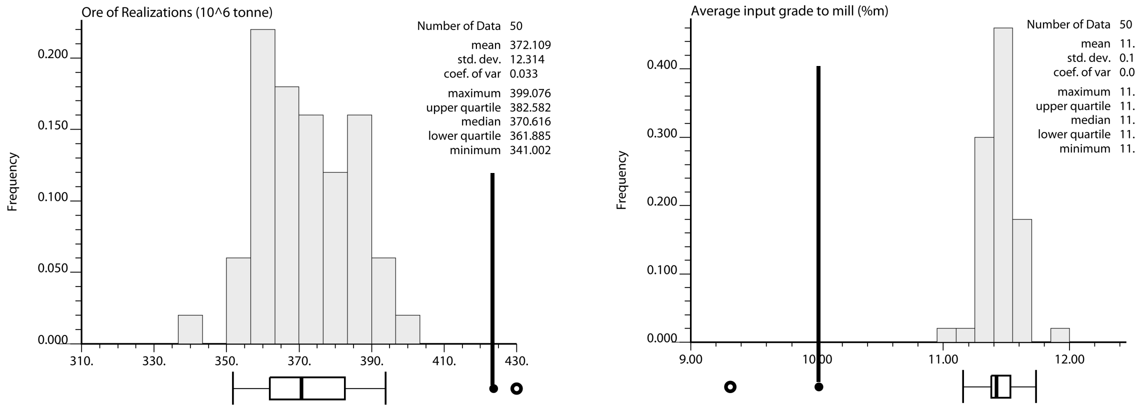


Figure 14. Histograms and box plots of total tonnage of ore (left), and the average head grade in bitumen %mass (right). E-type result by circle and kriged result by black line.

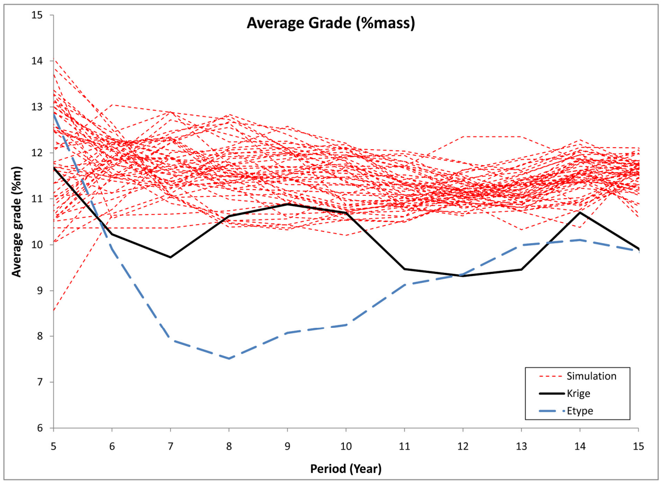


Figure 15. Average head grade simulation realizations, kriging, and E-type models.

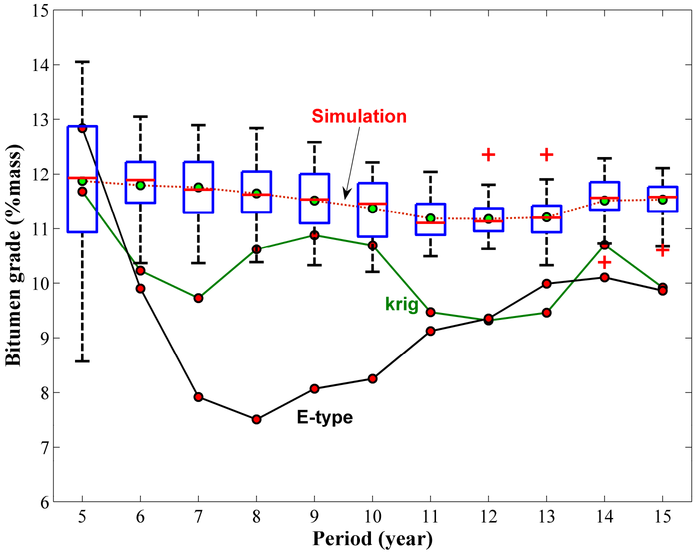


Figure 16. Box plot of simulation head grades (%mass), head grade for kriging, E-type and average simulation models.

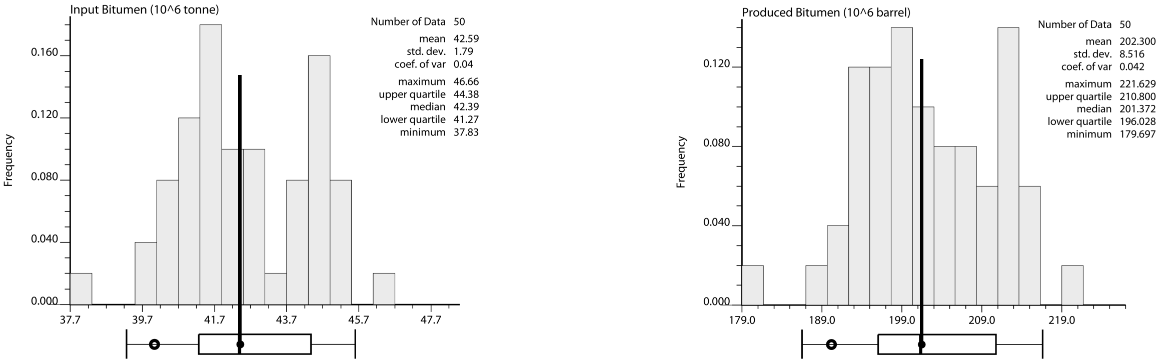


Figure 17. Histograms and box plots of tonnage of bitumen produced (a-left), number of barrels of sweet blend produced in million bbls (b-right). E-type result (circle), kriged result (black line).

Figure 17 illustrates the histograms and box plots for the bitumen as the final product in terms of million tonnes of bitumen and number of barrels of sweet blend produced. The very interesting aspect of these histograms is that the median of the total tonnage of bitumen produced for all the realizations is 42.39 million tonnes, which is almost equal to the tonnage of bitumen produced by the kriged model 42.42 million tonnes.

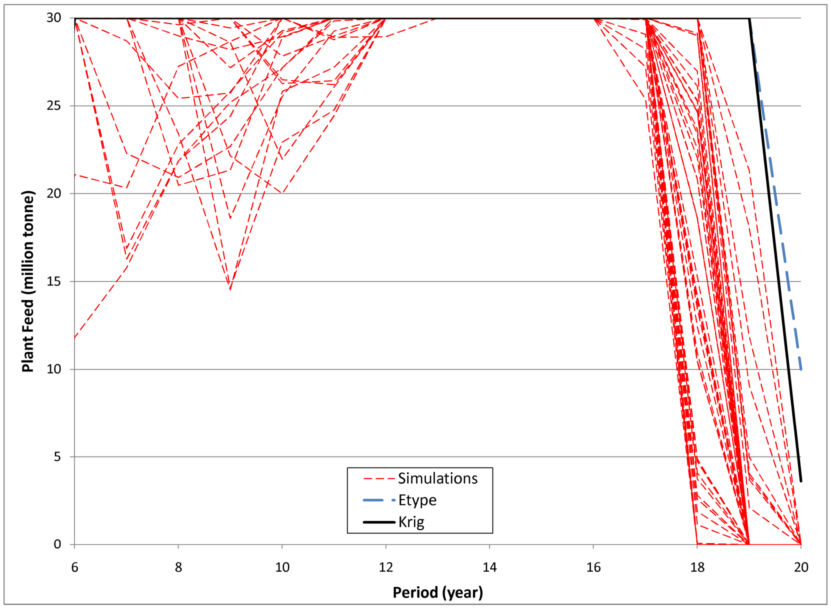


Figure 18. Plant feed (realizations red dash line), kriged values (black line) and E-type (blue dash line).

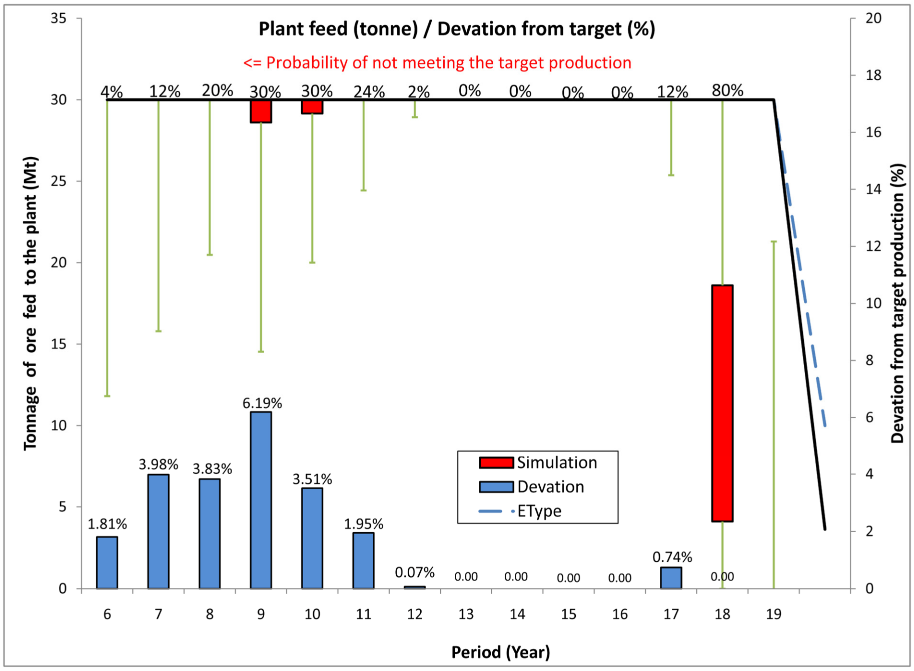


Figure 19. Box plot of the simulation plant feed (red box and green lines), kriged values (black line) and E-type (blue dash line) and deviation from target production.

Comparing the statistics from Figure 14 and Figure 17 reveals that the tonnage of bitumen produced by 424 million tonnes of ore with an average grade of 10% for the kriged model is equal to the median of all the realizations with a lower quartile 361 Mt and upper quartile of 382 Mt of ore with an average grade of 11.44%.

Figure 18 illustrates the plant feed through the mine life for the kriged, E-type, and all the realizations. The higher ore tonnage with lower average grade in the kriged and E-type models has resulted in almost two years longer mine life comparing to the simulation results (Figure 20b). A very important feature of the graph represented in Figure 18 is the target production not met in years 6 to 12 by some simulation realizations. One should take into account that the shortfall in production has happened although we have used four years of pre-stripping and a buffer stockpile. The effect of the grade uncertainty on the production targets would be more severe if the stockpile and pre-stripping strategy was not adopted. Figure 19 illustrates the box plot for the plant feed, the percentage deviation from the target production, and the probability that we would not meet the target production. Between years 8 to 11 the worst case happens where there is a chance of 20% to 30% that we would not meet the production targets. The maximum deviation from the 30 million tonnes target is at year 9 with 6.19% average deviation. Figure 19 clearly illustrates that what are the negative impacts of a deterministic approach to mine planning based on an estimated block model.

Figure 20a shows the histogram and box plot for the NPV, the kriged model NPV is not within the inter-quartile of the simulated models; this is because of the longer mine life and lower average grade for the kriged model. The comparison of the NPV of the kriged model and the simulation results under these circumstances is not a proper economic measure.

Figure 21 shows the cumulative net present value for the kriged, E-type, and simulation results. The longer mine life for the kriged and E-type results led to a flatter slope of the NPV cumulative curves, which has resulted in a smaller NPV compared to the simulation results. Figure 20a and Figure 21 should be analyzed together the median of for the simulation results is 1.8 billion dollars whereas the E-type has an NPV of 1.2 billion dollars compared to 1.5 billion for the kriged model.

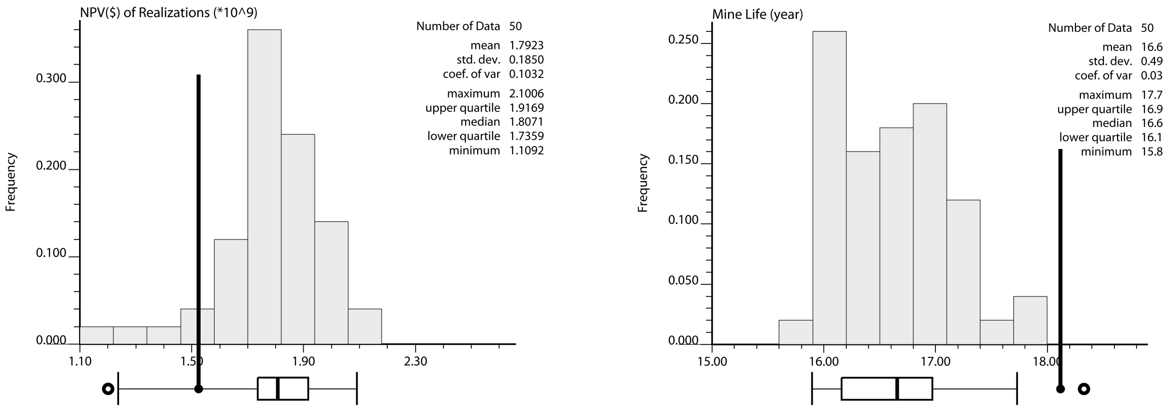


Figure 20. Histograms and box plots of NPV in billion dollars (a-left), and the mine life in years (b-right). E-type result by circle and kriged result by black line.

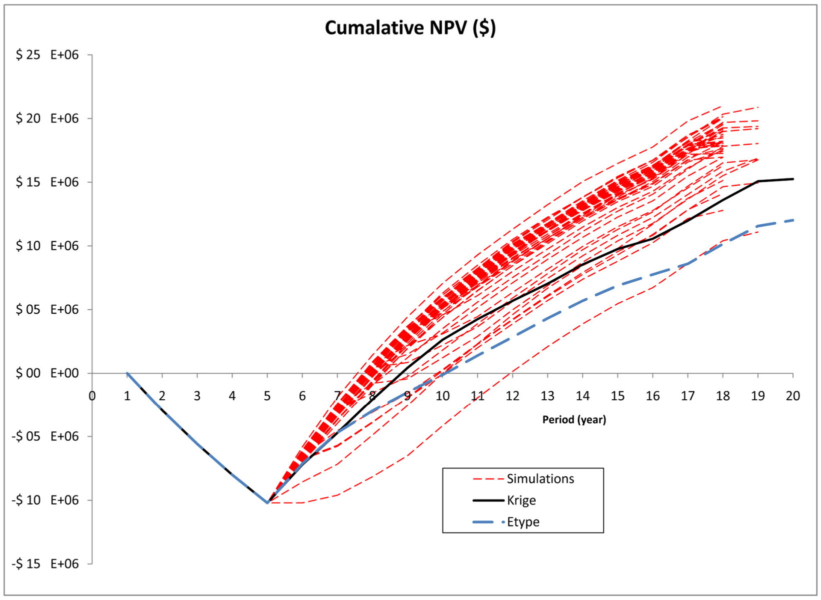


Figure 21. Cumulative discounted cash flow all over the periods for 50 realizations (red lines) and kriging values (black line), a close view at bottom.

4. Conclusion

A method is presented to transfer geological uncertainty into mine planning. First, Sequential Gaussian Simulation is used to generate fifty realizations of an oil sands deposit. An optimum final pit limits design is carried out for each SGS realization while fixing all other technical and economic input parameters. Afterwards, the long-term schedule of each final pit shell is generated. Uncertainty in the final pit outline, net present value, production targets, and the head grade are assessed and presented. The results show that there is significant uncertainty in the long-term production schedules. In addition, the long-term schedule based on one particular simulated ore body model is not optimal for other simulated geological models. The mine planning procedure is not a linear process and the mine plan generated based on the Kriging estimate is not the expected result from all of the simulated realizations.

5. References

[1] Deutsch, C. V. and Journel, A. G., (1998), "GSLIB geostatistical software library and user's guide", © Oxford University Press, New York,

[2] Dimitrakopoulos, R., Farrelly, C., and Godoy, M. C., (2002), "Moving forward from traditional optimisation: Grade uncertainty and risk effects in open pit mine design", *Transactions of the IMM, Section A Mining Industry*, Vol. 111, pp. A82-A89.

[3] Dimitrakopoulos, R. and Ramazan, S., (2003), "Uncertainty-based production scheduling in open pit mining", *in Proceedings of SME Annual Meeting*, © SME, pp. 355.

[4] Gemcom Software International, I., (1998-2008), "Whittle strategic mine planning software", ver. 4.2, Vancouver, B.C.: Gemcom Software International.

[5] Godoy, M. C. and Dimitrakopoulos, R., (2004), "Managing risk and waste mining in long-term production scheduling of open pit mine", *in Proceedings of SME Transaction*, © SME, pp. 43-50.

[6] Isaaks, E. H., (1990), "The application of Monte Carlo methods to the analysis of spatially correlated data, Ph.D. thesis, Stanford University", PhD Thesis, © Stanford University, Stanford, CA,

[7] Jaremko, D., (2009), "Project Status", *Oilsands review : the unconventional oil authority*, Vol. 4, 1, pp. 53-59.

[8] Lerchs, H. and Grossmann, I. F., (1965), "Optimum Design of Open-Pit Mines", *The Canadian Mining and Metallurgical Bulletin, Transactions*, Vol. LXVIII, pp. 17-24.

[9] Leuangthong, O., Schnetzler, E., and Deutsch, C. V., (2004), "Geostatistical Modeling of McMurray Oil Sands Deposits", 2004 CCG Annual Report Papers, © Centre for Computational Geostatistics, Edmonton, September 2004, 309-1 to 309-10.

[10] Smith, M. L. and Dimitrakopoulos, R., (1999), "The influence of deposit uncertainty on mine production scheduling", *International Journal of Surface Mining, Reclamation and the Environment*, Vol. 13, pp. 173-178.

Modeling variable pit slopes in the open pit production scheduling MILP formulation

Hooman Askari-Nasab and Kwame Awuah-Offei[1]
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

Open pit mine plans define the complex strategy of displacement of ore and waste over the mine life. The objective of the mine plan is to maximize the future cash flows within the technical and physical constraints. Various mixed integer linear programming (MILP) formulations have been used for production scheduling of open pit mines. In the MILP formulation the slope constraints guarantee that all the overlying blocks are mined prior to mining a given block. Traditionally, the slope constraints have been modeled with cone templates representing the required wall slopes of the open pit mine. The overall pit slopes constructed by these templates are a function of the width and height of the geological block model. Therefore, the overall slopes are fixed in one bearing. This paper presents a general MILP model for open pit mine scheduling with variable slopes constraints. The methodology utilizes a directed graph to capture the precedence of extraction of blocks and pit slopes in different bearings. Depth-first-search algorithm was used to traverse the graph for constructing the slope constraints in the MILP problem. TOMLAB/CPLEX was used as the implementation platform to efficiently integrate the mathematical solver package CPLEX with MATLAB. A case study on intermediate scheduling of an iron ore mine over twelve periods was carried out to validate the models.

1. Introduction

Mixed integer linear programming (MILP) mathematical optimization have been used by different researchers to tackle the long-term open-pit scheduling problem (Caccetta and Hill, 2003; Ramazan and Dimitrakopoulos, 2004; Dagdelen and Kawahata, 2007). The MILP models theoretically have the capability to consider diverse mining constraints such as multiple ore processors, multiple material stockpiles, and blending strategies. In this paper, we present a general MILP model for open pit mine scheduling with variable pit slopes constraints. The methodology utilizes a directed graph to capture the precedence of extraction of blocks and pit slopes in different bearings. Depth-first-search algorithm is used to traverse the graph for constructing the slope constraints in the MILP problem.

[1] Assistant Professor, Department of Mining & Nuclear Engineering, University of Missouri-Rolla, USA

2. MILP theoretical framework and models

In this paper, we extend our model based on the basic concepts of the MILP model presented by Ramazan and Dimitrakopoulos (2004). We revise the MILP model for better performance and introduction of variable pit slopes into the formulation. The parameters, decision variables, and indices used in the formulation are as follows.

Parameters

g_n^e average grade of element e in ore segment of block n.

$gu^{t,e}$ upper bound on grade (maximum grade) of element e in period t.

$gl^{t,e}$ lower bound on grade (minimum grade) of element e in period t.

o_n ore tonnage in block n.

w_n waste tonnage in block n.

u_n unknown waste tonnage in block n.

bt_n total block tonnage equal to $o_n + w_n + u_n$

$pu^{t,e}$ upper bound on processing capacity of ore containing element e in period t (tonnes).

$pl^{t,e}$ lower bound on processing capacity of ore containing element e in period t (tonnes).

mu^t upper bound on mining capacity in period t (tonnes).

ml^t lower bound on mining capacity in period t (tonnes).

$r^{t,e}$ processing recovery, is the proportion of element e recovered by processing it in period t.

$p^{t,e}$ price of product in present value terms obtainable per unit of product (element e) sold in period t.

$cs^{t,e}$ selling cost of product in present value terms per unit of product (element e) sold in period t.

$cp^{t,e}$ extra cost in present value terms per tonne of ore for mining the material as ore and processing element e in time period t.

cm^t cost in present value terms of mining a tonne of waste in period t.

dbv_n^t discounted block value of extracting block n in period t. dbv_n^t is the discounted cash flow generated by extracting block n in period t.

Indices

$t \in \{1,....,T\}$ Index for scheduling periods from 1 to T

$n \in \{1,...,N\}$ index for blocks form 1 to N

$e \in \{1,...,E\}$ index for elements of interest in each block

$\underline{\boldsymbol{\mathcal{N}}} = \{1,...,N\}$ set of all blocks in the block model (or set of all vertices in the directed graph G).

$\underline{\boldsymbol{\mathcal{A}}} = \{1,...,A\}$ set of all edges in the directed graph G ($\underline{\boldsymbol{\mathcal{N}}}, \underline{\boldsymbol{\mathcal{A}}}$).

Decision variables

b_n^t binary variable, equal to 1 if block n is to be mined in period t, otherwise 0. Where $b_k^t \in \{0,1\}$.

2.1. Calculating the discounted economic block value

In simple terms the discounted block value or the discounted profit of block n is calculated by equation (1).

discounted profit = discounted revenue - discounted costs (1)

in other words:

$$dbv_n^t = \underbrace{[\sum_{e=1}^{E} o_n \times g_n^e \times r^{t,e} \times (p^{t,e} - cs^{t,e})}_{discounted\ revenues} - \underbrace{\sum_{e=1}^{E} o_n \times cp^{t,e}] - [(o_n + w_n + u_n) \times cm^t]}_{discounted\ costs} \quad (2)$$

2.2. Mixed integer programming model

The objective function is to generate a schedule which will provide the order of extraction of blocks over the mine life. The goal of the schedule is to maximize the overall discounted cash flow of the mining project, while satisfying constraints such as: grade blending, mining capacity, processing capacity, precedence of extraction of blocks, and safe overall pit slopes, over the scheduling period. The MIP objective function is represented by equation (3) based on the assumption of one processing path. Multiple processing paths could be added to the model if necessary.

2.2.1 Objective function model 1

$$\max \sum_{t=1}^{T} \sum_{n=1}^{N} dbv_n^t \times b_n^t \quad (3)$$

Subject to:

2.2.2 Grade blending constraints

$$\sum_{n=1}^{N} (g_n^e - gu^{t,e}) \times o_n \times b_n^t \le 0 \qquad t \in \{1,...,T\}, \quad e \in \{1,...,E\} \quad (4)$$

$$\sum_{n=1}^{N} (g_n^e - gl^{t,e}) \times o_n \times b_n^t \ge 0 \qquad t \in \{1,...,T\}, \quad e \in \{1,...,E\} \quad (5)$$

These inequalities ensure that the grade of the elements of interest and contaminants are within the allowable range. This is controlled for all the blocks and mining cuts.

2.2.3 Processing constraints

$$\sum_{n=1}^{N} o_n \times b_n^t \le pu^t \qquad t \in \{1,...,T\} \tag{6}$$

$$\sum_{n=1}^{N} o_n \times b_n^t \ge pl^t \qquad t \in \{1,...,T\} \tag{7}$$

These inequalities ensure that the total ore processed in each period is within the acceptable range of processing plant capacity. The assumption here is that there is one process line. The model could be extended to multiple processes of different elements of interest.

$$\sum_{n=1}^{N} (o_n + w_n + u_n) \times b_n^t \le mu^t \qquad t \in \{1,...,T\} \tag{8}$$

$$\sum_{n=1}^{N} (o_n + w_n + u_n) \times b_n^t \ge ml^t \qquad t \in \{1,...,T\} \tag{9}$$

These inequalities ensure that the total tonnage of material mined in each period is within the acceptable range of mining capacity in that period. The mining capacity is a function of the capacity of mining equipment available and the possible contract mining equipment capacity.

2.2.4 Precedence of extraction

All the overlying blocks that must be mined prior to mining block n have to be determined. Traditionally the slopes have been implemented through one ore more cone templates representing the required wall slopes of the open pit mine. In this study we are going to model the required wall slopes by a directed graph representing the order of extraction of blocks.

$$J \times b_n^t - \sum_{j=1}^{J} \sum_{i=1}^{t} b_j^i \le 0 \qquad \forall n \in \{1,...,N\}, \quad t \in \{1,...,T\}, \quad j \in P(J) \tag{10}$$

Using equation (10) will result in one constraint per block n per period t.Where $P(J) \subset \underline{\boldsymbol{N}}$ is a set of all the blocks that must be extracted prior to mining block n to ensure that block n is exposed for mining and to maintain the desirable safe slope.

Where:

- J is the total number of blocks overlying block n. In other words J is the number of blocks in set $P(J)$.
- i is the counter for scheduling periods.

The other method is to use J number of constraints for block n per period t as represented by equation (11).

$$b_n^t - \sum_{i=1}^{t} b_j^i \leq 0 \qquad \forall n \in \{1,...,N\}, \quad t \in \left\{1,...,T\right\}, \quad j \in P(J) \; (11)$$

Evidently using equation (11) will increase the number of inequality constraints in the MIP formulation. We will develop a variable slope model for setting up the $P(J)$ set, using directed graph theory, based on equation (10).

2.2.5 Reserve constraints

We assume that a final pit limits is superimposed on the block model and we are going to schedule the extraction of all the blocks in the model. In other words, all the blocks within the pit outline are going to be extracted.

$$\sum_{t=1}^{T} b_n^t = 1 \qquad \forall n \in \{1,...,N\} \qquad (12)$$

The MIP model presented above could be used as a final pit limits optimization tool. The economic block model would be the input into the MIP model. The MIP optimization will generate a schedule and the final pit limits at the same time. To achieve this goal, the reserve constraint demonstrated by equation (12) should be defined as an inequality constraint as demonstrated in equation (13).

$$\sum_{t=1}^{T} b_n^t \leq 1 \qquad \forall n \in \{1,...,N\} \qquad (13)$$

$$b_n^t \in \{0,1\} \qquad \forall n \in \{1,...,N\}, \quad t \in \left\{1,...,T\right\} \qquad (14)$$

3. Precedence of extraction and pit slope modeling using directed graphs

We will focus on the techniques of how to construct the set of blocks $P(J)$ as demonstrated in equation (10). Let's denote $P(J)$ as the set of all the blocks that must be extracted prior to mining block n to ensure that block n is exposed for mining and to maintain the desirable overall safe slope. We will use a directed graph to model the precedence of extraction between blocks. We defined a directed graph G ($\underline{\boldsymbol{\mathcal{N}}}$, $\underline{\boldsymbol{\mathcal{A}}}$) by the set of vertices, $\underline{\boldsymbol{\mathcal{N}}}$ (blocks); connected by ordered pairs of elements called arcs, $\underline{\boldsymbol{\mathcal{A}}}$. Each vertex (block) has an index $\{1,...,N\}$ and it carries other block associated information such as tonnage, grade, block economic value, etc. Each arc $a = (i, j)$ is considered to be a directed arc from block i to block j, where $i \; and \; j \in \{1,...,N\}$; j is called the head and i is called the tail of the arc; j is said to be a direct successor of i, and i is said to be a direct predecessor of j. If a path made up of one or more successive arcs leads from i to j, then j is said to be a successor of i, and i is said to be a predecessor of j.

There are several data structures used for graph realization. The most popular approaches are, the edge list structure, the adjacency list structure, and the adjacency matrix. We illustrate the adjacency matrix approach with an example. To model the block extraction precedence and the overall pit slopes an adjacency matrix represented by matrix $A(N \times N)$ is defined. The non-diagonal entry $A(i, j)$ is equal to one if there is an arc from vertex i to vertex j, and the diagonal entry $A(i,i)$ is equal to zero, or in other words, the number of arcs from vertex i to itself is zero.

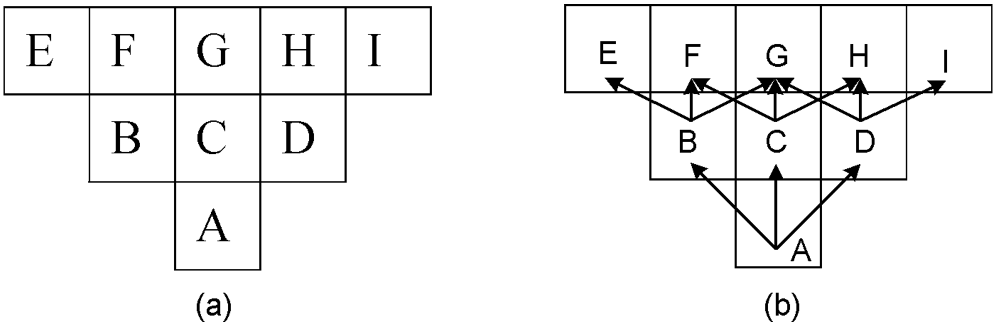


Figure 1 – Directed graph of extraction precedence

Figure 1-a illustrates a two dimensional example of a set of blocks labeled from A to I alphabetically. The set of blocks $\{B,C,D,E,F,G,H,I\}$ comprise the entire predecessor set of blocks that must be removed prior to extraction of A. The set of blocks $\{B,C,D\}$ include the immediate predecessor set. Figure 1-b demonstrates the order of removal of blocks by pairs of arcs. To remove block A the minimal set of blocks that are required to be removed prior to extraction of block A is captured by a directed graph connecting each block to at least to three blocks immediately above it. The real problem is in three-dimension and $P(J)$ has the shape of an inverted cone; each block is at least connected to the nine immediate blocks above. The number of immediate blocks connected to a block for defining variable slopes and accurate slope modeling would be considerably more than nine blocks.

	A	B	C	D	E	F	G	H	I
A	0	1	1	1	0	0	0	0	0
B	0	0	0	0	1	1	1	0	0
C	0	0	0	0	0	1	1	1	0
D	0	0	0	0	0	0	1	1	1
E	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0

Figure 2 - Adjacency matrix for example in Figure 1.

Figure 2 demonstrates how the adjacency matrix is constructed for the example in Figure 1. If there is an arc from the vertex index i to vertex j then cell $A(i,j)$ would be set to one. The direction of the arcs are mapped from rows to columns. For instance there is an arc from C to F represented by $A(C,F)=1$, but there is no arc from F to C represented by $A(F,C)=0$.

Initially a slope profile is defined with the safe overall slopes in different regions defined by two variables: the desired slope and azimuth. For each slope profile, the slope requirements are converted into an inverted cone that defines the total amount of rock that must be mined to. At each azimuth specified, the cone has the required slope.

Figure 3 illustrates a two dimensional view of an inverted cone with a 45 degree slope all around. The desired 45 degree slope is achieved by using directed arcs constructed by only the nine immediate blocks on top of each block. When the nine arcs are applied to all the blocks, chaining the block dependencies by the adjacency matrix will result in every block in the inverted cone to be mined.

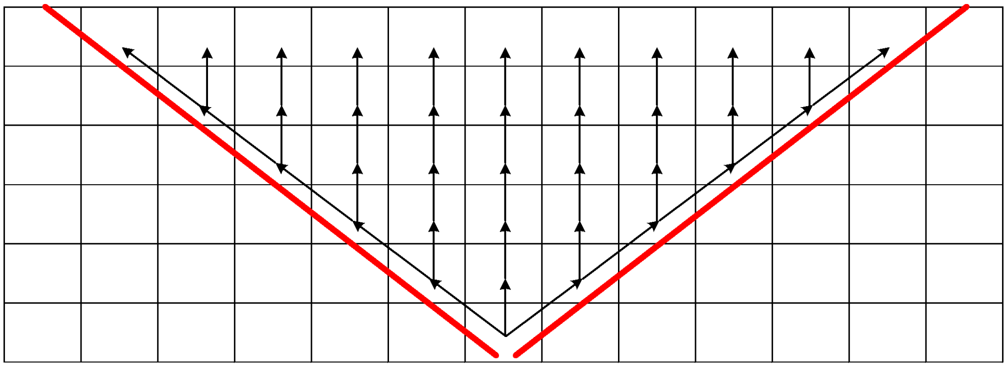


Figure 3 – Inverted cone and the directed graph for 45 degree slope.

Figure 4 demonstrates how variable slopes are achieved in different slope profiles and different azimuths. To achieve the desired slope the immediate predecessor blocks, which construct the directed graph would be more or less than the nine immediate blocks discussed in Figure 3. The more benches examined above for constructing the directed graph, the more accuracy in slope reproduction would be achieved.

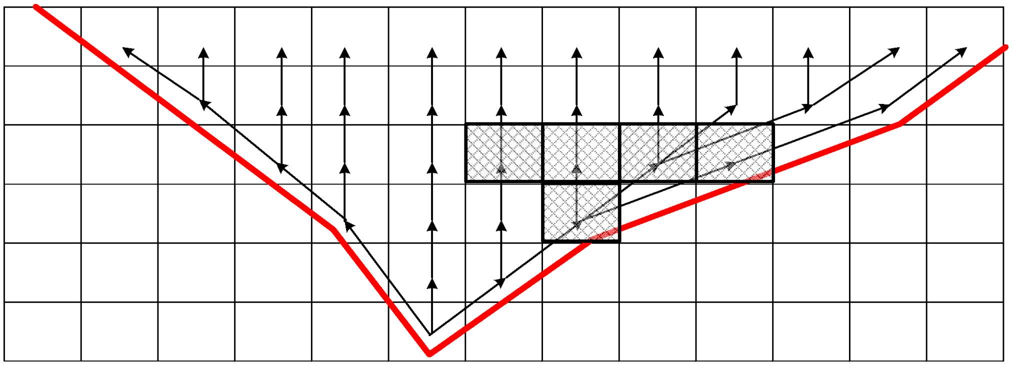


Figure 4 – Inverted cone and the directed graph defining variable slopes.

We have used TOMLAB/CPLEX as the implementation platform to efficiently integrate the mathematical solver package CPLEX with MATLAB. MATLAB has a built in graph type: the sparse matrix. We have also used the MatlabBGL package (Gleich, 2006) for working with graphs in MATLAB. One of the powerful implementations of graph data structures and algorithms is the Boost Graph Library (BGL) (Siek et al., 2002). It contains efficient algorithms implemented as generic C++ template specifications. The MatlabBGL library uses the MATLAB sparse matrix as a graph type. MatlabBGL also adds a wide range of graph algorithms to MATLAB environment by wrapping the Boost Graph Library algorithms with functions which are callable from MATLAB.

The precedence of extraction sparse adjacency matrix is constructed using the inverted cone templates with variable slopes. The sparse adjacency matrix is used to create the inequalities constraints expressed in equation (10). If the adjacency matrix illustrated in Figure 2 is examined more closely one would find that each block is connected to the immediate predecessor blocks through directed arcs. In equation (10), $P(J)$ is the set of all the blocks that must be extracted prior to mining block n. For instance, in Figure 1 the set of blocks $\{B,C,D,E,F,G,H,I\}$ comprise the entire predecessor set of blocks that must be removed prior to extraction of A. To be able to generate the set $P(J)$ out of the adjacency matrix represented in Figure 2, a traversing or search algorithm called depth-first-search (DFS) from the MatlabBGL package is used. DFS is an algorithm for traversing or searching a tree, tree structure, or graph. One starts at the root selecting the current block as the root in the directed graph (block A in our example) and explores as far as possible along each branch before backtracking. Formally, DFS is an uninformed search that progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found, or until it hits a node that has no children. Then the search backtracks, returning to the most recent node it hasn't finished exploring.

Therefore, to construct the inequalities constraints expressed in equation (10), the DFS algorithm is called to search and determine the predecessor set of blocks through adjacency matrix. Afterwards, based on the $P(J)$ set, inequalities constraints of equation (10) are constructed. The scale of the MIP formulation presented in section 2.2 starts exceeding the capacity of conventional mathematical optimization tools very quickly. This is due to the large number of constraints that are generated by equation (10). As a remedy to the large number of constraints new MILP formulation are required to be able to tackle the long-term mine planning problem. The new model needs to take into account the smaller set of the immediate predecessor blocks instead of the complete set of blocks covering each block.

4. Results and discussion of iron ore mine case study

A case study of scheduling a push-back of an iron ore deposit was carried out to verify and validate the models. The blocks within the push-back are scheduled in twelve periods. Three types of ore; top magnetite, oxide, and bottom magnetite are classified in the deposit. The block model contains the estimated magnetic weight recovery (MWT%) of iron ore; the contaminants are phosphor (P%) and sulphur (S%). The blocks in the geological model represent a volume of rock equal to $50m \times 25m \times 15m$.

The objective function aimed to maximize the net present value with a discount rate of 10% per period. TOMLAB/CPLEX (Holmström, 1989-2009) was used for implementation and solving the MILP formulation. Table 1 summarizes the total tonnage of ore and waste material in the push-back, and average grade of ore and contaminants. Table 1 also illustrates the mining, processing, and blending constraints imposed to the scheduling problem over 12 periods. The maximum allowable average of sulphur in each period is 1.8%, whereas the average grade of acceptable phosphor is 0.14%.

2598 blocks (integer variables) were scheduled over 12 periods this made a coefficient matrix, A, defined by equations (3) to (14) of a size $A(280,944 \times 62,352)$ with 1,797,352 nonzero elements. The CPLEX solver found a solution within 2% gap of the theoretical optimal solution. Figure 5 illustrates the extraction schedule generated by the MILP formulation of the same cross section 98400m.

Table 1 – Description of the push-back

Description	Value	Description	Value
Total tonnage of rock	155,295,000 tonnes	Maximum mining capacity	13 Mt per period
Total ore tonnage	84,059,000 tonnes	Maximum processing capacity	7.15 Mt per period
Total tonnage of contained Fe	61,811,000 tonnes	Sulphur grade (S%) allowed	1.8% per period
Average grade of MWT%	73.5%	Phosphor grad (P%) allowed	0.14% per period
Average grade of sulphur	1.5%	Number of scheduling periods	12

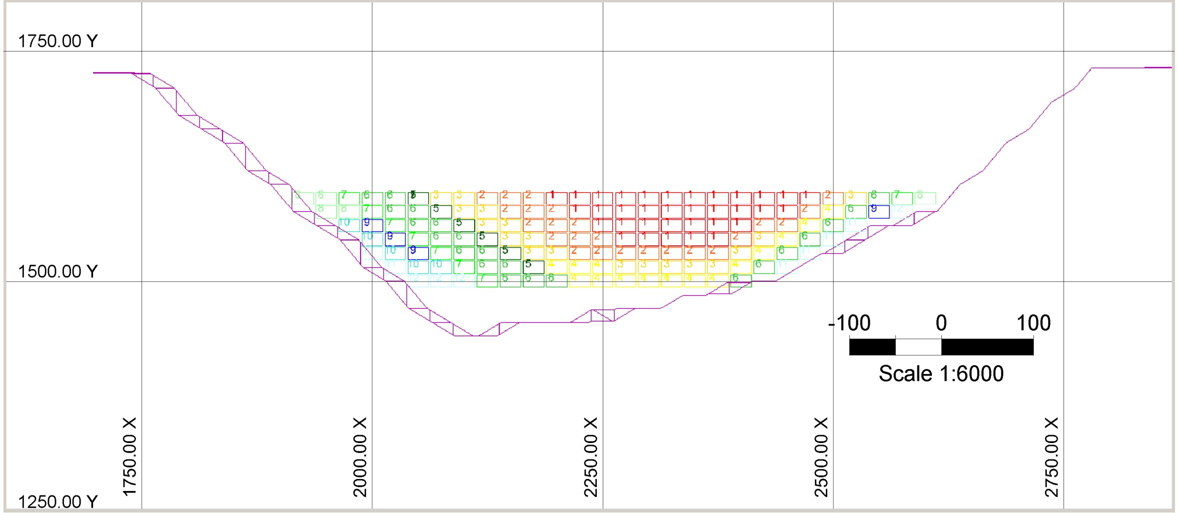


Figure 5 – Cross section 98400 showing the push-back and the schedule, looking east (meters).

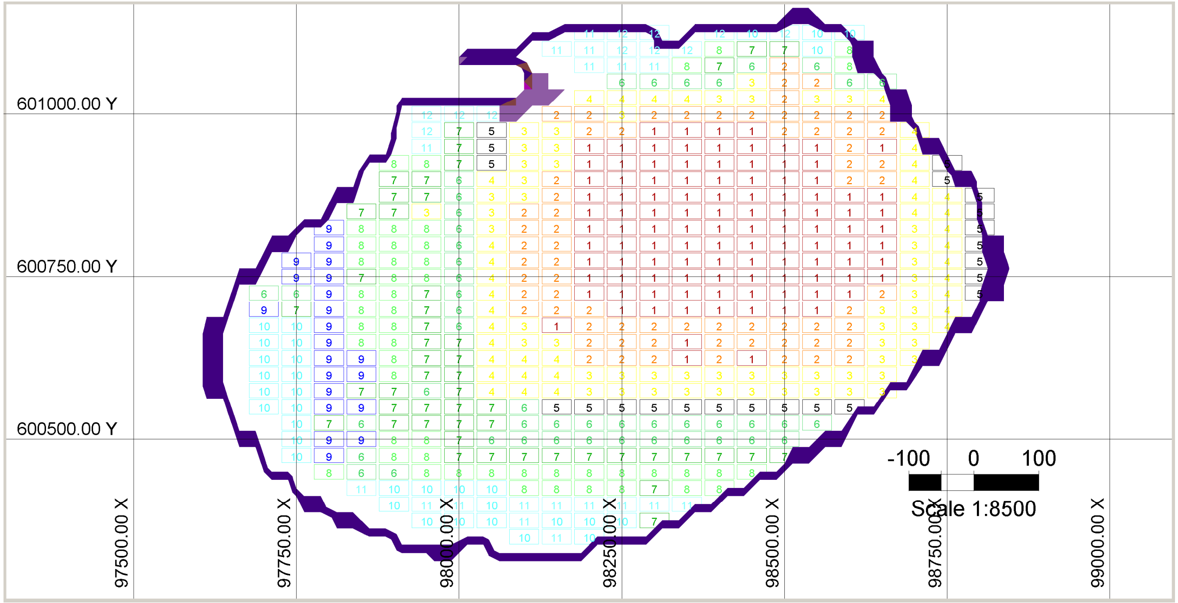


Figure 6- Plan view of schedule on bench 1570 (metes).

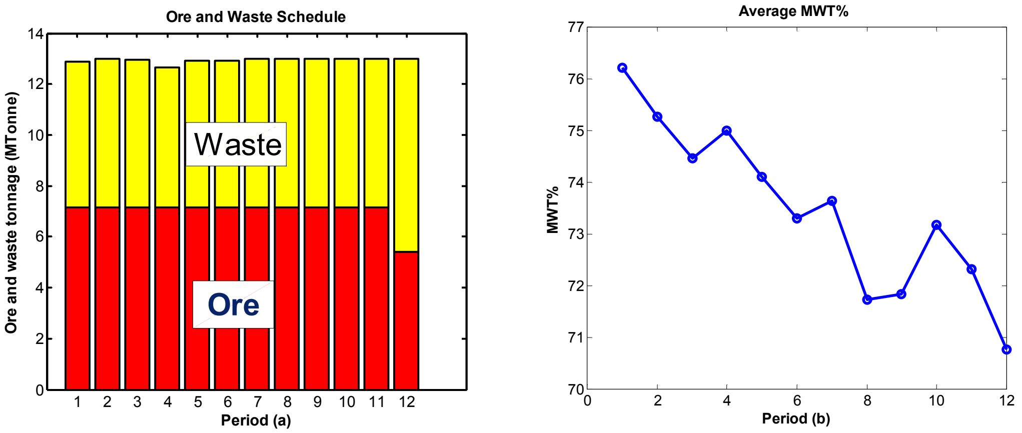


Figure 7 – (a) scheduled ore and waste over twelve periods; (b) average grade of ore in each period.

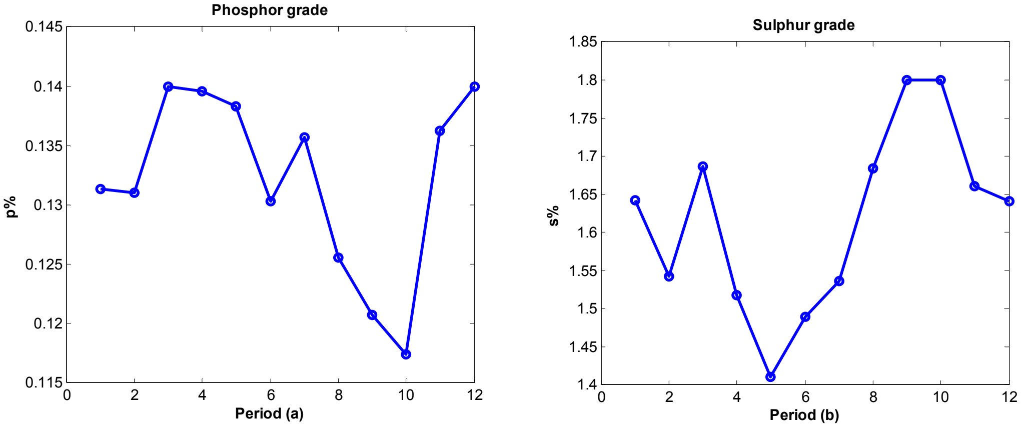


Figure 8 – (a) average grade of phosphor in each period; (b) average grade of sulfur in each period.

Since the MILP model is maximizing the net present value, it has high graded in early years to maximize the cash flows in the early periods (Figure 7b). Figure 8a and Figure 8b illustrate the allowable maximum grade of deleterious material, sulphur 1.8% and phosphor 0.14% are met.

5. Conclusions

The applications of the MILP model developed in this study showed that it has the capability of generating production schedules within a close gap to the theoretical optimal net present values for mining operations. Too many binary variables are required to formulate a life-of-mine schedule for a typical mine with the formulation presented in this study. It becomes almost impossible to solve such a problem with current state of optimization solvers. As the future work, we will focus on reformulating the problem with reduced number of integer variables and complexity.

6. References

[1] Caccetta, L. and Hill, S. P., (2003), "An application of branch and cut to open pit mine scheduling", *Journal of Global Optimization*, Vol. 27, November, pp. 349-365.

[2] Dagdelen, K. and Kawahata, K., (2007), "Opportunities in Multi-Mine Planning through Large Scale Mixed Integer Linear Programming Optimization", *in Proceedings of 33rd APCOM*, © GECAMIN LTDA, Santiago, Chile, pp. 337-342.

[3] Gleich, D., (2006), "MatlabBGL", ver. 2008-10-21: Version 4.0: David Gleich.

[4] Holmström, K., (1989-2009), "TOMLAB /CPLEX - v11.2", ver. Pullman, WA, USA: Tomlab Optimization.

[5] Ramazan, S. and Dimitrakopoulos, R., (2004), "Traditional and new MIP models for production scheduling with in-situ grade variability", *IJSME*, Vol. 18, 2, pp. 85-98.

[6] Siek, J., Lee, L.-Q., and Lumsdaine, A., (2002), "The boost graph library : user guide and reference manual", © Addison-Wesley, Boston, Pages xxiv, 321 p.

An agent based framework for open pit mine planning

Hooman Askari-Nasab and Kwame Awuah-Offei[1]
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

Long term production scheduling optimization has been a challenging issue for the mining industry because of the size and complexity of the problem. The current planning algorithms have limitations addressing the stochastic variables underlying the mine planning problem. In this paper an intelligent agent-based mine planning framework based on reinforcement learning is introduced. The long term mine planning is modeled as a dynamic decision network. The intelligent agent interacts with the block model by means of stochastic simulation and employs Q-learning algorithm to learn the sequence of push-backs that maximizes the net present value of the mining operation. The intelligent open pit simulator, IOPS, was implemented with an object oriented design in Java®. A comparative application case study was carried out to verify and validate the models. The proposed method was used in planning an iron ore deposit and the results were compared to the Milawa scheduler used in Whittle® software. The outcome of the study demonstrated that the intelligent agent framework provides a powerful basis for addressing real size open pit mine planning problems.

1. Introduction

The mining industry is faced with ever increasing complexities due to intense global competition, lower grade mineral deposits, price volatility, and geological uncertainty. More rigorous algorithms and enhanced numerical techniques are required to overcome the complexities currently facing the mining industry. The mine planning process defines the ore body depletion strategy over time. The planning of an open pit mine considers the temporal nature of the exploitation to determine the sequence of block extraction in order to maximize the generated income throughout the planning period. The optimal plan must determine the optimized ultimate pit limits and the mining schedule but such an objective results in a computationally intractable problem. Whittle (1989) outlined the complexity of the problem as: (i) the pit outline with the highest value cannot be determined until the block values are known; (ii) the block values are not known until the mining sequence is determined; and (iii) the mining sequence cannot be determined unless a pit outline is

[1] Assistant Professor, Department of Mining & Nuclear Engineering, University of Missouri-Rolla, USA

available. The optimal final pit limit algorithms conventionally neglect the time dimension of the problem and search for an ultimate contour that maximizes the total sum of the profits of all the blocks in the contour. The extraction sequence is then decided within the predetermined final pit limits. The optimized schedule cannot be attained without examining all possible combinations and permutations of the extraction sequence. Therefore, the scheduling algorithms must be able to deal with limitations of computing resources, time and space.

Open pit mine planning studies typically have focused on one of two objectives: (i) maximization of the discounted present value of cash flows (Tolwinski and Underwood, 1992; Elveli, 1995; Erarslan and Celebi, 2001; Halatchev, 2005; Dagdelen and Kawahata, 2007), or (ii) optimization of the plant feeding conditions (Youdi et al., 1992; Chanda and Dagdelen, 1995; Rubio, 2006; Yovanovic and Araujo, 2007). Current production scheduling methods are not just limited to, but can be divided into: heuristic methods; parametric analysis; operations research methods; and artificial intelligence techniques. The most common operations research methods include: mixed integer programming (MIP) (Gershon, 1983; Dagdelen, 1985; Ramazan and Dimitrakopoulos, 2004; Dagdelen and Kawahata, 2007), dynamic programming (Onur and Dowd, 1993) , goal programming (Chanda and Dagdelen, 1995; Esfandiari et al., 2004), and branch and bound techniques (Caccetta and Hill, 2003). Mixed integer programming mathematical optimization models have the capability to consider multiple ore processors and multiple elements during optimization. This flexibility of mathematical programming models result in production schedules generating significantly higher net present value than those generated by the other traditional methods. However, MIP formulations for optimization of production scheduling require too many binary variables, which makes the MIP models almost impossible to solve for actual open pit mining operations (Ramazan et al., 2005). Artificial intelligence methods such as machine learning expert system concepts (Tolwinski and Underwood, 1992; Elveli, 1995); genetic algorithms (Denby and Schofield, 1994; Denby et al., 1996; Wageningen et al., 2005); and applications of neural networks (Achireko and Frimpong, 1996; Frimpong and Achireko, 1997) have also been used to address the mine planning problem.

The key limitations of current mine planning methods are (i) inability to solve actual size mine problems; (ii) limitation in dealing with stochastic processes governing ore reserves, commodity price, cut-off grade, and production costs; (iii) inadequacy of the current final pit limits optimization techniques in taking into account the time aspect of exploitation; and (iv) shortcoming in defining the economics of ore with respect to the economics of the entire mining process, from ore to the finished product.

Research advances have led to concrete proposals and early applications of intelligent agents in mine planning and design (Askari-Nasab et al., 2005; Askari-Nasab and Szymanski 2007). The primary objective of this paper is to review the development of an intelligent agent-based theoretical framework for real size open pit mine planning. The study is a hybrid research work comprising algorithm development based on reinforcement learning concepts (Watkins, 1989; Sutton and Barto, 1998), and algorithm implementation in Java® programming language. A stochastic simulation model based on modified elliptical frustum (Askari-Nasab et al., 2004; Askari-Nasab et al., 2007) has been developed and used to model the geometry of the open pit layout expansion. The simulator returns the amount of ore, waste and the annual cash flow of the operation. The long term planning of the open pit mine is modeled as a dynamic decision network. The intelligent agent interacts with the open pit environment through simulation and employs Q-learning algorithm (Watkins, 1989) to maximize the net present value of the mining operation. The developed algorithms are implemented and applied to a real-world mining operation. The numerical applications of the developed models are compared with the results of common software used in industry to verify and validate the models. Finally, the potential application of the mine planning framework and significance of the research in mine planning is discussed.

2. Intelligent open pit planning theoretical framework

The reinforcement learning problem is formalized by the interaction of two basic entities: the agent and the environment. The agent is the learner and decision-maker. The agent's environment is comprised of everything that it cannot completely control. Thus, the environment defines the task that the agent is seeking to learn. A third entity, the simulation, mediates the interactions between the agent and the environment. The agent takes sensory input from the environment, and produces output actions that affect it. The interaction is usually an ongoing non-terminating process (Sutton and Barto, 1998).

Figure 1 illustrates the intelligent open pit optimal planning conceptual framework based on reinforcement learning terminology. The intelligent planning framework comprise independent, interactive and interrelated subsystems with processes, using reinforcement learning as the main engine to maximize the net present value of mining operations. The model illustrated in Figure 1 consists of three main entities of the reinforcement learning problem, agent, environment, and simulation. The main integral parts of the theoretical framework are as follows: (i) environment: consists of geological block model and economic block model; (ii) simulation: open pit production simulator that captures the discrete dynamics of open pit layout expansion, and materials transfer with the respective annual cash flows. The simulation model consists of a number of interrelated subsystems. The development and performance of the simulation components are discussed in (Askari-Nasab et al., 2004; Askari-Nasab, 2006; Askari-Nasab et al., 2007); (iii) agent: The simulated results are transferred to the intelligent open pit agent where Q-learning algorithm (Watkins, 1989) serves as the engine. The production simulator passes the respective amount of ore, waste, and the cash flows of the production periods to the agent. Development of the intelligent agent mine planning architecture is based on mathematically idealized forms of the reinforcement learning problem. The main concepts of optimality and the models in this study are developed and adapted from Sutton & Barto (1998) and Wooldridge (2002).

The reinforcement learning problem is meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The intelligent planning agent interacts with the block model through the production simulator and selects actions that are defined in terms of the changes in the push-back parameters and as the result, changes in the pit geometry. The simulation and the block model respond to those actions and present new possible pit push-backs to the agent. The open pit dynamics simulator in conjunction with the block model returns numerical rewards, which is the cash flow of each simulated production period. The primary goal of the agent is to maximize the NPV of the operation over time. This means maximizing not only the immediate reward, which is the cash flow of the next production period, but also the cumulative reward in the long run, which is the NPV.

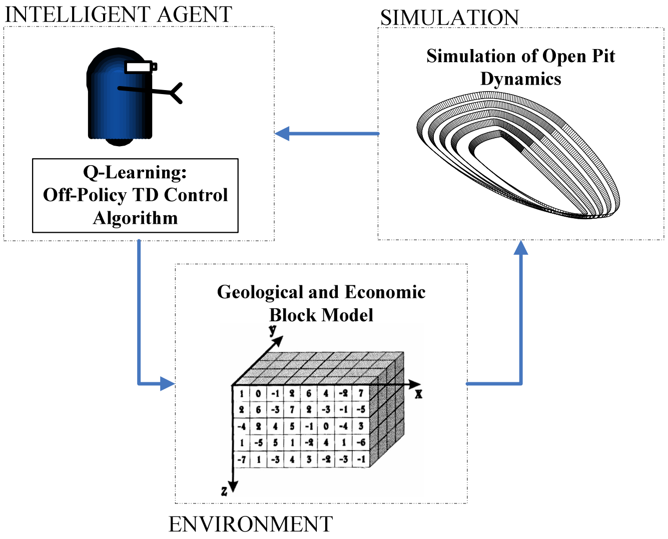


Figure 1- Intelligent open pit optimal planning frame work.

Figure 2 illustrates the mine planning intelligent agent architecture. The pit geometry evolution is viewed as series of snapshots over time. The agent and the simulation interact at each sequence of discrete time steps, $t = 1,...,n$. The simulation of the mining operation starts with the initial box cut at state, $s_{t-1} \in S_t$, and the agent responds by choosing the next pushback, $a_{t-1} \in A_t$, to be performed in this stage. Where S is the set of possible push backs, and $A(s_t)$ is the set of changes possible in the pit geometry in state s_t .

As a result of this action, the simulation and environment can respond with a number of possible states. However, only one state will actually result. On the basis of this second state of the environment, the agent again chooses an action to perform. The environment responds with one of a set of possible actions available, the agent then chooses another action, and so on. More specifically, the learning agent and simulation interact at each of a sequence of discrete time steps. At each time step t, the agent receives some representation of the open pit state, $s_t \in S$. On the basis of S , the agent selects an action, $a_t \in A(s_t)$, One time step later, in part as a consequence of its action, and interaction with the block model the agent receives a numerical reward, which is the cash flow of that period of mining operation, $r_{t+1} \in R$. As the result the agent finds itself in a new state, s_{t+1} . At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. This mapping is called the agent's policy and is denoted by, π_t , where $\pi_t(s,a)$ is the probability that $a_t = a$ if $s_t = s$.

Reinforcement learning methods specify how the agent changes its policy as a result of its experience. The agent's goal, roughly speaking, is to maximize the total amount of reward it receives over the long run. The objective is to maximize the *expected return*, where the return (see Figure 2), R_t given by Equation (1), is defined as a specific function of the immediate reward sequence. In Equation (1), γ is the discount factor and is a number between 0 and 1. The discount factor describes the preferences of an agent for current rewards over future rewards. When γ is close to 0, rewards in the distant future are viewed as insignificant. i in Equation (2) is the interest rate for time slice, t.

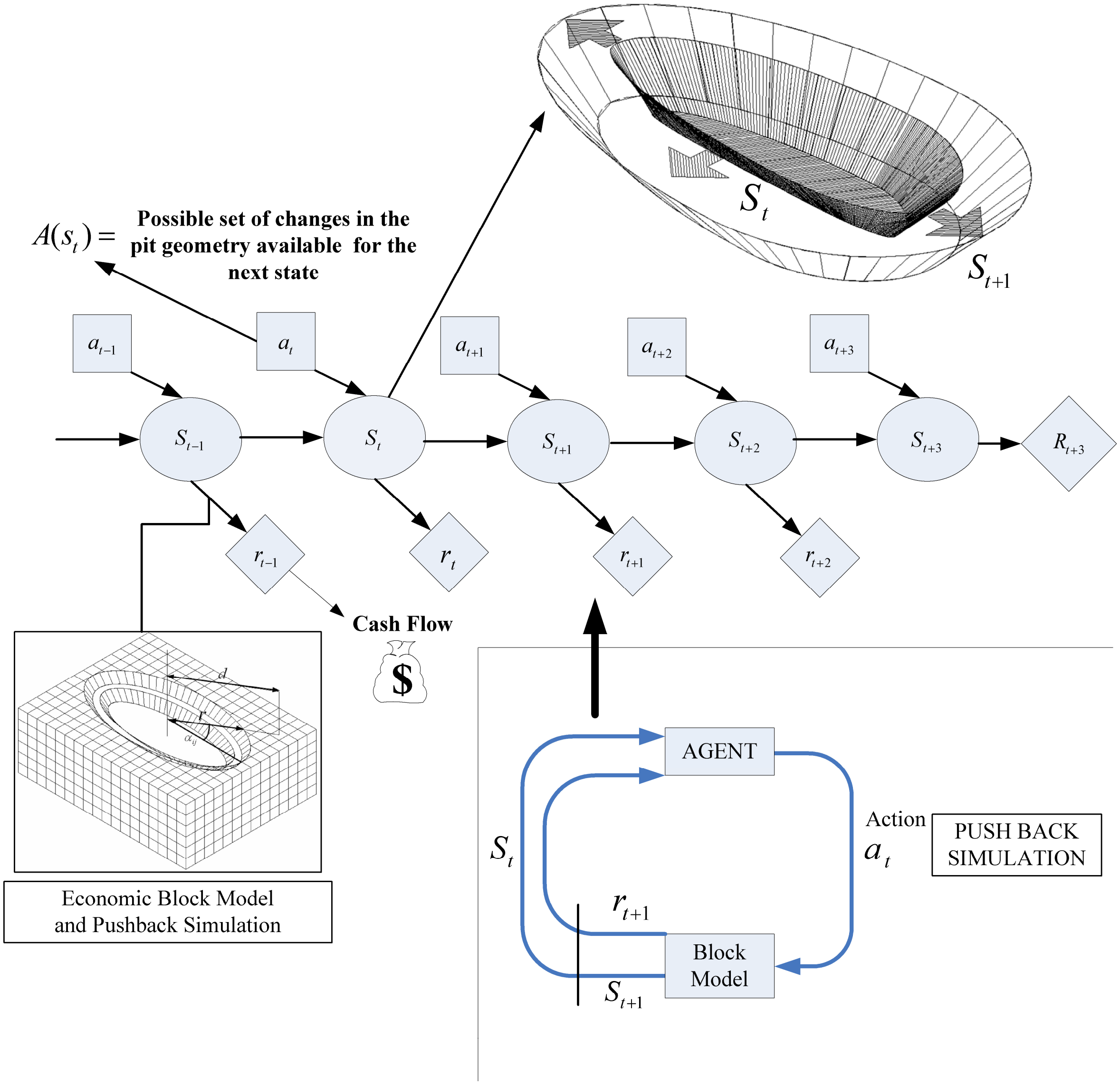


Figure 2 - Intelligent mine planning agent model as reinforcement learning problem.

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots + \gamma^k r_{t+k+1} = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \tag{1}$$

$$\gamma = \frac{1}{1+i} \tag{2}$$

Almost all reinforcement learning algorithms are based on estimating *value functions*--functions of states that estimate how good it is for the agent to be in a given state or how good it is to perform a given action in a given state. The notion of "how good" here is defined in terms of expected return. Accordingly, value functions are defined with respect to particular policies. Figure 3 illustrates a schematic of the open pit simulation at a discrete time step t and the open pit current status of S. For clarity of illustration it is assumed that there are just three possible push-backs a_1, a_2, a_3 that satisfy the targets of the next production period. Following one of the push-back designs the open pit will expand to the status of$s_1^{'}, s_2^{'}$, or $s_3^{'}$. The *value* of state$s$under policy π, denoted by $V^{\pi}(s)$, is the expected return or the NPV, when starting in s and following the policy thereafter, until reaching the final pit limits. For the Markov Decision Process representing the open pit dynamics in Figure 2, $V^{\pi}(s)$ can be defined as Equation (3).

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\{\textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\} \tag{3}$$

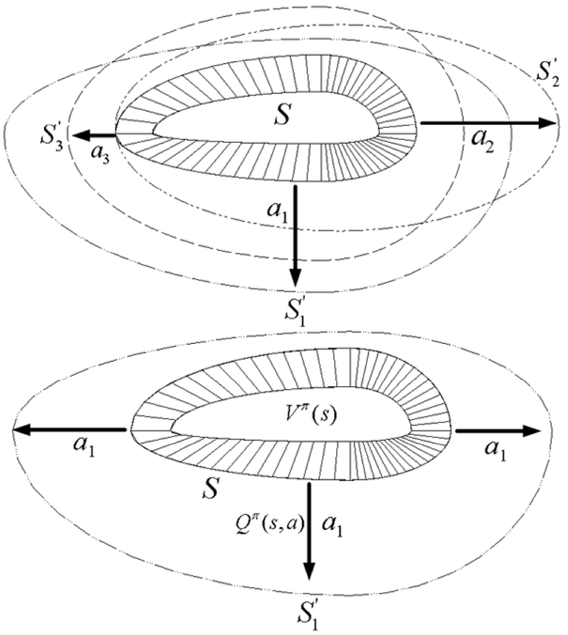


Figure 3 - Schematic of open pit simulation at a discrete time step t.

$E_{\pi}\{\ \}$ denotes the expected NPV given that the agent follows policy π, and t is any time step. The policy π is the current production schedule. The function V^{π} is called the *state-value function for policy* π. Similarly, the value of taking action a in state s under a policy π, denoted $Q^{\pi}(s,a)$ is defined as the expected NPV of the operation starting from s, taking the action a, and thereafter following the current schedule (policy π). Q^{π} is called the *action-value function for policy* π given by Equation (4).

$$Q^{\pi}(s,a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\} = E_{\pi}\{\textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\} \tag{4}$$

The Q-learning algorithm (Watkins, 1989) is used in this study to directly approximate Q^{π}, the optimal mine pushback design.

3. Algorithm development

Figure 4 illustrates the detailed flow chart of the intelligent optimal mine planning algorithm based on Q-learning algorithm (Watkins, 1989). The steps of the algorithm are as follows:

Step 1

The algorithm starts with (i) arbitrarily initializing the $Q(s,a)$, which is the expected discounted sum of future monetary returns of expanding the open pit from status S to the $S^{'}$ by choosing the push-back a and following an optimal policy thereafter; (ii) set the number of simulation trials that the algorithm is run. In other words the number of times that the open pit dynamics are being simulated from the initial box cut to the final pit limits.

Step 2

The push-back simulator captures the open pit layout evolution as a result of the material movement. At this stage the algorithm stochastically simulates a number of practical push-back designs for the next production period. The result of the simulation is k push-backs $a_1, a_2, ..., a_k$ that satisfy the tonnage production of the next period. Following each of these push-backs $a_1, a_2, ..., a_k$, the open pit will expand to the status of $s_1^{'}, s_2^{'}, ..., s_k^{'}$. The value of state s under policy π, denoted $V^{\pi}(s)$ is the expected return or the NPV of the sequence, when starting in s and following the policy thereafter until reaching the final pit limits.

Step 3

Simulated push-backs $a_1, a_2, ..., a_k$ are fitted on the economic block model, where the cash-flows $r_1, r_2, ..., r_k$ of each push-back are returned to the program.

Step 4

The epsilon greedy algorithm is called. The action selection rule is to select the action or one of the actions with highest estimated action value, that is, to select the push-back at time step t with the highest cash flow. The algorithm behaves greedily most of the time, which means it will select a push-back with the highest cash-flow among $r_1, r_2, ..., r_k$. But every once in a while, say with small probability ε, instead the algorithm selects an action at random, independently of the action-value estimates of the push-back. Subsequently the chosen push-back is implemented and the agent finds itself in pit status $S^{'}$ and observes the cash flow r.

Step 5

After being initialized to arbitrary numbers in step 1, Q-values $Q(s,a)$ are updated based upon previous experience as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a^{'}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \qquad (5)$$

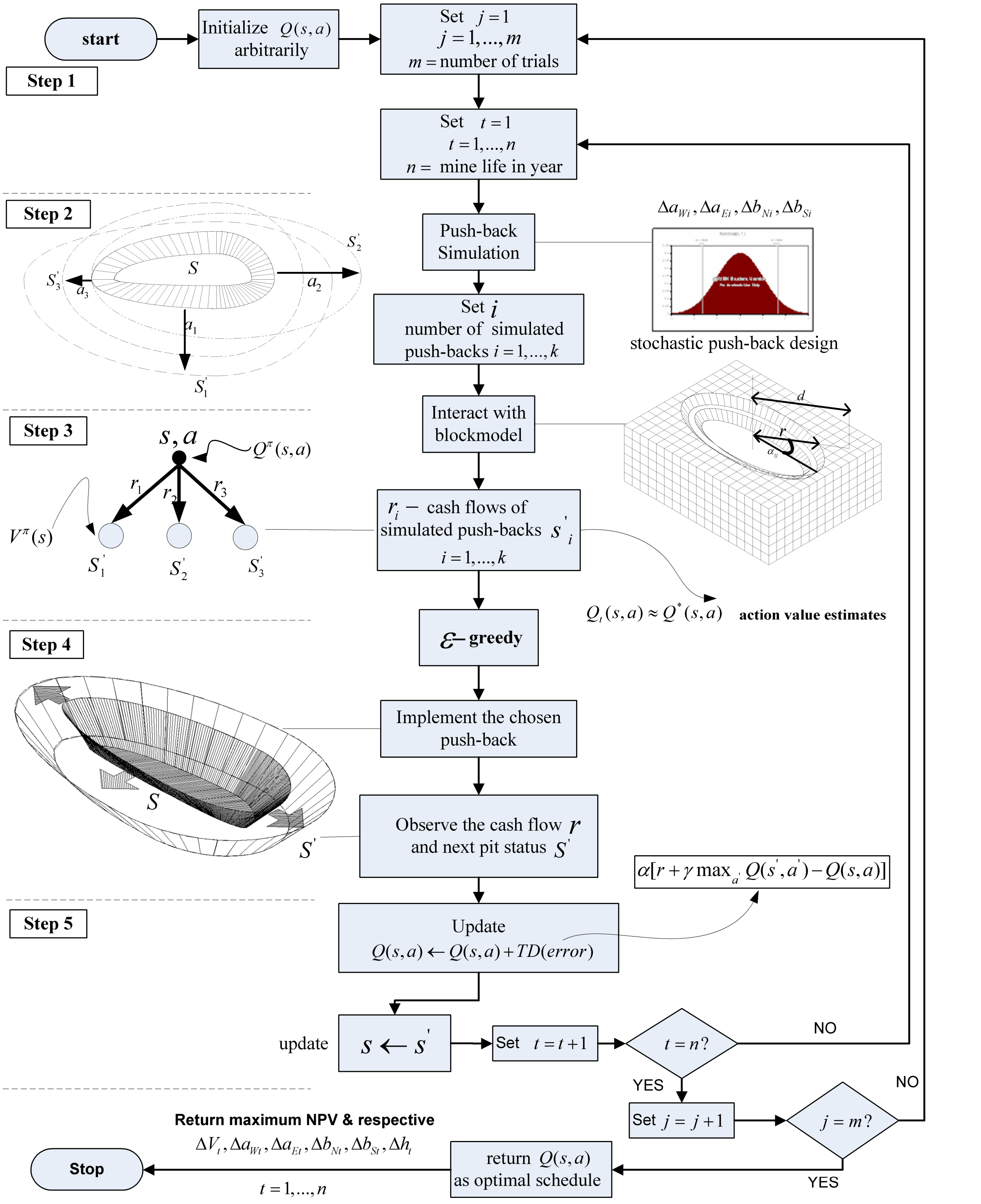
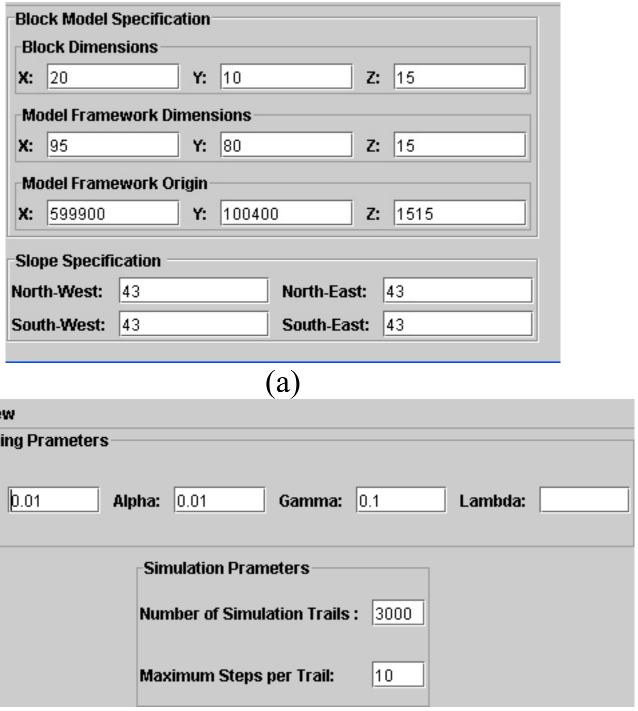

start
Step 1
Initialize $Q(s,a)$ arbitrarily
Set $j=1$
$j=1,\ldots,m$
$m=$ number of trials
Set $t=1$
$t=1,\ldots,n$
$n=$ mine life in year
Step 2
S
S_1'
S_2'
S_3'
a_1
a_2
a_3
Push-back Simulation
$\Delta a_{Wi}, \Delta a_{Ei}, \Delta b_{Ni}, \Delta b_{Si}$
stochastic push-back design
Set i
number of simulated push-backs $i=1,\ldots,k$
Step 3
s,a
$Q^{\pi}(s,a)$
r_1
r_2
r_3
$V^{\pi}(s)$
S_1'
S_2'
S_3'
Interact with blockmodel
d
r
α_{ij}
r_i − cash flows of simulated push-backs s_i'
$i=1,\ldots,k$
$Q_t(s,a)\approx Q^*(s,a)$
action value estimates
Step 4
ε– greedy
Implement the chosen push-back
S
S'
Observe the cash flow r and next pit status S'
$\alpha[r+\gamma \max_{a'} Q(s',a')-Q(s,a)]$
Step 5
Update
$Q(s,a)\leftarrow Q(s,a)+TD(error)$
update
$s\leftarrow s'$
Set $t=t+1$
$t=n?$
NO
YES
Set $j=j+1$
$j=m?$
NO
YES
Return maximum NPV & respective
$\Delta V_t, \Delta a_{Wt}, \Delta a_{Et}, \Delta b_{Nt}, \Delta b_{St}, \Delta h_t$
$t=1,\ldots,n$
Stop
return $Q(s,a)$ as optimal schedule

Figure 4 - Open pit Q-learning algorithm.

where: Q is the action-value function; α is a step-size parameter set to 0.01; S_t is the open pit geometrical state; a_t is a possible push-back at stage S; r_{t+1} is the cash flow of the simulated push-back; and γ is the discount factor. After updating the Q-values the algorithm moves to the next push-back and this process continues until it reaches the final pit limits. The algorithm will start the next episode of the push-back simulation by a random initial starting point in the pit. The number of iterations of simulation is controlled by the user. The algorithm is guaranteed to converge to the correct Q-values with the probability one under the assumption that the environment is stationary and depends on the current state and the action taken in it. Every state-action pair continues to be visited. Once these values have been learned, the optimal action from any state is the one with the highest Q-value.

4. Numerical Applications of the Intelligent Open Pit Simulator

A case study of an iron ore deposit is carried out to verify and validate the models. The extraction schedule from the Intelligent Open Pit Simulator is compared to the results of the Milawa algorithm and parametric analysis using Whittle® (Gemcom Software International, 1998-2006). The Intelligent Open Pit Simulator application was implemented in Java® (Sun Microsystems, 1994-2006) and MATLAB® (MathWorks, 2005) environment. This exercise consisted of class and object identification based on the Java Reinforcement Learning Library, JavaRL, (Kerr et al., 2003). The program requires the block model file as the input. The block model parameters are set through the block model specification tab illustrated in Figure 5(a). The Q-learning parameters and number of simulation iterations are set through the learning tab illustrated in Figure 5(b).



Block Model Specification

Block Dimensions

X: 20 Y: 10 Z: 15

Model Framework Dimensions

X: 95 Y: 80 Z: 15

Model Framework Origin

X: 599900 Y: 100400 Z: 1515

Slope Specification

North-West: 43 North-East: 43

South-West: 43 South-East: 43

(a)

Q-Learning Prameters

Epsilon: 0.01 Alpha: 0.01 Gamma: 0.1 Lambda:

Simulation Prameters

Number of Simulation Trails : 3000

Maximum Steps per Trail: 10

(b)

Figure 5 - (a) Block model specification (b) Q-learning parameters.

The iron ore deposit is explored with 159 exploration drill holes and 113 infill drill holes totalling 6,000 meters of drilling. Three types of ore, top magnetite; oxide; and bottom magnetite are classified in the deposit. Processing plant is based on magnetic separators so the main criterion to send material from mine to the concentrator is weight recovery. Kriging is used, to estimate the geological block model grades (Krige, 1951). The small blocks represent a volume of rock equal to 20 m×10 m×15 m. The model contains 114,000 blocks that makes a model framework with dimensions of 95×80×15. Figure 6 illustrates a multi cross-section of the deposit along sections 100100-east, 600245-north, and elevation of 1,590 m.

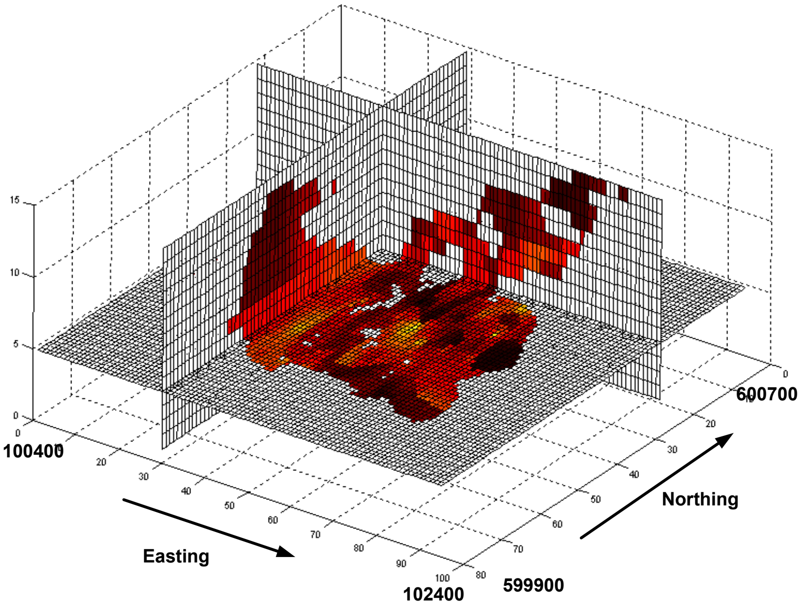


Figure 6- Three dimensional view of the deposit (coordinates in meters).

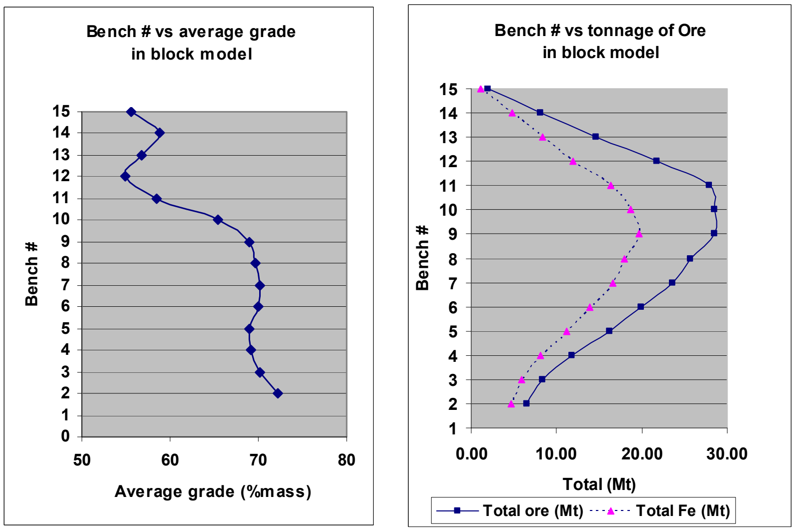


Figure 7 - Tonnage of ore and grade bench by bench.

The block model contains almost 243 million tonnes of indicated resource of iron ore with an average grade of 63%. Table 1 summarizes the block model information. Figure 7 shows the average grade, total amount of ore, and iron ore concentrate on a bench-by-bench basis.

The final pit limits are determined using the LG algorithm (Lerchs and Grossmann, 1965), using Whittle (Gemcom Software International, 1998-2006) software. Slope stability and geo-mechanical studies recommended a 43º overall slope in all regions. The average slope error in Whittle model is 0.9 degree and there are 35 possible structure arcs per block in the model which in total makes 3,075,666 arcs or edges in the graph model. The Pit Shells node in Whittle represents a set of pit shells generated by economic parametric analysis using the LG algorithm. This process reads in the block model from the Block Model node, pit slope constraints from Slope Set node, calculates block values using the economic and operational data contained in this node, and produces optimal pit outlines. The economic and mining parameters are based on: (i) mining cost = \$2/tonne; (ii) processing cost = \$2/tonne; (iii) selling price = \$15/tonne (Fe); (iv) maximum mining capacity = 20 Mt/year; (v) maximum milling capacity = 15 Mt/year; (vi) density of ore and waste = 4.2 tonne/m^3; and (vii) annual discount rate = 10%.

Table1- Summary of the ore and waste in the geological block model.

Rock Type	Blocks in model	Total (Mt)	Total Fe element (Mt)	Grade % Min	Grade % Avg	Grade % Max
Ore	19328	243.533	159.140	13	63.5	89
Waste	94672	1192.867	-	-	-	-

It is usual to produce multiple pit outlines in a single run and this process is controlled by the revenue factors in the optimization tab. The program finds a sequence of optimal push-backs based on varying the profitability of the deposit. In the generation of the pit shells, revenue factors in the range of 0.45 to 1.4 were used with variable geometric step sizes to scale base case price up and down, in order to control what nested pits are to be produced. It should also be considered that selection of a final pit has direct impact on the expected economic ore reserve. In terms of maximizing NPV, the lowest revenue factor that produces a pit sufficiently large to justify mining should also be the portion of the deposit to be mined first. Estimation of a project's NPV requires that timing of cash flow be accurately known so that an appropriate discount factor can be applied. This immediately introduces a problem for pit optimization software because the year of mining for any block of ore or waste will not be known until the mine production has been scheduled. The LG algorithm, which is the basis for Whittle software treats all mining activities as though it occurs simultaneously, with no discount factor applied. This usually results in selection of a final pit that is larger than the true maximum NPV pit.

Calculating the NPV requires knowing the relative time difference between blocks mined within a particular pit shell. This is dependent on the mill and mine capacities, practical sink rate (benches mined per year) and the equipment that can be practically operated within a specific cutback. Whittle provides a number of methods that work with the set of nested pits to provide a feasible production schedule. In this study the Milawa NPV algorithm was used. Milawa defines a variable bench interval between subsequent push-backs such that once a fixed number of benches have been mined out in the interior push-back then mining can commence on the next pushback. Thus, there is always a vertical lag of so many benches between push-backs. Milawa allows the lag to vary between push-backs and then searching for the combination of lags which is optimal either with respect to cash flow or managing stripping ratio.

The results of the Shells Node generated 77 nested pits with the respective total amount of ore, waste, and the NPV shown by Figure 8 for the best case, worst case, and Milawa algorithm. The appropriate push-backs are chosen in a way that the annual production targets are met in the long-term plan. The selected phases are represented by pits 17, 25, 43, 59, 65 and the final pit expected around pit 70. Successive schedules are run to different final pits from the first push-back to the pit shell number 77 in incremental steps of one. Pit shell number 68 with 209 million tonnes of ore and 182 million tonnes of waste has the highest NPV among all other pit shells and was chosen as the final pit limits for the production scheduling stage.

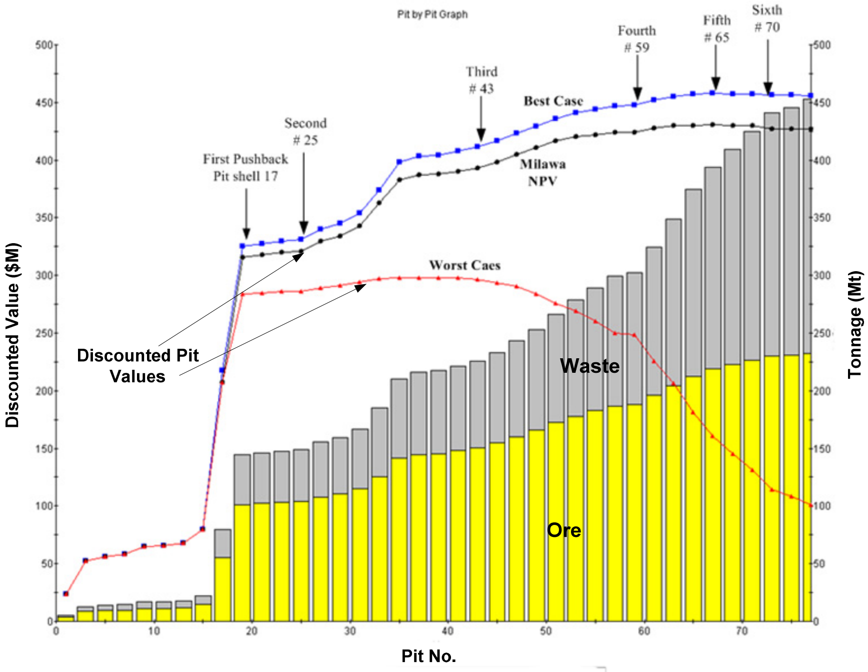


Figure 8 - Pit by pit graph.

The final pit outline in the previous section is the input for the comparison of Milawa NPV schedule and the Intelligent Agent algorithm. The comparative study is based on the following assumptions: (i) no stockpiles or materials re-handling was considered; (ii) blending of materials was not considered; (iii) the mill head grade and the annual mill feed was not set as a rigid constraint. The mill feed requirements are not the governing variables of the optimization in this case study; and (iv) all the planning parameters are kept the same in IOPS as the Whittle case study. The focus has been just on NPV maximization at this stage of the study. The final pit limits imported into IOPS are illustrated in Figure 9 with the respective dimensions of the major and minor axes of the frustum capturing the pit geometry. These dimensions are as follows: $a_W = 1,050\ m$; $a_E = 600\ m$; $b_N = 280\ m$; $b_S = 370\ m$; $h = 210\ m$.

The minimum mining width for the bottom of the pit was considered as an ellipse with major and minor axes of 60 m at any given time. The acceptable annual production targets were set to a maximum of 20 Mt; minimum of 19 Mt; and an average yearly production of 20Mt. IOPS simulates different mining starting points for each simulation episode based on a reference starting point coordinate provided by the user. Maximum three benches were allowed to be mined per year. The experiment was based on maximum mining capacity of 20 Mt/year and maximum milling capacity of 15 Mt/year. IOPS was used to run Q-learning algorithm with 3000 iterations with different scenarios of mining starting points. The probability that the agent "explores" as opposed to "exploiting" was set to $\varepsilon = 0.01$ in the epsilon-greedy algorithm. The learning rate for the intelligent agent, $\alpha = 0.01$; and the discount rate for delayed rewards, $\gamma = 0.1$.

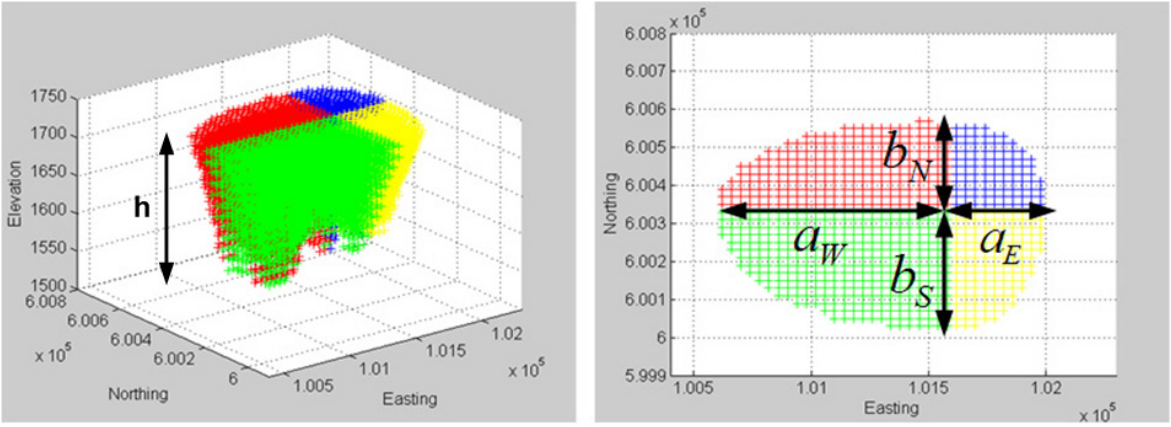


Figure 9 - Three-dimensional view and plan view of the final pit limits (meter).

5. Summary of results

The annual production schedule generated by IOPS compared to the results of Milawa NPV schedule are illustrated in Figure 10 and 11. From the analysis and comparisons of the results the following conclusions were drawn: (i) the optimized final pit limits show the total amount of 391 million tonnes of material consisting of 209 million tonnes of ore and 182 million tonnes of waste; (ii) Whittle 4-X yielded an NPV of $430 million over a 21-year of mine life at a discount rate of 10% per annum; (iii) IOPS yielded in an NPV of $438 million under the same circumstances and over the same mine life; (iv) The IOPS results proposed a starting point at 10160-east and 600340-north, which is located inside the smallest pit generated with nested pits in Whittle; (v) the fluctuations of annual production in both methods are caused by not setting the annual mill feed as the governing variable; (vi) IOPS shows a more consistent annual ore production compared to the Milawa NPV; and (vii) the Milawa NPV algorithm in Whittle 4-X is one of the standard tools widely used in industry.

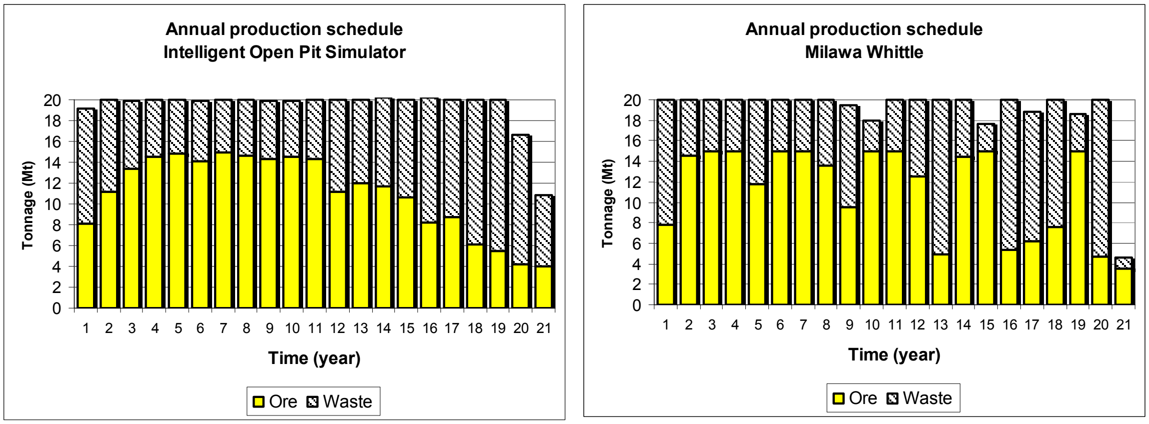


Figure 10- Comparative annual production schedule.

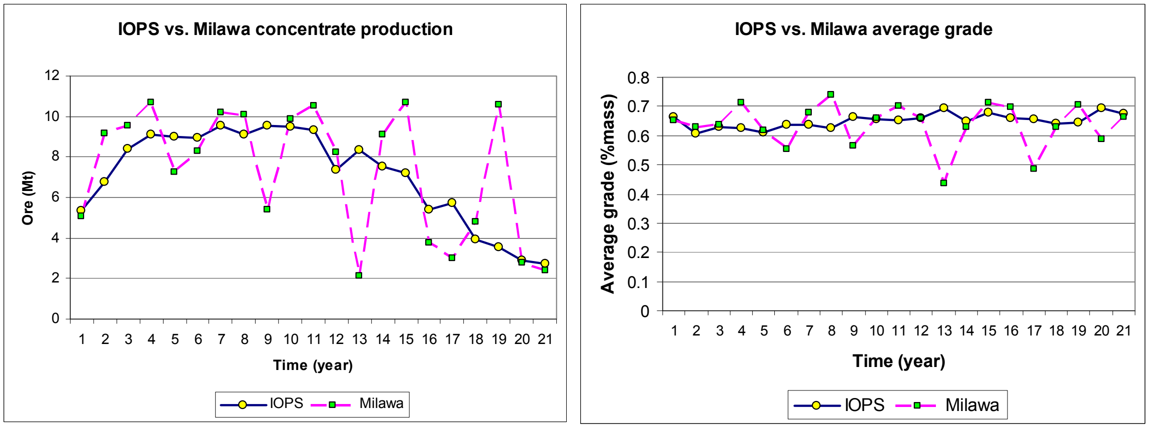


Figure 11- IOPS vs. Milawa results.

6. Conclusions

An intelligent agent theoretical framework for real size mine planning was developed based on reinforcement learning algorithms. The long term planning of the open pit mine is modelled as a dynamic decision network. The intelligent agent interacts with the open pit environment through simulation and employs Q-learning algorithm to maximize the net present value of the mining operation. An intelligent open pit production simulator, IOPS, is developed and implemented in Java® and MATLAB®. A stochastic simulation model captures the dynamics of open pit layout expansion. The developed algorithms are applied to a real-world mining operation. The numerical applications of the developed models are compared with the industry standard algorithms used in Whittle software.

The optimized final pit limits show the total amount of 391 million tonnes of material consisting of 209 million tonnes of ore and 182 million tonnes of waste. Whittle® software yielded an NPV of $430 million over a 21-year of mine life at a discount rate of 10% per annum. IOPS generated an NPV of $438 million under the same conditions. The focus of the case study at this stage has been on verifying and validating the models, which has been successful. The NPV from the IOPS schedule shows that the intelligent agent framework provides a powerful basis for addressing the real size open pit mine planning problem. Further focused research is required to develop and test the models based on intelligent agents to include more critical mine planning variables such as: variable optimized cut-off grades, constant annual mill feed, blending parameters, and stockpiles. Stochastic simulation as one of the major entities of the developed models has the strength to address the random field and dynamic processes involved in mine planning. The intelligent agent framework has the potential to be used for the optimal integration of mining and mineral processing systems, and development of a framework to quantify uncertainty relevant to mine planning and engineering design.

7. References

[1] Achireko, P. K. and Frimpong, S., (1996), "Open Pit Optimization using Neural Networks on Conditionally Simulated Blocks", *in Proceedings of 26th Applications of Computers and Operational Research in the Mineral Industry*, © SME, University Park, Pennsylvania, pp. 137-144.

[2] Askari-Nasab, H., (2006), "Intelligent 3D interactive open pit mine planning and optimization", PhD Thesis Thesis, © University of Alberta, Edmonton, Pages 167.

[3] Askari-Nasab, H., Awuah-Offei, K., and Frimpong, S., (2004), "Stochastic simulation of open pit pushbacks with a production simulator", *in Proceedings of CIM Mining Industry Conference and Exhibition*, © Edmonton, Alberta, Canada, pp. on CD-ROM.

[4] Askari-Nasab, H., Frimpong, S., and Awuah-Offei, K., (2005), "Intelligent optimal production scheduling estimator", *in Proceedings of 32nd Application of Computers and Operation Research in the Mineral Industry*, © Taylor & Francis Group, London, Tucson, Arizona, USA, pp. 279-285.

[5] Askari-Nasab, H., Frimpong, S., and Szymanksi, J., (2007), "Modeling Open Pit Dynamics using Discrete Simulation", *International Journal of Mining, Reclamation and Environment*, Vol. 21, 1, pp. 35-49.

[6] Askari-Nasab, H. and Szymanski , J., (2007), "Open Pit Production Scheduling Using Reinforcement Learning", *in Proceedings of 33rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 321-326.

[7] Caccetta, L. and Hill, S. P., (2003), "An application of branch and cut to open pit mine scheduling", *Journal of Global Optimization*, Vol. 27, November, pp. 349-365.

[8] Chanda, E. K. and Dagdelen, K., (1995), "Optimal blending of mine production using goal programming and interactive graphics system", *International Journal of Surface Mining Reclamation and Environment*, Vol. 9, pp. 203-208.

[9] Dagdelen, K., (1985), "Optimum multi-period open pit mine production scheduling by Lagrangian parameterization", Ph.D. Thesis Thesis, © Colorado School of Mines, Golden, CO,

[10] Dagdelen, K. and Kawahata, K., (2007), "Oppurtunities in Multi-Mine Planning through Large Scale Mixed Integer Linear Programming Optimization", *in Proceedings of 33rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 337-342.

[11] Denby, B. and Schofield, D., (1994), "Open-pit design and scheduling by use of genetic algorithms", *Transactions of the IMM Section A*, Vol. 103, January - April 1994,, pp. A21-A26.

[12] Denby, B., Schofield, D., and Hunter, G., (1996), "Genetic algorithms for open pit scheduling - extension into 3-dimensions", *in Proceedings of 5th International Symposium on Mine Planning and Equipment Selection*, © A.A.Balkema/Rotterdam/Brookfield, Sao Paulo, Brazil, pp. 177-186.

[13] Elveli, B., (1995), " Open pit mine design and extraction sequencing by use OR and AI concepts", *International Journal of Surface Mining. Reclamation and Environment*, Vol. 9, pp. 149-153.

[14] Erarslan, K. and Celebi, N., (2001), "A simulative model for optimum open pit design", *The Canadian Mining and Metallurgical Bulletin*, Vol. 94, October, pp. 59-68.

[15] Esfandiari, B., Aryanezhad, M. B., and Abrishamifar, S. A., (2004), "Open pit optimization including mineral dressing criteria using 0–1 non-linear goal programming", *Mining Technology, Transactions of the Institutions of Mining and Metallurgy*, Vol. 113, January, pp. A3-A13.

[16] Frimpong, S. and Achireko, P. K., (1997), "The MCS/MFNN Algorithm for Open Pit Optimization", *International Journal of Surface Mining, Reclamation & Environment*, Vol. 11, pp. 45-52.

[17] Gershon, M., (1983), "Mine scheduling optimization with mixed integer programming", *Mining Engineering*, Vol. 35, pp. 351-354.

[18] Halatchev, R. A., (2005), "A model of discounted profit variation of open pit production sequencing optimization", *in Proceedings of Application of Computers and Operations Research in the Mineral Industry*, © Taylor & Francis Group, London, Tucson, Arizona, pp. 315-323.

[19] International, G. S., (1998-2006), "Whittle strategic mine planning software", ver. 4.0: Gemcom Software International Inc.

[20] Kerr, A. J., Neller, T. W., Pilla, C. J. L., and Schompert, M. D., (2003), "Java Resources for Teaching Reinforcement Learning", *in Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '03)*, © Computer Science Research, Education, & Applications (CSREA) Press, Las Vegas, Nevada, pp. 1497-1501.

[21] Krige, D. G., (1951), "A statistical approach to some basic mine valuation and allied problems at the Witwatersrand",Thesis, © University of Witwatersrand, South Africa,

[22] Lerchs, H. and Grossmann, I. F., (1965), "Optimum design of open-pit mines", *The Canadian Mining and Metallurgical Bulletin, Transactions*, Vol. LXVIII, pp. 17-24.

[23] MathWorks, (2005), "MATLAB", ver. 7.04, MA, USA: MathWorks Inc.

[24] Onur, A. H. and Dowd, P. A., (1993), "Open pit optimization-part 2: production scheduling and inclusion of roadways", *Transactions of the Institution of Mining and Metallurgy*, Vol. 102, May-August, pp. A105-A113.

[25] Ramazan, S., Dagdelen, K., and Johnson, T. B., (2005), "Fundamental tree algorithm in optimising production scheduling for open pit mine design", *Mining Technology : IMM Transactions section A*, Vol. 114, 1, pp. 45-54.

[26] Ramazan, S. and Dimitrakopoulos, R., (2004), "Traditional and new MIP models for production scheduling with in-situ grade variability", *International Journal of Surface Mining, Reclamation & Environment*, Vol. 18, 2, pp. 85-98.

[27] Rubio, E., (2006), "Mill Feed Optimization for Multiple Processing Facilities using Integer Linear Programming", *in Proceedings of Proceeding of Fifteenth International Symposium of Mine Planning and Equipment Selection (MPES),*, © Torino, Italy, pp. 1207-1213.

[28] Sun Microsystems, I., (1994-2006), "Java Programming Language", ver. 1.4.2_08, 4150 Network Circle, Santa Clara, CA, USA

[29] Sutton, R. S. and Barto, A. G., (1998), "Reinforcement Learning, An Introduction", © The MIT Press, Cambridge, Massachusetts, Pages 432 pp.

[30] Tolwinski, B. and Underwood, R., (1992), "An algorithm to estimate the optimal evolution of an open pit mine", *in Proceedings of 23rd APCOM Symposium*, © SME, Littleton, Colorado, University of Arizona, pp. 399-409.

[31] Wageningen, A. V., Dunn, P. G., and Muldowney, D. M., (2005), "sequence optimization for long-term mine planning", *in Proceedings of 32 nd Application of Computers and Operation Research in the Mineral Industry*, © Taylor & Francis Group, London, Tucson, Arizona, USA, pp. 667-673.

[32] Watkins, C. J. C. H., (1989), "Learning from delayed rewards", PhD thesis Thesis, © University of Cambridge,

[33] Whittle, J., (1989), "The facts and fallacies of open-pit design," in *Manuscript, Whittle Programming Pty Ltd.* North Balwyn, Victoria, Australia

[34] Wooldridge, M., (2002), "An Introduction to Multi-Agent Systems", © John Wiley and Sons Limited, Chichester,UK, Pages 348.

[35] Youdi, Z., Qingziang, C., and Lixin, W., (1992), "Combined approach for surface mine short term planning optimization", *in Proceedings of 23rd APCOM Symposium*, © SME, Colorado, pp. 499-506.

[36] Yovanovic, A. P. and Araujo, A. C. d., (2007), "Operational Model and Computational Intelligence: A New Approach to Mineral Processing Optimization", *in Proceedings of 33 rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 491-498.

Open pit optimisation using discounted economic block values

Hooman Askari-Nasab and Kwame Awuah-Offei[1]
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

Strategic mine planning and the management of the future cash flows are a vital core of surface mining operations. The time dimension, which is an integral part of the scheduling problem, is not embedded in traditional ultimate pit outline optimisation algorithms. This study explores the validity of the theorem that a pit outline determined by an optimal long-term schedule algorithm is constrained by the conventional Lerchs and Grossmann's (LG) optimised pit outline. This hypothesis was investigated through a case study using the intelligent open pit simulator (IOPS) founded on agent-based learning theories. The optimal push-back schedule was determined using IOPS prior to determination of the optimised final pit outline. The economic block values were discounted with respect to the allocated extraction time, followed by final pit limits optimisation using LG algorithm.

1. Introduction

The mine planning process defines the ore body depletion strategy over time. The planning of an open pit mine considers the sequential nature of the exploitation to determine the order of block extraction in order to maximise the generated cash flow throughout the mine life. The optimal plan must determine the optimised ultimate pit limits and the mining schedule, however such an objective results in a computationally intractable problem. Therefore, the conventional optimum design of the final pit limits either based on graph theory (Lerchs and Grossmann, 1965; Zhao and Kim, 1992) or network flow algorithm (Johnson and Barnes, 1988; Yegulalp and Arias, 1992) aims at maximising the total profit rather than the total discounted cash flow. Traditionally, the process of open pit long-term scheduling includes first, finding the final pit limits which maximises the profit based on Lerchs and Grossmann's (1965) algorithm (LG). The ultimate pit limit design is usually followed by a life-of-mine (LOM) schedule. The goal of this schedule is to maximise the net present value of the operation within the predetermined LG ultimate pit limits constrained by mining, processing, geological, and technical limitations.

[1] Assistant Professor, Department of Mining & Nuclear Engineering, University of Missouri-Rolla, USA,

Whittle (1989) outlined the complexity of the time aspect of the exploitation as: (i) the pit outline with the highest value cannot be determined until the block values are known; (ii) the block values are not known until the mining sequence is determined; and (iii) the mining sequence cannot be determined unless a pit outline is available. To overcome this complex problem a number of research attempts have been made to optimise the production planning and pit limit problems, simultaneously.

Tolwinski and Underwood (1992) and Elveli (1995) used a method that combines dynamic programming, stochastic optimisation, and artificial intelligence with heuristic rules to obtain ultimate pit limit and production planning concurrently. Denby et al. (1996) employed genetic algorithm and simulated annealing by generation of random pit population and assessment of a fitness function to acquire the production and final pit concurrently. Erarslan and Celebi (2001) used a simulative optimisation approach to tackle the same problem.

Various models based on mixed integer programming mathematical optimisation have been used to solve the long-term open-pit scheduling problem (Caccetta and Hill, 2003; Ramazan and Dimitrakopoulos, 2004; Dagdelen and Kawahata, 2007). The mixed integer linear programming models theoretically have the capability to consider diverse mining constraints such as multiple ore processors and multiple material stockpiles and blending constraints. The applications of mixed integer programming models result in production schedules generating near theoretical optimal net present values. In practice, formulating a real size mine production planning problem by including all the blocks as integer variables will simply exceed the capacity of the current commercial mathematical optimisation solvers. Various methods of aggregation have been used to reduce the number of integer variables that are required to formulate the mine planning problem with mixed integer programming techniques. Ramazan and Dimitrakopoulos (2004) illustrated a method to reduce the number of binary integer variables by setting waste blocks as linear variables instead of integer variables. Ramazan et al. (2005) presented an aggregation method based on fundamental tree concepts to reduce the number of integer variables in the mixed integer programming formulation.

Caccetta and Hill (2003) have mathematically proven that obtaining the production schedule of a mine by first determining the final pit outline and then generating the schedule is sound. Their theorem considers an open pit mine in which all constraints have a non-negative upper bound and a zero lower bound. They have proven that any final contour generated by an optimal schedule would be a subset of the contour generated by the application of the LG algorithm. Dogbe and Frimpong (2004) employed a discounted economic block value to account for time in the pit optimisation process. They conducted a case study by dynamic programming LG algorithm for a two-dimensional pit. The authors concluded that discounting has no effect on the optimal pit contour although the reported pit value is overstated if the economic block values would be used instead of the discounted block values.

In a series of publications the authors developed intelligent agent-based theoretical framework for open pit mine planning (IOPS) (Askari-Nasab et al., 2005; Askari-Nasab, 2006; Askari-Nasab et al., 2007; Askari-Nasab and Szymanski, 2007; Askari-Nasab et al., 2008) comprising algorithms based on reinforcement learning (Sutton and Barto, 1998) and stochastic simulation models founded on modified elliptical frustum (Askari-Nasab et al., 2004; Askari-Nasab et al., 2007). IOPS has a component that simulates practical mining push-backs over the mine life. An intelligent agent interacts with the push-back simulator to generate an optimal push-back schedule using reinforcement learning. IOPS has been designed for long-term scheduling of large scale open pit mines. Traditional mine scheduling methods such as mixed integer linear programming focus on finding the sequence of extraction of individual blocks while meeting mining and processing constraints. Formulating a real size mine production planning problem by including all the blocks as decision variables will lead to a computationally intractable problem. To overcome the size problem, IOPS learns the optimal sequence of mine layouts expansions by simulating mining cuts instead of examining individual blocks. Simulation of mining cuts leads to reducing the number of variables formulating the problem. However, combining blocks into mining cuts will reduce the freedom of block variables analysed by the reinforcement learning algorithm. Simulating the mining cuts instead of individual blocks will affect the optimality of the IOPS solution. The notion of optimality in IOPS and throughout this paper is based on the assumption of using mining cuts as decision variables instead of individual blocks. However, considering the scale of the open pit mine planning problem and the numerous iterations that IOPS uses to simulate feasible push-backs, this approach will converge to a valid optimal push-back schedule. Given that IOPS handles push-backs, therefore it is not a proper tool for mid-range and short-range scheduling. IOPS could be coupled by mixed integer linear programming solvers to tackle short-range scheduling problems.

In this paper we will utilize the IOPS to scrutinize Caccetta and Hill (2003) theorem through a case study. The primary objective of this paper is to investigate whether the optimal long-term schedule is indeed constrained by the conventional LG ultimate pit layouts. IOPS is utilized to generate an optimal yearly push-back schedule without a predetermined final pit layout. IOPS assigns a time value to each block in the geological block model based on the optimal push-back schedule. Afterwards, the economic value of each block is discounted with respect to the extraction schedule (DEBV).

Next, the ultimate pit limit design using the LG algorithm with discounted economic block values (DEBV) and undiscounted economic block values (UEBV) are carried out, followed by comparison of results. The idea behind this approach is that, traditional pit optimisation algorithms assume that all blocks in the geological block model will be mined at the same time. In this sense, certain marginal ore blocks that may be classified as economic based on undiscounted economic block model by LG algorithm may in fact become uneconomical, given the excavation time difference between the waste blocks that precede the marginal ores.

Fig. 1 illustrates the stages of the study. In the following section IOPS theoretical framework and its implementation is reviewed. Subsequently, the ultimate pit limit design of an iron ore deposit with DEBV and UEBV are compared and results are presented and discussed. Finally, the conclusions are drawn and the potential and significance of the intelligent mine planning framework in assessing uncertainty in mine planning is discussed as future work.

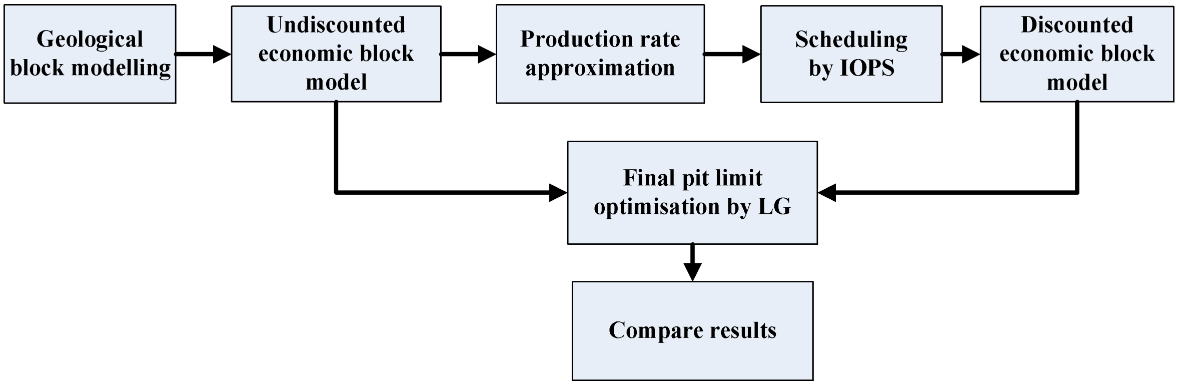

Geological block modelling
Undiscounted economic block model
Production rate approximation
Scheduling by IOPS
Discounted economic block model
Final pit limit optimisation by LG
Compare results

Figure 1 – Stages of the study.

2. Theoretical framework and models

2.1. Intelligent open pit planning framework

The theoretical framework, model development, and implementation of the intelligent agent-based open pit mine planning simulator (IOPS) are documented in detail in Askari-Nasab, 2006 (2006). Implementation of IOPS models were in MATLAB® and Java® programming language. Following is a review of the theoretical framework developed in previous studies. In the present study IOPS is used to schedule the push-back design of an iron ore deposit to assess the effect of discounted economic block values on the optimal ultimate pit limits.

The intelligent planning framework comprise independent, interactive and interrelated subsystems with processes, using reinforcement learning as the main engine to maximise the net present value of mining operations. The main integral parts of the theoretical framework are as follows: (i) environment: consists of geological block model and economic block model; (ii) simulation: open pit production simulator that captures the discrete dynamics of open pit layout expansion, and materials transfer with the respective annual cash flows. The simulation model consists of a number of interrelated subsystems. The development and performance of the simulation components are discussed in details in previous papers (Askari-Nasab, 2006; Askari-Nasab et al., 2007); (iii) agent: The simulated results are transferred to the intelligent open pit agent where a Q-learning algorithm (Sutton and Barto, 1998) serves as the engine. The production simulator passes the respective amount of ore, waste, and the cash flows of the production periods to the agent. Development of the intelligent agent mine planning architecture is based on mathematically idealised forms of reinforcement learning problem. The main concepts of optimality and the models in this study are developed and adapted from Sutton & Barto (1998) and Wooldridge (2002) . Fig. 2 illustrates the mine planning intelligent agent architecture.

The pit geometry evolution is viewed as series of snapshots over time. The agent and the simulation interact at each sequence of discrete time steps, $t = 1,...,n$. The simulation of the mining operation starts with the initial box cut at state, $s_{t-1} \in S_t$, and the agent responds by choosing the next pushback, $a_{t-1} \in A_t$, to be performed in this stage. As a result of this action, the simulation and environment can respond with a number of possible states.

However, only one state will actually result. On the basis of this second state of the environment, the agent again chooses an action to perform. The environment responds with one of a set of possible actions available, the agent then chooses another action, and so on. More specifically, the learning agent and simulation interact at each of a sequence of discrete time steps. At each time step t, the agent receives some representation of the open pit's state, $s_t \in S$, where S is the set of possible push backs. On the basis of S, the agent selects an action, $a_t \in A(s_t)$, where $A(s_t)$ is the set of changes possible in the pit geometry in state s_t. One time step later, in part as a consequence of its action, and interaction with the block model the agent receives a numerical reward, which is the cash flow of that period of mining operation, $r_{t+1} \in R$. As the result the agent finds itself in a new state, s_{t+1}. At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. This mapping is called the agent's policy and is denoted by, π_t, where $\pi_t(s,a)$ is the probability that $a_t = a$ if $s_t = s$.

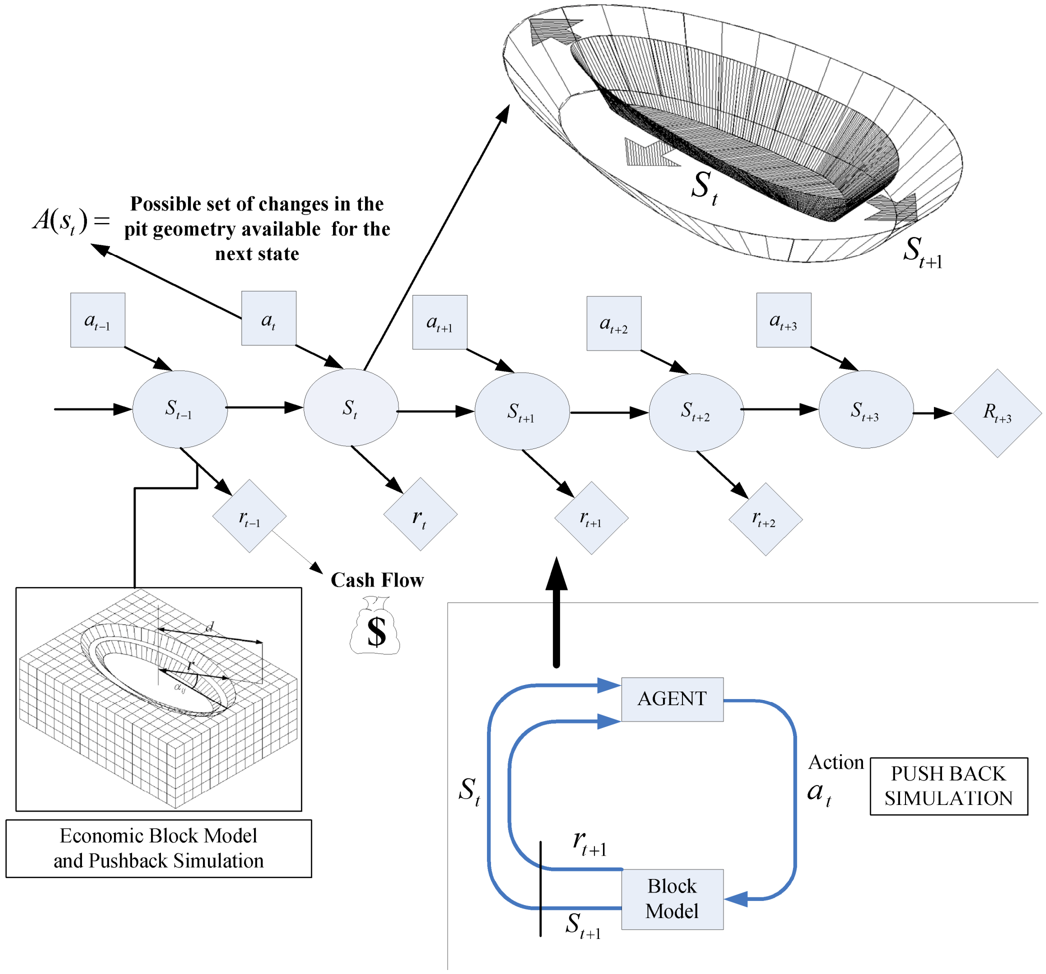


Figure 2- Intelligent mine planning agent model as a reinforcement learning problem.

Reinforcement learning methods specify how the agent changes its policy as a result of its experience. The agent's goal is to maximise the total amount of reward it receives over the long run. The objective is to maximise the *expected return*, where the return (see Fig. 2), R_t given by equation (1), is defined as a specific function of the immediate reward sequence. In equation (1), γ is the discount factor and is a number between 0 and 1; r_t is the numerical reward, which is the cash flow of the simulated push-back in period t; The discount factor describes the preferences of an agent for current rewards over future rewards. When γ is close to 0, rewards in the distant future are viewed as insignificant. i in equation (2) is the discount rate for time slice, t.

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... + \gamma^k r_{t+k+1} = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \tag{1}$$

Where: $\gamma = \dfrac{1}{1+i}$ (2)

Almost all reinforcement learning algorithms are based on estimating *value functions*--functions of states that estimate how good it is for the agent to be in a given state or how good it is to perform a given action in a given state. The notion of "how good" here is defined in terms of expected return. Accordingly, value functions are defined with respect to particular policies. Following one of the push-back designs the open pit will expand to the status of $s_1^{'}, s_2^{'}$, or $s_3^{'}$. The *value* of state s under policy π, denoted by $V^{\pi}(s)$, is the expected return or the NPV, when starting in s and following the policy thereafter, until reaching the final pit limits. For the Markov Decision Process representing the open pit dynamics in Fig. 2, $V^{\pi}(s)$ can be defined as equation (3).

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\{\textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\} \tag{3}$$

$E_{\pi}\{\ \}$ denotes the expected NPV given that the agent follows policy π, and t is any time step. The policy π is the current production schedule. The function V^{π} is called the *state-value function for policy* π. Similarly, the value of taking action a in state s under a policy π, denoted $Q^{\pi}(s,a)$ is defined as the expected NPV of the operation starting from s, taking the action a, and thereafter following the current schedule (policy π). Theoretically the interaction between the agent and the environment is a non-terminating process. In practice, the infinity sign in equation (3) represents a large number of simulation iterations. Q^{π} is called the *action-value function for policy* π given by equation (4).

$$Q^{\pi}(s,a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\} = E_{\pi}\{\textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\} \tag{4}$$

The Q-learning algorithm is used in this study to directly approximate the optimal action-value function denoted by equation (4), which is the optimal mine pushback design.

2.2. Algorithm development

Fig. 3 illustrates the detailed flow chart of the intelligent optimal mine planning algorithm based on Q-learning algorithm. The steps of the algorithm are as follows:

Step 1

The algorithm starts with (i) arbitrarily initializing the $Q(s,a)$, which is the expected discounted sum of future monetary returns of expanding the open pit from status S to the S' by choosing the push-back a and following an optimal policy thereafter; (ii) set the number of simulation trials that the algorithm is run. In other words the number of times that the open pit dynamics are being simulated from the initial box cut to the extent that all the ore in the model is depleted.

Step 2

The push-back simulator captures the open pit layout evolution as a result of the material movement. At this stage the algorithm stochastically simulates a number of practical push-back designs for the next production period. The result of the simulation is k push-backs $a_1, a_2, ..., a_k$ that satisfy the tonnage production of the next period. Following each of these push-backs $a_1, a_2, ..., a_k$, the open pit will expand to the status of $s_1', s_2', ..., s_k'$. The value of state s under policy π, denoted $V^{\pi}(s)$ is the expected return or the NPV of the sequence, when starting in s and following the policy thereafter until reaching the boundary that all the ore in the model is depleted.

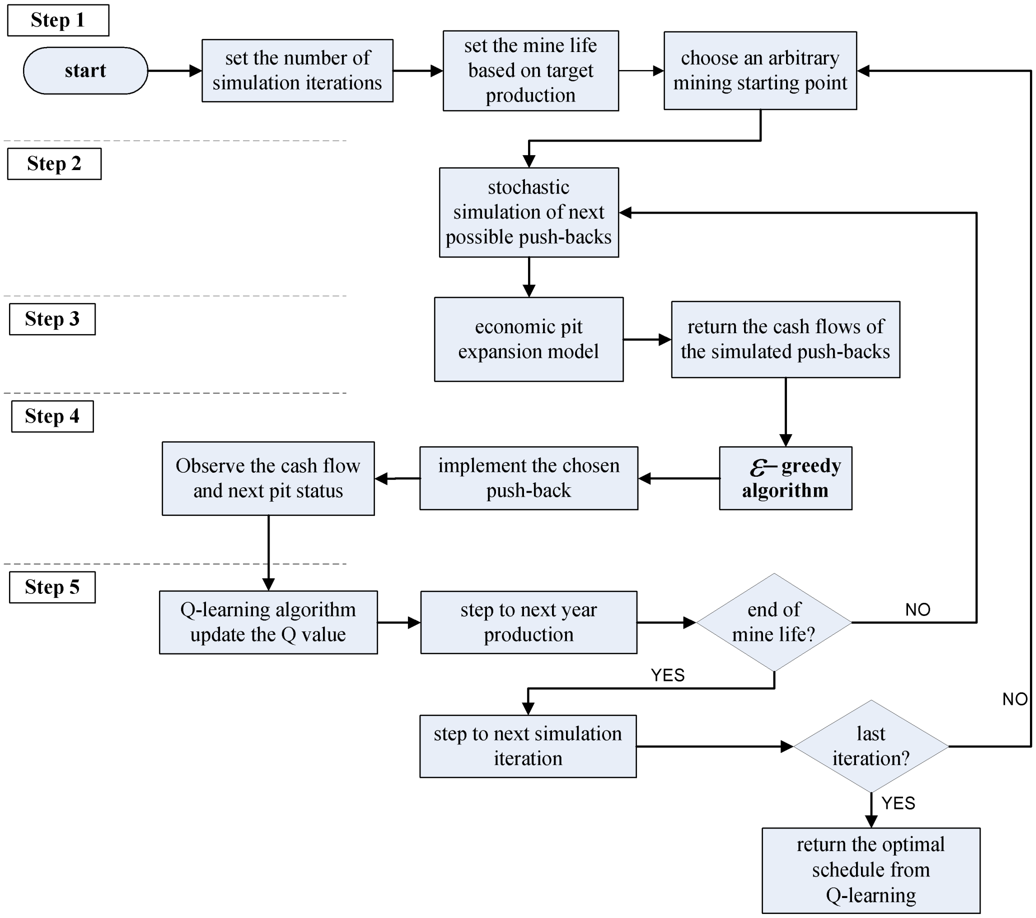

Step 1
start
set the number of simulation iterations
set the mine life based on target production
choose an arbitrary mining starting point
Step 2
stochastic simulation of next possible push-backs
Step 3
economic pit expansion model
return the cash flows of the simulated push-backs
Step 4
ε-greedy algorithm
implement the chosen push-back
Observe the cash flow and next pit status
Step 5
Q-learning algorithm update the Q value
step to next year production
end of mine life?
NO
YES
step to next simulation iteration
last iteration?
NO
YES
return the optimal schedule from Q-learning

Figure 3- Open pit Q-learning algorithm.

Step 3

Simulated push-backs $a_1, a_2, ..., a_k$ are fitted on the economic block model, where the cash-flows $r_1, r_2, ..., r_k$ of each push-back is returned to the program.

Step 4

The epsilon greedy algorithm is called (Sutton and Barto, 1998). The action selection rule is to select the action or one of the actions with highest estimated action value, that is, to select the push-back at time step t with the highest cash flow. The algorithm behaves greedily most of the time, which means it will select a push-back with the highest cash-flow among $r_1, r_2, ..., r_k$. But every once in a while, say with small probability ε, instead the algorithm selects an action at random, independently of the action-value estimates of the push-back. Subsequently the chosen push-back is implemented and the agent finds itself in pit status S' and observes the cash flow r.

Step 5

After being initialized to arbitrary numbers in step 1, Q-values are estimated on the basis of the experience. At this stage the algorithm updates $Q(s,a)$ based upon the previous experience as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \qquad (5)$$

where Q is the action-value function; α is a step-size parameter; S_t is the open pit geometrical state; a_t is the possible push-backs at stage S; r_{t+1} is the cash flow of the simulated push-back; γ is the discount factor. After updating the Q-values the algorithm moves to the next push-back and this process continues until it reaches the final pit limits. The algorithm will start the next iteration of the push-back simulation by a random initial starting point in the pit. The number of iterations of simulation is controlled by the user. The algorithm is guaranteed to converge to the correct Q-values with the probability one under the assumption that the environment is stationary and depends on the current state and the action taken in it. Every state-action pair continues to be visited. Once these values have been learned, the optimal action from any state is the one with the highest Q-value.

Economic block modelling: The profit from mining a block depends on the value of the block and the costs incurred in mining and processing. The cost of mining a block is a function of its spatial location, which characterises how deep the block is located relative to the surface and how far it is relative to its final dump. The spatial factor can be applied as a mining cost adjustment factor for each block according to its location to the surface. The economic block value of each block is given by equation (6).

$$EBV = (T_O \times g \times re \times P - T_O \times PC) - T \times MC \qquad (6)$$

Where T_O = amount of ore in the block (tonne), g = grade (%), re = recovery, the proportion of product recovered by processing the ore (%), P = the price obtainable per unit of product sold ($\$ \times tonne^{-1}$), MC = the cost of mining a tonne of waste ($\$ \times tonne^{-1}$), PC = cost per tonne of mining the material as ore and processing it ($\$ \times tonne^{-1}$), T = total amount of ore and waste in the block $(T_O + T_W)$ (tonne), and T_w = amount of waste in the block (tonne).

The objective function of final pit outline optimisation is to maximise the pit value given by equation (7) subject to safe slope constraints in all regions of the pit. All the current ultimate pit optimisation algorithms maximise equation (7). In other words the algorithms don't take into account the time aspect of extraction.

$$Uval_{pit} = \sum_{x=1}^{nr}\sum_{y=1}^{nc}\sum_{z=1}^{nl} EBV_{xyz} \tag{7}$$

where $Uval_{pit}$ = undiscounted pit value, x = block index in rows (northing), y = block index in columns (easting), z = block index in levels (elevation), nr = number of blocks in rows (x direction), nc = number of blocks in columns (y direction), and nl = number of blocks in levels (z direction). If the sequence of block extraction would have been known, then the pit optimisation objective function would have been to maximise the value of equation (8), which includes the temporal nature of mining operation. The economic value of each block is discounted with respect to its extraction time, t.

$$Dval_{pit} = \sum_{x=1}^{nr}\sum_{y=1}^{nc}\sum_{z=1}^{nl} \frac{EBV_{xyz}}{(1+i)^t} \tag{8}$$

where $Dval_{pit}$ = discounted pit value, i = discount rate, and t = time period in which the block is removed. IOPS was used to schedule the sequence of extraction. Each block was allocated a time attribute representing the extraction period. The economic value of each block was discounted with respect to the time attribute allocated by the IOPS schedule.

3. Case study: an iron ore pit outline optimisation using discounted block values

A case study of an iron ore deposit was carried out to assess Caccetta and Hill (2003) theorem, which states that obtaining the production schedule of a mine by first determining the final pit outline and then generating the schedule is sound. In this case study we used the IOPS to generate a yearly push-back schedule based on the input geological block model with no pit limits defined. The optimal schedule of the push-backs yielded the period of extraction of each block. The economic block values were discounted with respect to the scheduled periods. Afterwards, LG algorithm was used to find the ultimate pit limits for the discounted and undiscounted blocks models and then the results were compared.

Stage 1- geological block modelling: The deposit comprises two anomalous zones that are joined in depth. The southern part has a greater cover of rock and overburden than the northern side. The average cover of overburden and rock in the deposit is 156 meters. The southern part shows greater magnetic intensity. The iron ore deposit is set in a metamorphic complex of probable Paleozoic age which trends in a west-northwesterly direction. The metamorphic belt occurs as a series of discontinuous strike-parallel belts of rocks separated and interrupted by major block faults and regional thrust faults.

The general shape of the deposit is a tabular form elongated in a north-south direction. Fig. 4 illustrates collar of 139 exploration drillholes which were used in this study with the total length of 44,896 meters and the outline of the orebody. The overall exploration area is about 2,200m north-south by about 1,700m east-west. Maximum vertical thickness of ore is 139m on the east side of the deposit and the minimum vertical thickness is 1.5 m on the southwest side of the deposit. All, but one, drillholes are drilled vertically. Drillhole collars are surveyed and there were no in-hole surveys. Assay data have been taken for total iron (percent Fe), iron oxide (percent FeO), phosphorus (percent P), and sulphur (percent S) on the crude ore feed. Processing plant is based on magnetic separators, therefore the main criterion in selecting ore to be sent to the concentrator is the magnetic weight recovery (percent MWT) of iron ore measured by Davis Tube tests (Murariu and Svoboda, 2003). In this study MWT% was the basis for all the calculations, grade tonnage curves, average grades reported. Resources and reserves are also classified based on MWT%.

Drillhole compositing was carried out by first, grouping consecutive drillhole records of similar rock types together, and then the grades' composites were calculated.

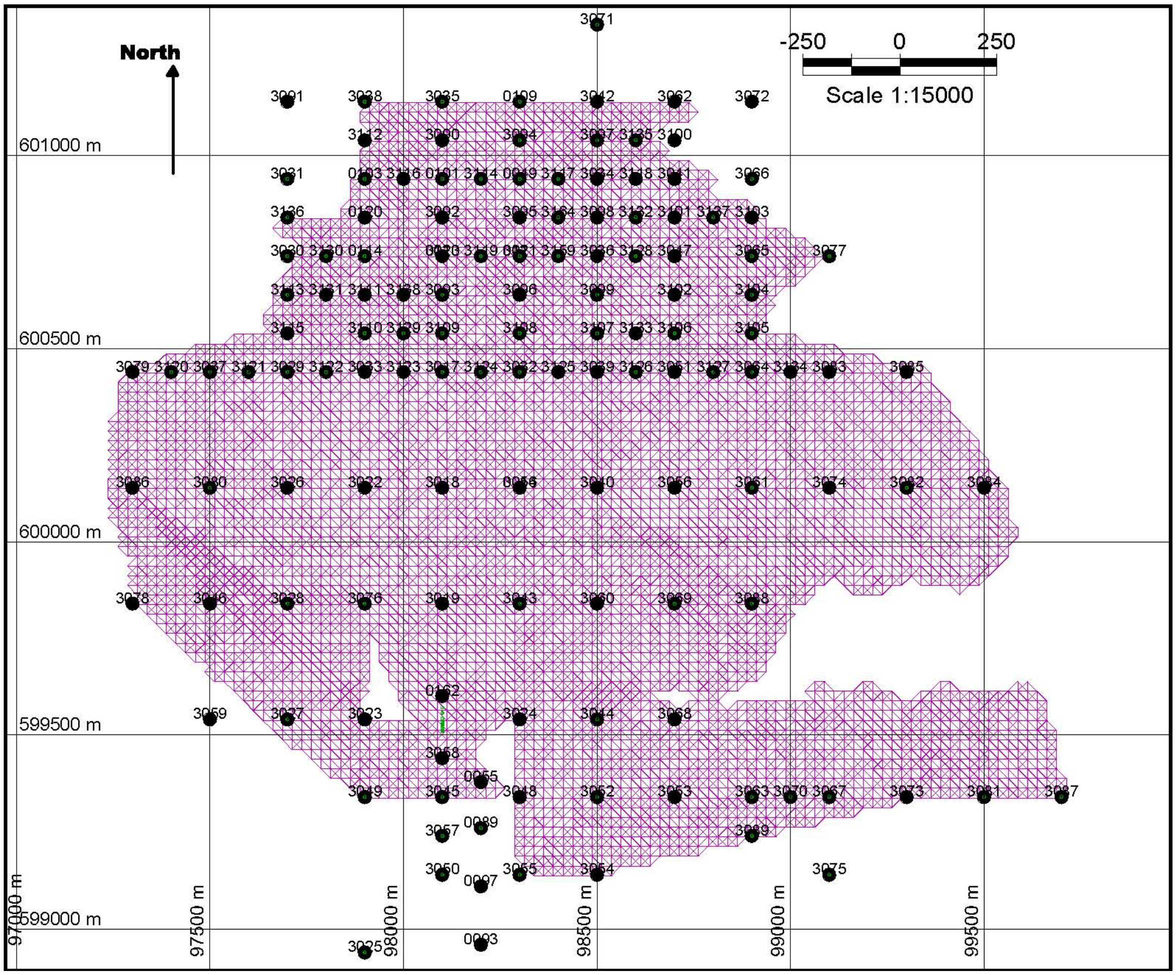


Figure 4- Plan view of the deposit and the location of drillholes.

Three types of ore (top magnetite, oxide, and bottom magnetite) are classified in the deposit. Table 1 summarises the different rock types with their respective densities. Kriging was used, to estimate the percentage of Fe, FeO, P, S, and MWT in constructing the geological block model (Deutsch, 2002). The small blocks represent a volume of rock equal to 25m×25m×15m. The model contains 883,200 blocks that makes a model framework with dimensions of 160X×120Y×46Z blocks.

Table 1 – Rock types modelled in the study.

Waste		Ore	
Rock Type	Density $(tonne \times m^{-3})$	Rock Type	Density $(tonne \times m^{-3})$
Alluvial overburden	1.85	Top Magnetite	4.13
Gneiss	2.65	Oxide	4.13
Quartz Schist	2.65	Bottom Magnetite	4.16
Undefined waste	2.65		

Stage 2 - economic block modelling and sequencing by IOPS: The initial estimate of the block model indicates that the resource contains in excess of 620 million tonnes of magnetic iron ore. Further detailed studies are required to classify the resource into inferred, indicated, and measured categories. Fig. 5 illustrates the cutoff grade – tonnage curve. The average grade above cut-off varies between 73% and 81% magnetic iron ore (MWT). Since the processing plant is based on magnetic separators, the efficiency of the plant has a crucial effect on the recoverable amount of iron ore and consequently on the cutoff grade.

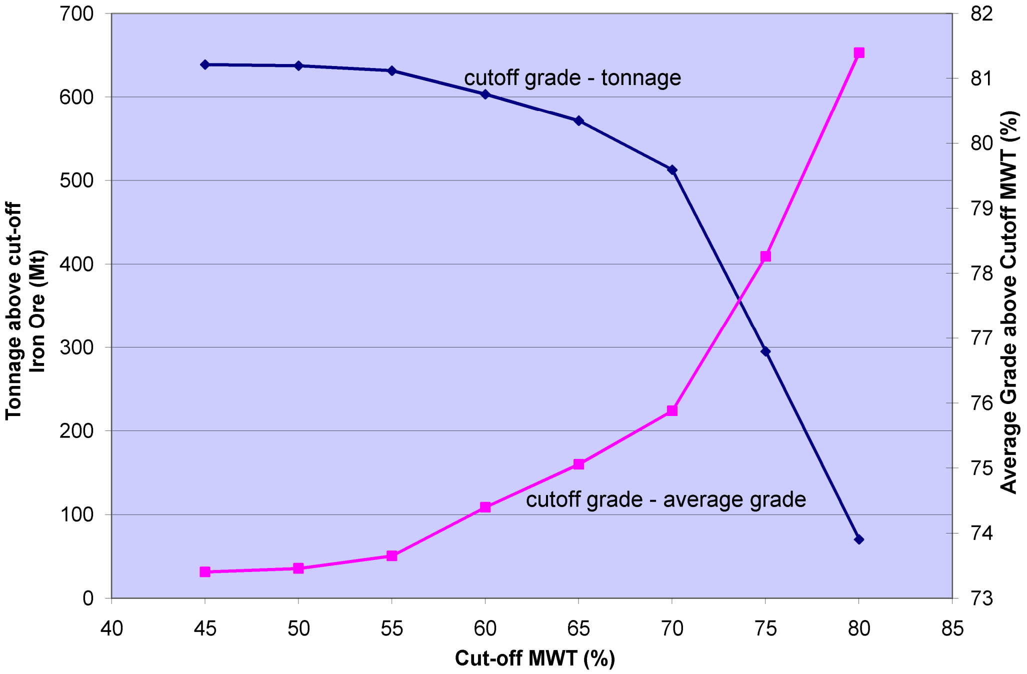


Figure 5- Cutoff grade-tonnage curve.

The intelligent open pit simulator, IOPS, (Askari-Nasab and Szymanski, 2007) was used to schedule the extraction sequence and to allocate extraction time tags to each block. IOPS was run for 3000 iterations with different scenarios of mining starting points with the following economic and technical parameters: (i) mining cost = \$2.02 $tonne^{-1}$; (ii) processing cost = \$4.65 $tonne^{-1}$; (iii) selling price = \$112 $tonne^{-1}$(Fe); (iv) maximum mining capacity = 150 $million\, tonne \times year^{-1}$; (v) mining recovery = 95%; (vi) processing recovery = 90%; and (vii) annual discount rate = 10%. Slope stability and geo-mechanical studies recommended the following overall slopes in different areas of the future open pit: northern wall 41 degrees; eastern wall 42 degrees; southern wall 39 degrees; and the western wall 46 degrees.

IOPS was used to simulate the optimal push-back schedule without a predefined final pit outline. The yearly push-back schedule was generated over sixty three years, with a mining rate of 150 $million\, tonne \times year^{-1}$. Since no final limits were defined the sixty three years is representing the time that is required to extract all the blocks within the block model enforcing feasible slope constraints. The optimal push-back schedule, with the maximised total discounted cash flow, was used to allocate an extraction time label to each block. Subsequently, the economic values of blocks were discounted with respect to the extraction time of each block within the schedule.

Stage 3 – optimal final pit limit design: The final pit limit design was carried out based on the LG algorithm using the Whittle software (1998-2007). The same economic and technical parameters applied in IOPS were used at this stage. IOPS discounted block model file contains the discounted economic value and an extraction time tag. To compare the models under the exact same circumstances block models were exported to the Whittle software as value models rather than grade block models. Value block models contain the economic value of each block in addition to the geological and grade information. The optimisation is carried out based on the pre-calculated block values. Two block models were the input into the LG algorithm: the original undiscounted economic block model, and the discounted economic block model generated by IOPS. Bench Phases technique, which incorporates the use of successively deeper benches was used to optimise the ultimate pit outline. In this technique, initially, only the top bench is considered for optimisation. A pit optimisation is performed and if there are any blocks worth mining they would constitute the next pit shell. Next, a bench is added and a pit optimisation is performed again and the results represent the next pit shell. The process of adding benches and optimising is repeated until a series of nested pits reach the final optimised pit limits.

4. Results and discussion

Fig. 6 illustrates the cumulative undiscounted cash flow of fifteen sample push-back simulation iterations out of 3000 iterations over 63 years. A range of feasible mining starting points and different push-back scenarios were simulated by IOPS. The learning agent interacts with the push-back simulation runs to generate the optimal push-back schedule. Given that the final pit limits is not defined at this stage, the simulation generates push-backs with increase in the cumulative cash flows until it reaches a climax where the cost of mining exceeds the generated revenue. Fig. 6 also demonstrates how the cumulative cash flows start declining when the pushback simulation passes the possible final pit limits.

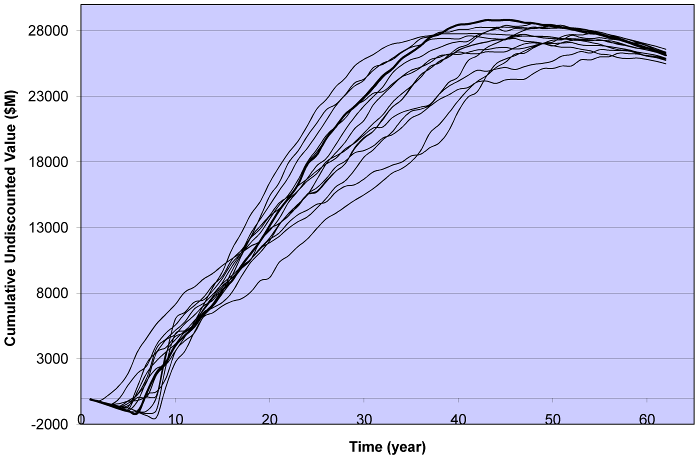


Figure 6- Cumulative undiscounted cash flows of fifteen simulation runs.

Fig. 7 shows the effect of a 10% discount rate on monetary values of fifteen sample simulation runs over time. For instance, the present value of a block scheduled to be extracted in year twenty four is almost 10% of the value of that block if it was extracted today. Fig. 7 also demonstrates how discounting could possibly affect the economic block values if the extraction time were known. Furthermore, it accentuates the importance of discounted pit value optimisation rather than the traditional objective function of profit optimisation in the optimal mine layout design.

The simulation results confirm that discounting decreases the value of the predicted cash flows after year fifteen. In fact, the objective function of optimising the discounted cash flows in mine planning places more weight on the dynamics of the operation in the first fifteen years rather than the mine life. Care must be taken in optimising the objective function of life-of-mine schedule based on the discounted cash flows. Additional important mining constraints such as: mine life, maintaining mill feed, and maintaining mill head grade must be included in the models. Fig. 8 illustrates the same concept with plotting the cumulative discounted cash flows over 63 years. The cumulative discounted cash flow curves reach a maximum value around year thirty eight and then flatten out for the rest of the mine life. The negative discounted cash flows at the early years show the substantial amount of overburden that needs to be removed. It is required to plan pre-stripping before getting into production to make sure that sufficient space and ore will be available in the later years to feed the processing plant.

Fig. 9 and Fig. 10 illustrate the total tonnage of ore and total tonnage of material in each nested pit for DEBV and UEBV models. The optimal final pit limits for the UEBV contains 635 Mt of iron ore, whereas the DEBV model shows 629 Mt of magnetic iron ore. Both models have an average grade of 73% magnetic weight recovery (MWT%). Although discounting the block values within IOPS was carried out over a long period of 63 years, the pit outlines for the two models did not show a significant difference in shape and the amount of ore and waste. This could be because of the shape of the iron ore deposit studied. Also, the tonnage of material that could be classifies as ore or waste near the pit limits is not that significant comparing to the total tonnage of ore in the resource. On the other hand, discounting would affect both costs and revenues at the same time. As a result, only the blocks that their economic block values are close to zero after discounting are not considered within the pit limits for the discounted model. Fig. 11 demonstrates the anticipated value difference between the undiscounted and discounted pit values.

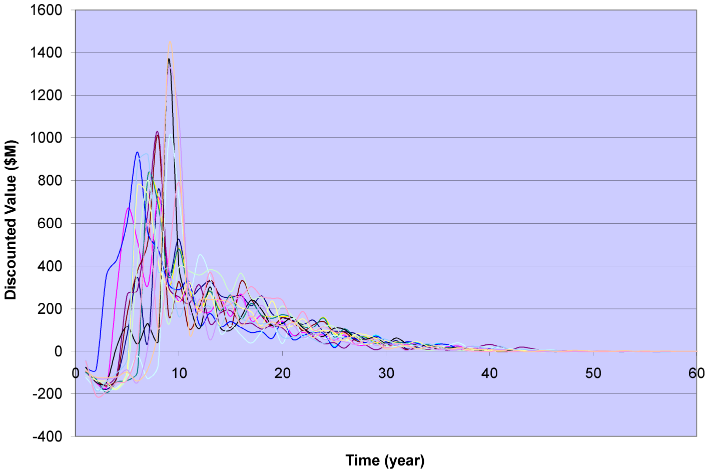


Figure 7- Discounted annual cash flows of fifteen simulation runs.

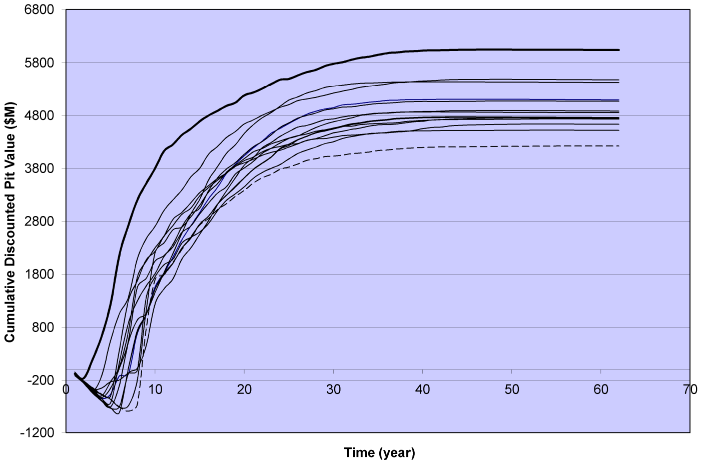


Figure 8- Cumulative discounted cash flows of fifteen simulation runs.

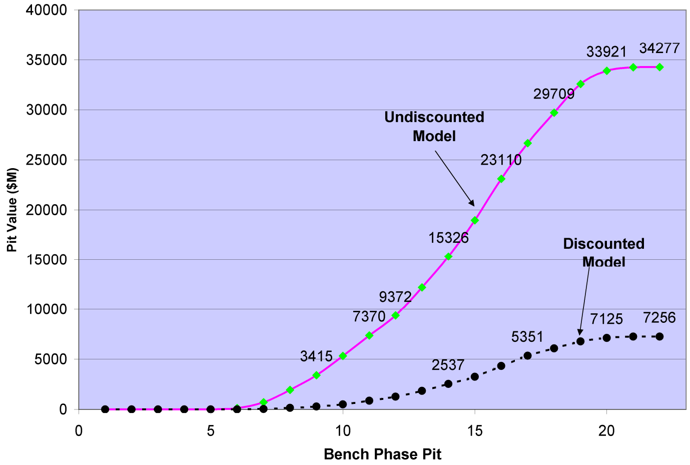


Figure 9- Tonnage of ore in the discounted and undiscounted nested pits.

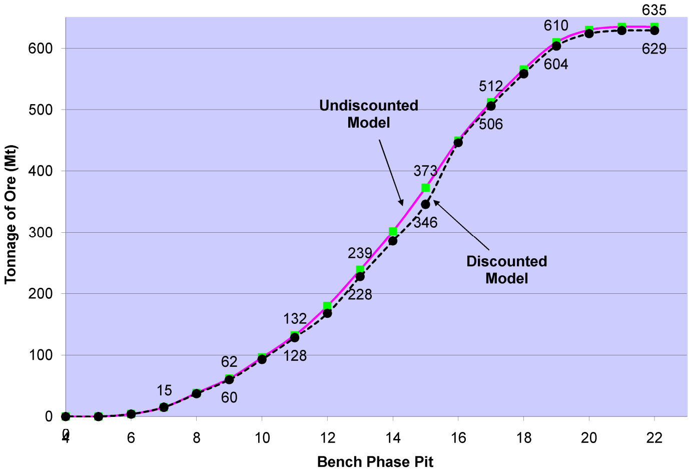


Figure10- Total tonnage of material in the discounted and undiscounted nested pits.

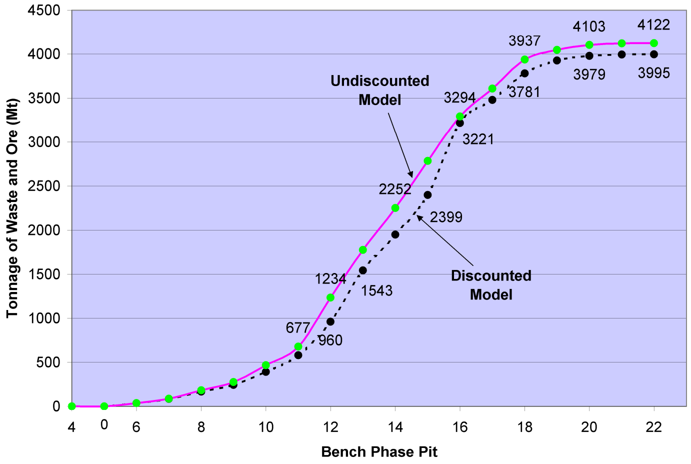


Figure 11- Undiscounted and discounted pit values.

The investigation of the shape of the final pit limits on cross sections in various directions revealed an overlap of two pits in most of the areas. Fig. 12 to 14 illustrate cross sections of the deposit with the final pit outlines based on the UEBV and DEBV models looking west. The legend is representing the iron ore grade based on MWT%. The two final pits lie on top of each other on the northern part of the deposit. On the southern part of the deposit the two pits separate and have more difference in shape as they progress in the eastern direction. Fig. 15 to 17 demonstrate the deposit and the pit outlines looking north. The UEBV and DEBV pits completely overlap on the eastern side and there is a small difference on the western wall.

The differences between the two outlines are due to the fact that the decrease in economic block values because of discounting in certain marginal areas is more than the effect of discounting on the cost of removing some of the overlying waste blocks, which are not within the same extraction period. In fact, the revenue generated by the marginal ore blocks are discounted with a higher rate comparing to the waste blocks positioned at an earlier extraction period. As a result in the LG algorithm the revenue generated by the marginal ore blocks can not cover the costs of removing the waste covering them.

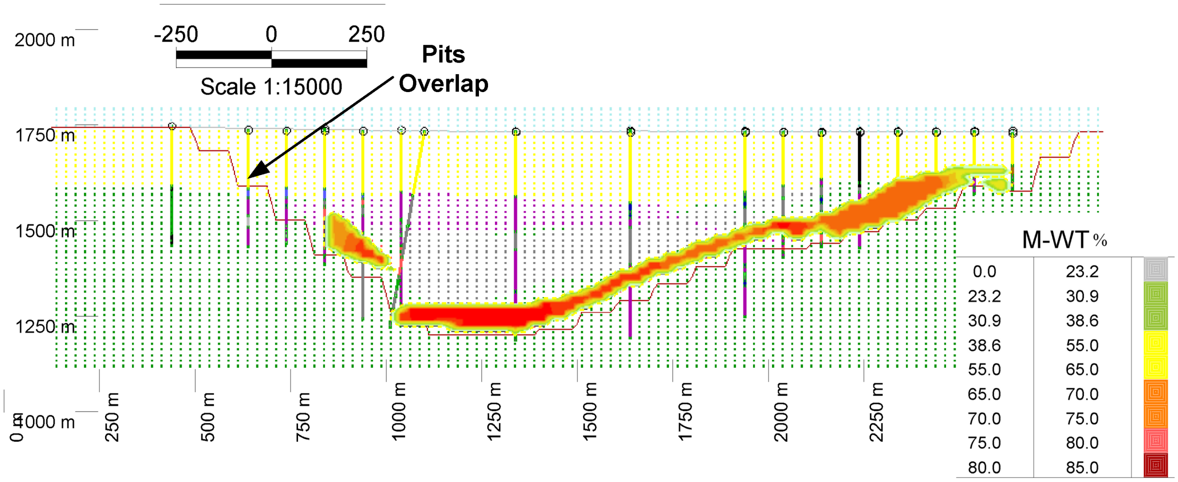


Figure 12- Section 98000 looking west- two pits completely overlap.

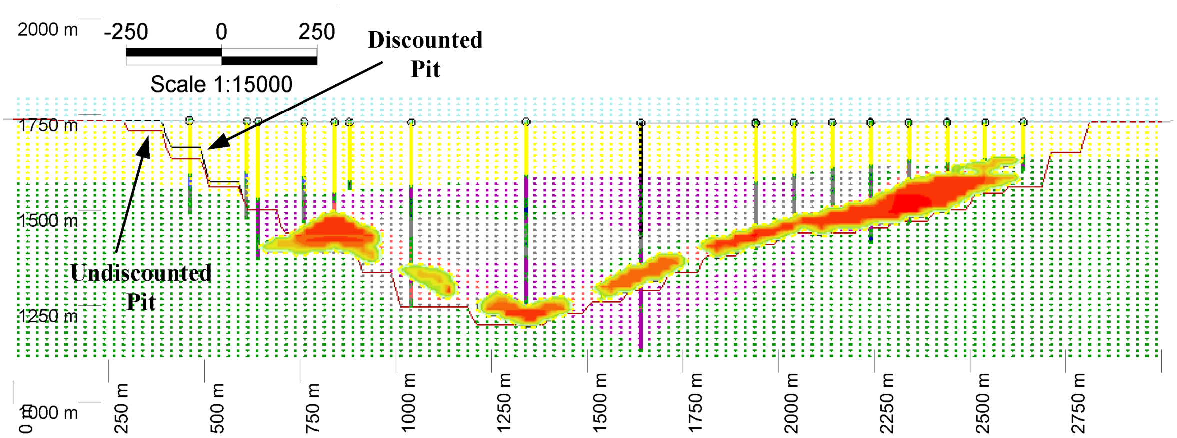


Figure 13- Section 98300 looking west- two pits overlap on the northern side.

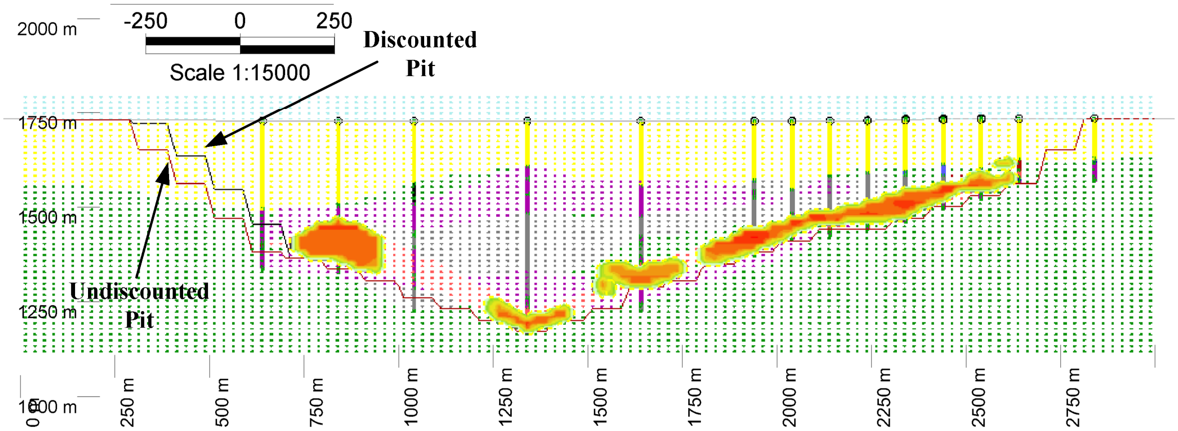


Figure 14- Section 98500 looking west – the difference gets larger.

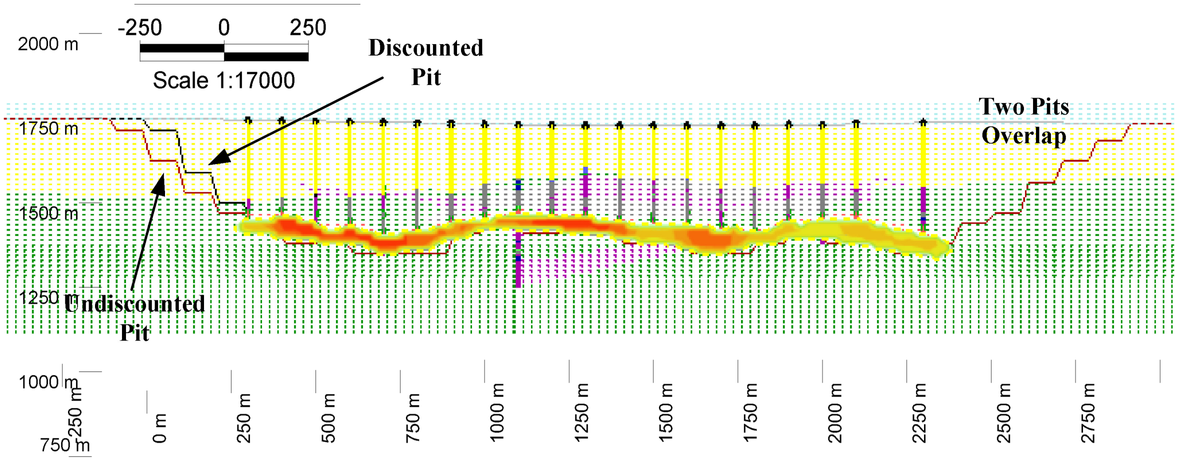


Figure 15- Section 599840 looking north – pits overlap on the east side.

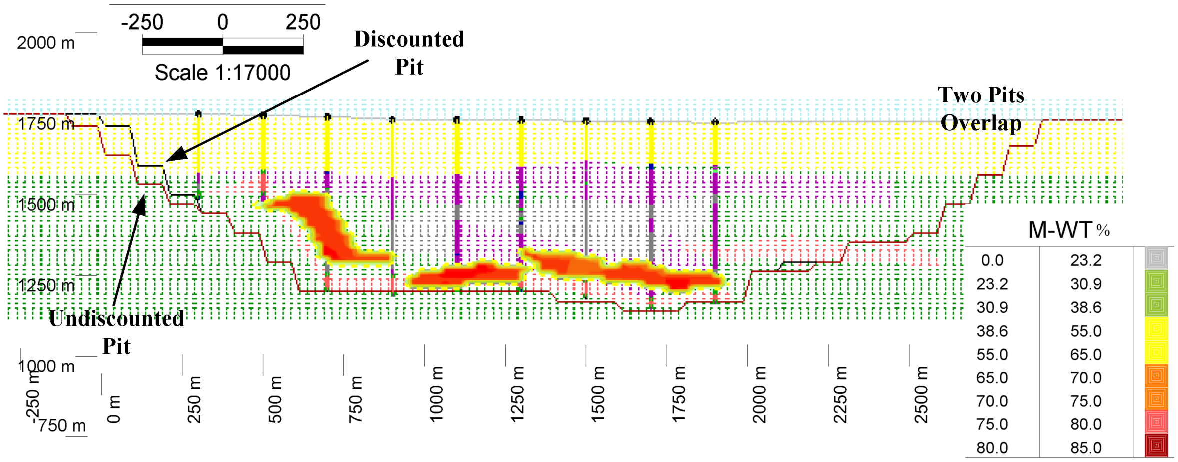


Figure 16- Section 600140 looking north – pits overlap on the east side.

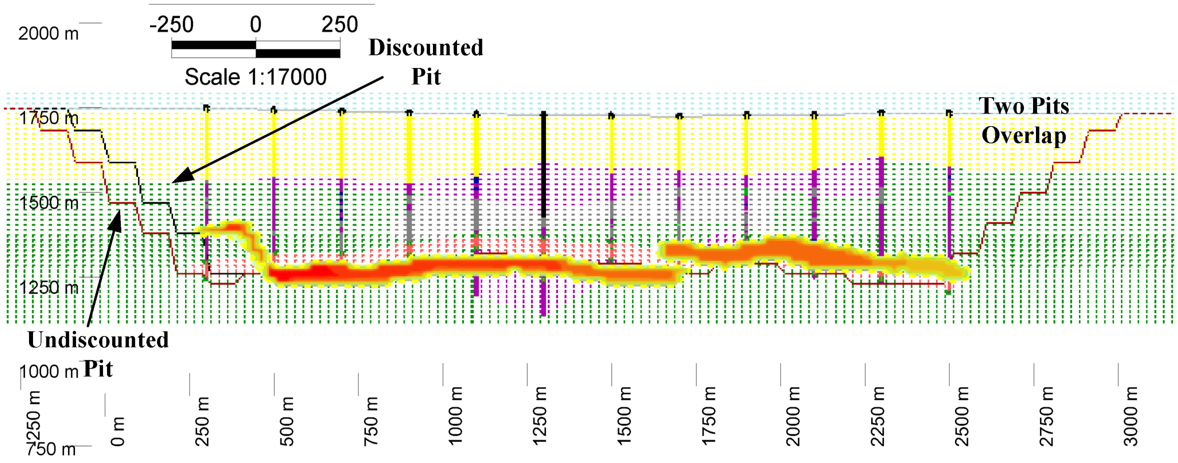


Figure 17- Section 600440 looking north – pits overlap on the east side.

The effect of discounting could have an impact on the optimised final outlines of larger deposits with a smaller annual production targets. The shape of the final pit limits would have a direct effect on the reserves and average grades reported. As a matter of fact the impact of the time aspect of extraction would influence how resources and reserves are classified. The results of the case study are an evidence for how using the classical approach could result in overestimation of reserves. Furthermore, the case study validates that the optimal long term schedule is constrained by the conventional ultimate pit layouts. It also, confirms Caccetta and Hill (2003) theorem stated in the introduction.

5. Conclusions

The paper investigated the validity of the theorem presented by Caccetta and Hill (2003) through a case study using the intelligent agent-based mine planning simulator (IOPS)(Askari-Nasab et al., 2005; Askari-Nasab, 2006; Askari-Nasab et al., 2007; Askari-Nasab and Szymanski, 2007; Askari-Nasab et al., 2008). Caccetta and Hill (2003) theorem states that a final pit limits designed directly by using an optimal long-term schedule will result in a pit outline that is a subset of the conventional final pit layouts generated by the Lerchs Grossmann's (LG) algorithm. The study has been based on the fact that if the optimal schedule is known, then the extraction time of each block would also be known. Given the extraction time, the block values could be discounted. Then, the present value of blocks would be used in determining the pit outline using the LG algorithm. The resultant pit outline using the discounted block values would be based on the maximum present value of the pit rather than the maximum profit. A comparative analysis between the undiscounted and discounted models would then reveal the variation between the two outlines. The intelligent agent-based mine planning simulator, IOPS, was used to determine the optimal push-back schedule prior to determining the final pit outline. The resultant push-back schedule was used to construct a discounted economic block model; this block model was used as input to the LG algorithm.

A case study of an iron ore deposit was completed on a tabular deposit extended in a north-south direction with 139 exploration drillholes. The processing plant was founded on magnetic separators. Consequently, magnetic weight recovery (MWT) of iron ore was used as the main criterion for defining ore. Drillhole compositing was taken place based on similar rock types with three types of ore: top magnetite, oxide, and bottom magnetite. Kriging was used to build the geological block model containing 883,200 blocks that makes a model framework with dimensions of 160X×120Y×46Z blocks. Three thousand iterations of simulation runs using IOPS with a mining capacity of 150 $\textit{million tonne} \times \textit{year}^{-1}$ generated a push-back schedule over 63 years. Given that, no final limits were defined the sixty three years is not the mine life and it is representing the time that is required to extract all the blocks within the block model enforcing feasible slope constraints. Afterwards, the block values were discounted with respect to the IOPS optimal schedule. Bench phases technique was used to determine the final pit limits using the LG algorithm for the discounted and undiscounted models.

The optimal final pit limits for the undiscounted model includes 635 Mt of iron ore, where the discounted model showed 629 Mt of magnetic ore. Both models had an average grade of 73% magnetic weight recovery (MWT%). Although discounting the block values within IOPS was carried out over a long period of 63 years, the pit outlines for the two models did not show a significant difference in shape and the amount of ore and waste. The differences between the two outlines are due to the fact that the decrease in economic block values because of discounting in certain marginal areas is more than the effect of discounting on the cost of removing some of the overlying waste blocks, which are not within the same extraction period. In fact, the revenue generated by the marginal ore blocks are discounted with a higher rate comparing to the waste blocks positioned at an earlier extraction period.

Examination of the shape of the pit on cross sections demonstrated overlap of two pits in most of the areas with deviations in the southern and eastern side of the pit. The analyses and comparisons of the results demonstrate that indeed the optimal long term schedule is constrained by the conventional ultimate pit layouts. Therefore, obtaining the optimal production schedule of a mine by first determining the final pit outline and then generating the schedule is a valid practice. However, this could overstate the reserves of larger deposits with a smaller annual production targets and might lead to over-capacity design.

The intelligent agent framework used in this study provides a powerful basis for addressing the real size open pit mine planning problems. Further focused research is underway to develop and test the models based on intelligent agents to include more critical mine planning variables such as: optimised cut-off grades, mill feed requirements, blending parameters, and stockpile constraints into the intelligent mine planning framework. Stochastic simulation as one of the major entities of the developed models has the ability to address the random field and stochastic variables involved in mine planning. The intelligent agent framework has the capability to be extended for the optimal integration of mining and mineral processing systems, and development of a framework to quantify uncertainty relevant to mine planning and engineering design.

6. References

[1] Askari-Nasab, H., (2006), "Intelligent 3D interactive open pit mine planning and optimization", PhD Thesis Thesis, © University of Alberta, Edmonton, Canada, Pages 167.

[2] Askari-Nasab, H., Awuah-Offei, K., and Frimpong, S., (2004), "Stochastic simulation of open pit pushbacks with a production simulator", *in Proceedings of CIM Mining Industry Conference and Exhibition*, © Edmonton, Alberta, Canada, pp. on CD-ROM.

[3] Askari-Nasab, H., Frimpong, S., and Awuah-Offei, K., (2005), "Intelligent optimal production scheduling estimator", *in Proceedings of 32nd Application of Computers and Operation Research in the Mineral Industry*, © Taylor & Francis Group, London, Tucson, Arizona, USA, pp. 279-285.

[4] Askari-Nasab, H., Frimpong, S., and Szymanksi, J., (2007), "Modeling open pit dynamics using discrete simulation", *International Journal of Mining, Reclamation and Environment*, Vol. 21, 1, pp. 35- 49.

[5] Askari-Nasab, H., Frimpong, S., and Szymanksi, J., (2008), "Investigating the continuous time open pit dynamics", *The Journal of the South African Institute of Mining and Metallurgy*, Vol. 108, 2, pp. 61-73.

[6] Askari-Nasab, H. and Szymanski, J., (2007), "Open pit production scheduling using reinforcement learning", *in Proceedings of 33rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 321-326.

[7] Caccetta, L. and Hill, S. P., (2003), "An application of branch and cut to open pit mine scheduling", *Journal of Global Optimization*, Vol. 27, November, pp. 349 - 365.

[8] Caccetta, L. and Hill, S. P., (2003), "An application of branch and cut to open pit mine scheduling", *Journal of Global Optimization*, Vol. 27, November, pp. 349-365.

[9] Dagdelen, K. and Kawahata, K., (2007), "Oppurtunities in Multi-Mine Planning through Large Scale Mixed Integer Linear Programming Optimization", *in Proceedings of 33rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 337-342.

[10] Denby, B., Schofield, D., and Hunter, G., (1996), "Genetic algorithms for open pit scheduling - extension into 3-dimensions", *in Proceedings of 5th International Symposium on Mine Planning and Equipment Selection*, © A.A.Balkema/Rotterdam/Brookfield, Sao Paulo, Brazil, pp. 177-186.

[11] Deutsch, C. V., (2002), "Geostatistical reservoir modeling", © Oxford University Press, New York, Pages 162 - 166.

[12] Dogbe, G. and Frimpong, S., (2004), "Integrated open pit optimization with periodic material scheduling", *in Proceedings of CIM Mining Industry Conference and Exhibition*, © Edmonton, pp. CD ROM.

[13] Elveli, B., (1995), " Open pit mine design and extraction sequencing by use OR and AI concepts", *International Journal of Surface Mining, Reclamation and Environment*, Vol. 9, pp. 149-153.

[14] Erarslan, K. and Celebi, N., (2001), "A simulative model for optimum open pit design", *The Canadian Mining and Metallurgical Bulletin*, Vol. 94, October, pp. 59-68.

[15] Gemcom Software International, I., (1998-2007), "Whittle strategic mine planning software", ver. 4.00, Vancouver, B.C.: Gemcom Software International.

[16] Johnson, T. B. and Barnes, R. J., (1988), "Application of the maximal flow algorithm to ultimate pit design", in *Engineering design : better results through*

operations research methods, Vol. 8, *Publications in operations research series*, R. R. Levary, Ed. New York, © North-Holland, pp. xv, 713.

[17] Lerchs, H. and Grossmann, I. F., (1965), "Optimum design of open-pit mines", *The Canadian Mining and Metallurgical Bulletin, Transactions*, Vol. LXVIII, pp. 17-24.

[18] Murariu, V. and Svoboda, J., (2003), "The applicability of Davis tube tests to ore separation by drum magnetic separators", *Physical Separation in Science and Engineering*, Vol. 12, 1, pp. 1-11.

[19] Ramazan, S., Dagdelen, K., and Johnson, T. B., (2005), "Fundamental tree algorithm in optimising production scheduling for open pit mine design", *Mining Technology : IMM Transactions section A*, Vol. 114, 1, pp. 45-54.

[20] Ramazan, S. and Dimitrakopoulos, R., (2004), "Traditional and new MIP models for production scheduling with in-situ grade variability", *International Journal of Surface Mining, Reclamation & Environment*, Vol. 18, 2, pp. 85-98.

[21] Sutton, R. S. and Barto, A. G., (1998), "Reinforcement Learning, An Introduction", © The MIT Press, Cambridge, Massachusetts, Pages 432.

[22] Tolwinski, B. and Underwood, R., (1992), "An algorithm to estimate the optimal evolution of an open pit mine", *in Proceedings of 23rd APCOM Symposium*, © SME, Littleton, Colorado, University of Arizona, pp. 399 - 409.

[23] Whittle, J., (1989), "The facts and fallacies of open-pit design," in *Manuscript, Whittle Programming Pty Ltd.* North Balwyn, Victoria, Australia

[24] Wooldridge, M., (2002), "An Introduction to Multi-Agent Systems", © John Wiley and Sons Limited, Chichester, UK, Pages 348.

[25] Yegulalp, T. M. and Arias, J. A., (1992), "A fast algorithm to solve the ultimate pit limit problem", *in Proceedings of 23rd APCOM Symposium*, © AIME, Littleton, Colorado, pp. 391-397.

[26] Zhao, Y. and Kim, Y. C., (1992), "A new optimum pit limit design algorithm", *in Proceedings of 23rd APCOM Symposium*, © SME, Littleton, Colorado, University of Arizona, pp. 423-434.

Open pit limits optimization using linear programming

Eugene Ben-Awuah and Hooman Askari-Nasab

Abstract

Mine production scheduling optimization has been a challenging issue for the mining industry due to computational difficulties encountered. Currently, some of the mine production scheduling algorithms used has limitations in terms of producing a realistic production schedule thereby leading to impracticable schedules. In this project, we have reviewed the traditional linear programming models for ultimate pit limit optimization. We have discussed and modeled the final pit limit optimization problem using transportation algorithm. The mining block precedence relationship has been modeled using linear constraints. From the illustrative example and the case study carried out, it was found out that the linear programming model has the advantage of optimizing in terms of a large number of decision variables and constraints. However, this method in practice results in a number of equations which becomes too large to handle. Further studies must therefore be carried out to reduce the number of equations formed as well as the method of solving the resulting equations for an optimized solution.

1. Introduction

Linear Programming has been identified as having many advantages in solving mine planning and scheduling problems (Gershon, 1983). However due to computational difficulties for real size mining operations, it has not been extensively used in the field of mine optimization. Some of the computational difficulties will be outlined as well as the advantages of linear programming over other algorithms. This project will investigate the formulation of the final pit limit optimization problem using transportation algorithm. The extraction precedence of blocks, which is the most important constraint controlling the mine production sequencing will also be discussed.

Heuristic optimization techniques have been one of the most widely used methods for ultimate pit limit analysis. However these methods on many occasions fail to produce a realistic production planning schedule (Huttagosol and Cameron, 1992). Such impracticable schedules lead to overestimation of cash flows which can result in inappropriate investments.

1.1. Problem definition

Linear Programming Models have been used in solving many problems in the mining industry. Popular among them is the transportation algorithm which has been used in optimizing mine equipment operations and mine schedules. Using the concept of linear programming as applied in the transportation algorithm, this project will optimize the mining sequence of a deposit which has been modeled in one period. This will involve modeling the main constraint that will control the mining sequence in order to obtain the ultimate pit limits.

1.2. Production scheduling optimization

Production scheduling optimization process is a major step in mine planning. It attempts to maximize the net present value of the total profits from the production process while satisfying all the operational constraints such as mining slope, grade blending, ore production and mining capacity during each scheduling period (Gholamnejad and Osanloo, 2007). After the optimized production schedule is obtained, it is used to control the mining operation sequencing and this eventually results in the ultimate pit limit at the end of the mine life.

1.3. Mining sequence

A very important aspect of mine scheduling is the taking into consideration the mining block locations and not just the tonnages or their economic block value. In generating the ultimate pit limit this serves as the main constraint that must be modeled. From Gershon (1983), most linear programming applications used for mine production scheduling analyses define the production levels in terms of tonnages that will be mined at that level. Though this approach has the advantage of reducing the number of constraints thereby reducing the computational time, it has a disadvantage of ignoring the detailed mining sequence per level which is the most important aspect of production scheduling. All effort must be made to ensure that the algorithm being used takes care of the detailed mining sequence with respect to the production block precedence relationships.

1.4. The ultimate pit limit

The ultimate pit limit can be defined as the determination of the final mining limits of a mineral deposit within certain set constraints such that maximum profitability will be derived from the mining process (Hartman, 1992). This is basically done by maximizing the difference between the revenue generated from mining the desirable mineral and the cost of mining the waste material associated with the desirable mineral. In determining the ultimate pit limit, one period of mining is usually used and this period is referred to as the life of mine.

There are different methods of designing the ultimate pit limit. The method used may vary depending on some factors like the size of the deposit, the reliability and size of the data to be used, the cost of data processing time and the experience of the design engineer. Obtaining the ultimate pit limit serves as the starting point for short and long term-range planning activities. The size, geometry and location of the ultimate pit facilitates the planning of important mine infrastructure such as the waste dumps, access roads, processing plant, tailings and water dams, reclamation infrastructures and many other surface structures.

As stated earlier, in modeling the ultimate pit limit certain physical and economic parameters will be assigned to the geologic model. The material that falls within the pit will have to meet two objectives (Kennedy, 1990). A block cannot be mined unless:
All blocks overlying it has been mined.
It is capable of paying for its cost of mining, processing and marketing.
This will result in an ultimate pit limit that maximizes the total profit of the deposit within the physical and economic parameters used. The pit design may change as these design parameters change with time. Figure 1 shows a cross section of an ultimate pit limit as imposed on a block model.

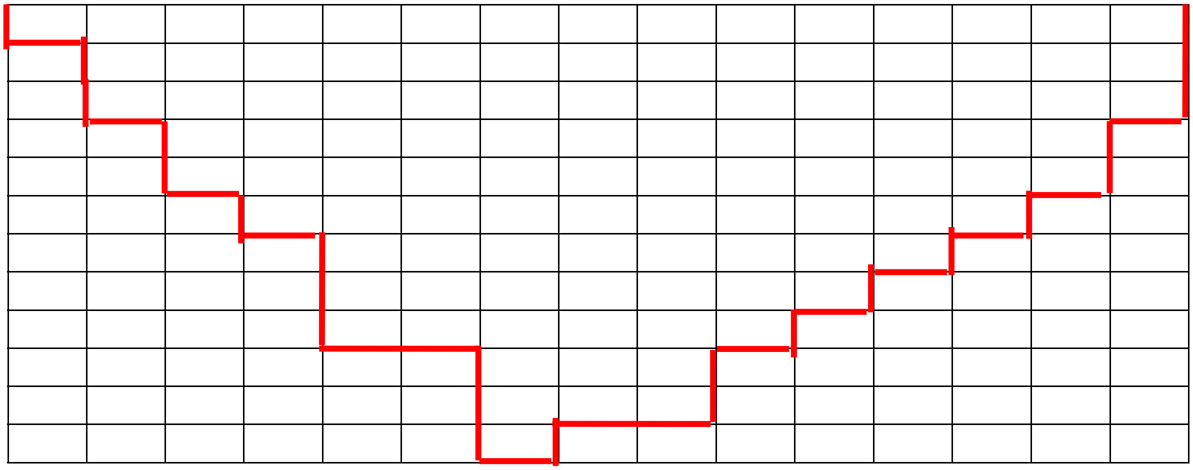


Figure 1: Cross sectional area of an ultimate pit limit as imposed on a block model

1.5. Advantages and disadvantage of using linear programming (LP) to find the ultimate pit limit

1.5.1 Advantages

LP models can optimize in terms of a large number of decision variables and constraints (Gershon, 1983). General sets of LP Models equations can be developed for multiple block models without resorting to a complete redesign of the model for each mine (Gershon, 1983). LP models permits an easy conceptualization for the formulation of the model and provide a good platform for additional study (Huttagosol and Cameron, 1992).

The LP matrix resulting from this formulation exhibits a special structure with a property known as unimodularity which ensures that our decision variables are integer values of 0 or 1 in the optimal solution (Gershon, 1983).

1.5.2 Disadvantages

This method in practice results in a number of equations which becomes too large to handle increasing the cost of processing exponentially (Huttagosol and Cameron, 1992). This will be demonstrated in this project.

2. Theoretical framework and models

2.1. The linear programming model

Mathematical modeling is usually used to find the best solution to a problem that requires that a constraint or sets of constraints about how best to use some given amount of limited resources to achieve some given goals or objectives (Ahmed, 2009). The major steps that are followed in mathematical modeling are:

1) Converting the given problem into a mathematical model that depicts all the important aspects of the problem.
2) Exploring the most suitable solution of the problem.

Linear Programming basically requires that the mathematical model formed has to be made up of linear functions (Ahmed, 2009). The standard form of any linear programming model as stated in (Luenberger and Ye, 2008) is represented by equations (1), (2), (3), (4) and (5) as:

$$\text{maximize} \qquad Z = c_1x_1 + c_2x_2 + \dots\dots\dots\dots\dots\dots + c_nx_n \tag{1}$$

subject to the following constraints:

$$a_{11}x_1 + a_{12}x_2 + \dots\dots\dots\dots + a_{1n}x_n \leq b_1 \tag{2}$$

$$a_{21}x_1 + a_{22} + \dots\dots\dots\dots\dots + a_{2n}x_n \leq b_2 \tag{3}$$

$$\begin{matrix} . & & . \\ . & & . \\ . & & . \end{matrix}$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots\dots\dots\dots\dots + a_{mn}x_n \leq b_n \tag{4}$$

$$and\ all\ x_i \geq 0 \tag{5}$$

Where Z is the objective function representing the parameter to be maximized and x_1, x_2, ……..x_n are real numbers to be determined and are known as decision variables. The decision variables $x_1, x_2, \dots\dots\dots\dots, x_n,$ represent levels of n competing activities. The variables c_1, c_2, …………..c_n are fixed real constants which represent the coefficient of the decision variables in the objective function equation and a_{11}, a_{21}, …………..a_{mn} are also fixed real constants representing the coefficient of the decision variables in the constraints equations. The variables b_1, b_2, …………b_n are fixed real constants in the constraint equation.

In a more efficient vector notation, the linear programming model can be written as stated by equations (6), (7) and (8) as:

$$\text{maximize} \quad c^T x \qquad (6)$$

$$\text{subject to:} \quad \mathbf{A}x \leq b \qquad (7)$$

$$x \geq 0 \qquad (8)$$

Here x is an n-dimensional column vector, c^T is an n-dimensional row vector, **A** is an m x n matrix, and b is an m-dimensional column vector (Luenberger and Ye, 2008).

2.2. The general formulation of a transportation problem

Consider the transportation of some amount of commodities from m origins to n destinations to meet some demand requirements. Origin i contains an amount a_i whereas destination j has a requirement of amount b_j. The unit cost associated with transporting the commodities from the origin i to destination j is c_{ij}. Our objective is to find the most suitable transportation pattern that satisfies all our requirements and minimizes the total transportation cost (Luenberger and Ye, 2008).

This problem can be expressed mathematically in equations (9), (10), (11) and (12) as finding a set of $x_{ij,}$ i=1, 2, ……..,m; j=1, 2, ………..n to:

$$\text{minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij} \qquad (9)$$

$$\text{subject to:} \quad \sum_{j=1}^{n} x_{ij} \leq a_i \quad \text{(i=1, 2, ……., m) (supply constraints)} \qquad (10)$$

$$\sum_{i=1}^{m} x_{ij} \geq b_j \quad \text{(j=1, 2, ………, n) (demand constraints)} \qquad (11)$$

$$x_{ij} \geq 0 \quad \text{(i=1, 2, ….., m; j=1, 2, ……, n)} \qquad (12)$$

In this transportation context, the variables x_{ij} refers to the amount of commodities to be transported from an origin i to a destination j (Luenberger and Ye, 2008).

2.3. The ultimate pit limit problem as a transportation problem

Considering an orebody which has been represented by a geologic and an economic block model, the ultimate pit limit problem can be represented by equations (13), (14) and (15) as (Luenberger and Ye, 2008):

$$\text{maximize} \qquad z = \sum_{k=1}^{p} a_k x_k \tag{13}$$

$$\text{subject to:} \qquad -x_j + x_k \le 0 \quad \text{for all } k \text{ and } j \in D_k \tag{14}$$

$$x_k \in (0,\, 1) \quad \text{for all } k \tag{15}$$

Where: a_k = the economic value of block k,

x_k = portion of material mined from block k,

k = block number (1, 2, ……….p),

D$_k$ = a set of blocks overlying block k, that is, if $j \in D_k$, then j must be removed first before k is.

The constraint which controls the mining sequence as defined by the pit slope requirements is equation (14). The variable x_k can either be zero or one. If block k is to be included in the ultimate pit, then x_k = 1 else x_k = 0. The ultimate pit limit problem can be written in matrix notation as stated in equations (16), (17), (18) and (19) as finding the values of X that (Huttagosol and Cameron, 1992):

$$\text{maximize} \qquad aX \tag{16}$$

$$\text{subject to:} \qquad AX \le 0 \tag{17}$$

$$IX \le 1 \tag{18}$$

$$X \ge 0 \tag{19}$$

Where: X = a column vector of the decision variables,

A = qxp matrix of the block precedence relationships,

I = pxp identity matrix,

a = row vector of economic block values,

1 = px1 column vector of ones.

The matrix A has two non-zero elements of +1 and -1 in each row which is equivalent to a demand-supply incident matrix in a transportation problem. This equivalent optimization model resembles a special structure of minimum cost flow called the transportation problem.

2.4. Formulating the mining sequence constraint

Let's consider figure 2 and 3, which is a block model of 18 blocks which are in two layers of 9 blocks each. To model a scenario like this we start with a decision variable for each block as (Gershon, 1983):

X_{ijk} = 0 if block *ijk* is not mined and 1 if block *ijk* is mined

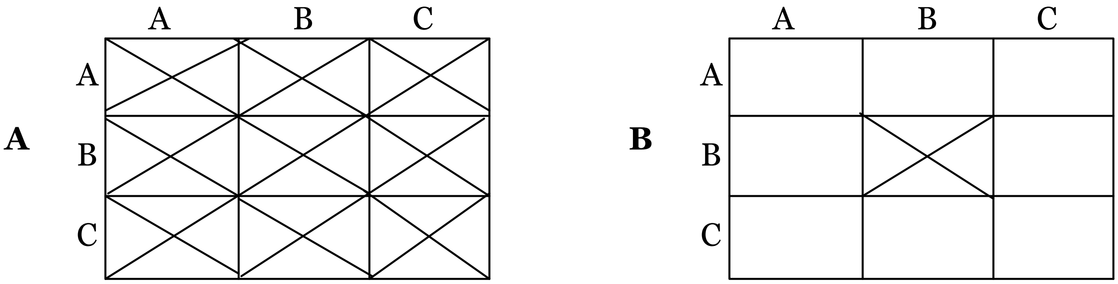


Figure 2: Plan view of the top blocks and bottom blocks in the model (Modified after Gershon, 1983)

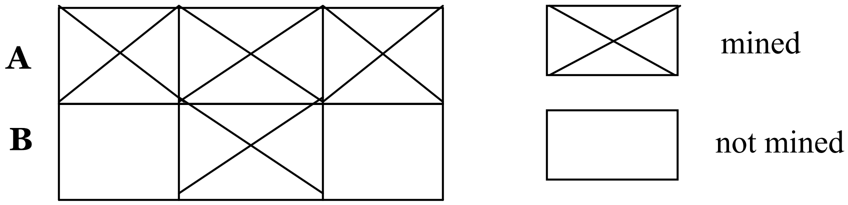


Figure 3: Cross-section of the block model

To be able to mine any given block on level **B**, all the nine blocks on level **A** should be mined first. This means nine separate linear constraints has to be developed for every block on level **B** to ensure a proper mining sequence.

Let's consider the mining of the block X_{BBB}, in figure 2 and 3. To mine this block, the blocks X_{AAA}, X_{ABA}, X_{ACA}, X_{BAA}, X_{BBA}, X_{BCA}, X_{CAA}, X_{CBA} and X_{CCA} must be mined first. Following the formulation discussed earlier, this constraint can be represented mathematically as in equations (20), (21), (22), (23), (24), (25), (26), (27) and (28) (Gershon, 1983):

$$-X_{AAA}+X_{BBB} \leq 0 \tag{20}$$

$$-X_{ABA}+X_{BBB} \leq 0 \tag{21}$$

$$-X_{ACA}+X_{BBB} \leq 0 \tag{22}$$

$$-X_{BAA}+X_{BBB} \leq 0 \tag{23}$$

$$-X_{BBA}+X_{BBB} \leq 0 \tag{24}$$

$$-X_{BCA}+X_{BBB} \leq 0 \tag{25}$$

$$-X_{CAA}+X_{BBB} \leq 0 \tag{26}$$

$$-X_{CBA}+X_{BBB}\leq 0 \tag{27}$$

$$-X_{CCA}+X_{BBB}\leq 0 \tag{28}$$

These set of constraints will have to be written for all the nine blocks on level **B**.

3. Methodology

The principal objective of this project is to carry out a pit limit optimization process that will optimize the profit from the orebody.

This project will realize the objectives by:

- Modeling the equation for the objective function, this is to maximize the net present value of the deposit.
- Modeling the main constraints that will control the mining sequence as defined by the pit slope requirements.
    a) Writing a Matlab Code using the optimization toolbox to solve the resulting transportation equations for an optimized pit limit.

4. Illustrative example with 18 blocks

A computer code was written to implement the design of an ultimate pit limit using the transportation algorithm in linear programming. The computer code was written with Matlab and small synthetic input data was used to verify the developed code. Attributes are associated with each block. These attributes are the 3D location of each block, block tonnage, economic block value, grade of magnetic weight recovery, grade of sulphur, grade of phosphor, ore tonnes and waste tonnes. The main mineral considered for profit is the recoverable Iron which is represented by the grade of magnetic weight recovery. This model contains 18 blocks with dimensions of 100m x 100m on 2 levels with each level having 9 blocks that are 30m thick. Each block may contain both ore and waste material. The model contains a total of 3,116,100 tonnes of ore and a total of 30,347,063 tonnes of material. A production schedule for the orebody model is developed to mine the blocks in one period to maximize the total economic block value within the binding constraints. This results in the geometry of the ultimate pit limit.

A function known as *Linprog* in Matlab was used for this project. *Linprog* uses the Simplex Algorithm to optimize linear programming models. This algorithm is popular for its simplicity and efficiency in solving general transportation problems. The default form of the objective function in *Linprog* is a minimization function. Since our objective function is a maximization of the profit, we multiply our resulting objective function equation by negative one to convert it to a minimization problem.

Linprog finds the minimum of a problem specified by equations (29), (30), (31) and (32) as (Jamshidi et al., 2005):

$$\min f^T x \tag{29}$$

subject to: $$Ax \leq b \tag{30}$$

$$Aeq.x = beq \tag{31}$$

$$x \in (lb, ub) \tag{32}$$

Where f = the coefficient vector of the decision variables in the objective function,
x = the decision variables,
b = a vector in the inequality constraint,
beq = a vector in the equality constraint,
Aeq = the coefficient matrix of the decision variables in the equality constraint,
A = the coefficient matrix of the decision variables in the inequality constraint,
lb = the lower boundary constraint of the decision variables,
ub = the upper boundary constraint of the decision variables.

4.1. Generating the objective function vector

The complete Matlab code developed for this study (Blocks18) can be found in Appendix 1 attached and the symbols used are explained below:

Let X_{ijk} be our decision variable with a 0-1 integer value which represents the decision not to mine or to mine a given block respectively. The objective function for this block model will be a 1 x 18 row vector, denoted by *f,* which contains the economic block values.

4.2. Generating the mining sequence constraint

The mining sequence constraint per block per level is represented by the matrix *y* in the Matlab code and this is a 9 x 18 matrix. For the first block on level 2 to be mined the constraint is that all the 9 blocks on level 1 should be mined first:

$$y = \begin{bmatrix} 1 & 0 & . & . & . & -1 & 0 & . & . & . \\ 1 & 0 & . & . & . & . & -1 & 0 & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ 1 & 0 & . & . & . & . & . & . & 0 & -1 \end{bmatrix}$$

This matrix is generated for all the nine blocks on level 2 and vertically concatenated to form an 81 x 18 matrix denoted by *S* in the program.

The other required constraint states that our decision variables should be less than or equal to one and greater than or equal to zero. These constraints are represented by the matrix *x3* and *x4* respectively and for the 18 blocks on levels 1 and 2, each will have a size of 18 x 18.

$$x3 = \begin{bmatrix} 1 & 0 & . & . & . & . & 0 \\ 0 & 1 & 0 & . & . & . & 0 \\ . & 0 & 1 & 0 & . & . & . \\ . & . & . & . & . & . & .. \\ . & . & . & . & . & . & . \\ . & . & . & . & 0 & 1 & 0 \\ 0 & 0 & . & . & . & 0 & 1 \end{bmatrix} \qquad x4 = \begin{bmatrix} -1 & 0 & . & . & . & . & 0 \\ 0 & -1 & 0 & . & . & . & 0 \\ . & 0 & -1 & 0 & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & 0 & -1 & 0 \\ 0 & 0 & . & . & . & 0 & -1 \end{bmatrix}$$

These three matrices *x3*, *x4* and *S* are vertically concatenated to become a 117 x 18 matrix denoted by *A* in the program. The matrix *A* therefore contains the total number of constraints required for the 18 blocks. The remaining matrices are the matrices on the right hand side of the inequalities for the constraints. These are an 81 x 1 column vector of zeros for the mining sequence matrix represented by *x5*, an 18 x 1 column vector of ones for the *x3* matrix denoted by *x6* and another 18 x 1 column vector of zeros for the *x4* matrix denoted by *x7*. *x5, x6* and *x7* are vertically concatenated to form the matrix *b* of size 117 x 1 used in the program. Table 1 shows the summary of results after the optimization process for the 18 blocks.

Table 1: Summary of results for the 18 blocks

Block ID	Decision Variable	Economic Block Value ($M)
X_{AAA}	1	113.3756
X_{ABA}	1	10.0879
X_{ACA}	0	-3.2118
X_{BAA}	1	15.1904
X_{BBA}	1	3.2324
X_{BCA}	1	22.0876
X_{CAA}	1	18.1649
X_{CBA}	0	-0.4810
X_{CCA}	0	-3.2118
X_{AAB}	1	-3.2118
X_{ABB}	1	-3.2118
X_{ACB}	1	-3.2118
X_{BAB}	1	-3.2118
X_{BBB}	1	-3.2118
X_{BCB}	1	-3.2118
X_{CAB}	1	-3.2118
X_{CBB}	1	-3.2118
X_{CCB}	1	21.0493

Optimized function value = $177.49M

4.3. Discussion of results

It can be seen from the table that the optimized function value is obtained without the mining of blocks X_{ACA}, X_{CBA} and X_{CCA} on level 2. The system had to mine all the waste blocks on level 1 in order to get access to the ore blocks on level 2. The ultimate pit that results is shown by figure 4 and 5.

A

	A	B	C
A	☒	☒	☒
B	☒	☒	☒
C	☒	☒	☒

B

	A	B	C
A	☒	☒	□
B	☒	☒	☒
C	☒	□	□

Legend: ☒ mined □ not mined

Figure 4: Plan view of the top blocks and bottom blocks in the model after optimization

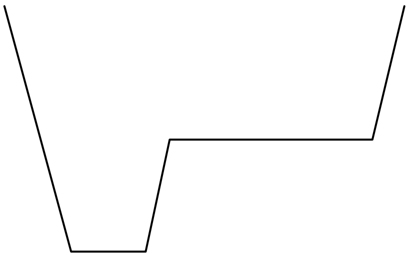


Figure 5: Cross-sectional view of the ultimate pit limit for the model after optimization

(Not to scale)

5. Case Study - Application of LP in the optimization of an iron ore deposit (120 blocks)

Similar to the first block model, some of the data contained in this second block model (Blocks120) used for this project are the 3D location of each block, block tonnage, economic block value, grade of magnetic weight recovery, grade of sulphur, grade of phosphor, ore tones and waste tones. The main mineral considered for profit in this project is the recoverable Iron which is represented by the grade of Magnetic Weight Recovery. This orebody model contains 120 blocks with dimensions of 250m x 250m on 4 levels with level 1 having 4 blocks, level 2 – 16 blocks, level 3 – 36 blocks and level 4 – 64 blocks. Each level is 15m thick. Each block model may contain both ore and waste material. The orebody model contains a total of 11,778,000 tonnes of ore and a total of 248,454,068 tonnes of material. A production schedule for the orebody model is developed to mine the blocks in one period to maximize the total economic block value within the binding constraints. This results in the geometry of the ultimate pit limit.

5.1. Generating the objective function vector

Let X_{ijk} be our decision variable with a 0-1 integer value which represents the decision not to mine or to mine a given block respectively. The objective function for this block model will be a 1 x 120 row vector, denoted by *f,* which contains the economic block values.

5.2. Generating the mining sequence constraint

The mining sequence constraint for the model for all the levels are represented by the matrix *A* in the Matlab Code and have a size of 5088 x 120. The remaining matrices are those on the right hand side of the inequalities for the constraints. These are vertically concatenated to form the matrix *b* resulting in the size of 5088 x 1. The complete Matlab Code for this orebody model containing 120 blocks can be found in Appendix 2 attached. The optimized function value is $310.5608M.

5.3. Discussion of results

From the results, the optimized function value was obtained without the mining of block 3 on level 4. The system had to mine all the waste blocks on levels 1 and 2 in order to get access to the ore blocks on levels 3 and 4. The ultimate pit is shown in figure 6 and 7.

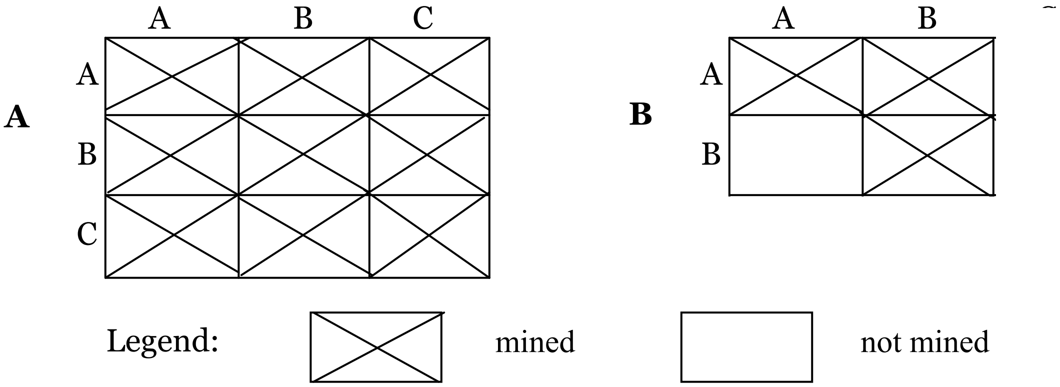


Figure 6: Plan view of some of the top blocks mined and the bottom block that was not mined in the model after optimization

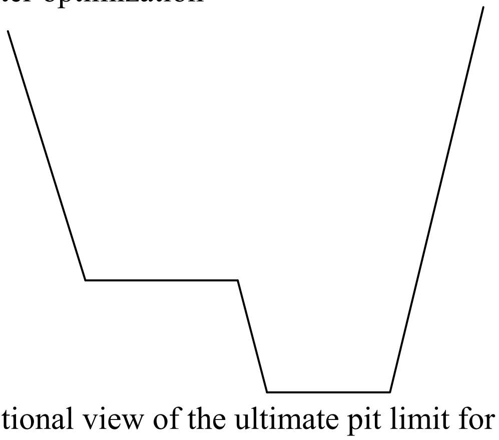


Figure 7: Cross-sectional view of the ultimate pit limit for the model after optimization (Not to scale)

6. Conclusions & future work

The LP model can optimize in terms of a large number of decision variables and constraints (Gershon, 1983). In the examples, there was up to 120 decision variables which were optimized. General sets of LP Models equations can be developed for multiple block models without resorting to a complete redesign of the model for each mine (Gershon, 1983). Developing the model from optimizing 18 blocks to 120 blocks just involved changing the filename and a few other block specific details. The LP matrix resulting from this formulation exhibits a special structure with a property known as unimodularity which ensures that our decision variables are integer values of 0 or 1 in the optimal solution (Gershon, 1983). This was demonstrated in the summary table for the optimized solution.

This method in practice results in a number of equations which becomes too large to handle increasing the cost of processing exponentially (Huttagosol and Cameron, 1992). In the case of the 18 blocks, the size of the constraint matrix, *A* was 117 by 18 which last for about 30 seconds when solving with Matlab *Linprog*. The size of the constraint matrix *A* increased to 5088 by 120 for the 120 blocks which also last for about 9 minutes solving with Matlab *Linprog*. The increase in time is very significant compared with the increase in the number of blocks.

It is recommended that, though this method of optimizing the pit limit is effective, it is computationally very expensive. Further studies must be carried out to reduce the number of equations formed as well as the method of solving the resulting equations for an optimized solution.

7. References

[1] Ahmed, S. M., (2009), "Module Presentation on Decision and Risk Analysis - Linear Programming", Retrieved April 9, 2009 from: http://www.slideshare.net/irs_ijs19/linear-programming-1

[2] Gershon, M. E., (1983), "Optimal Mine Production Scheduling: Evaluation of Large Scale Mathematical Programming Approaches", *International Journal of Mining Engineering*, Vol. 1, pp. 315-329.

[3] Gholamnejad, J. and Osanloo, M., (2007), "Using chance constrained binary integer programming in optimising long term production scheduling for open pit mine design", *Institute of Materials, Minerals and Mining*, Vol. 116, 2, pp. 58-66.

[4] Hartman, H. L., (1992), "SME Mining Engineering Handbook", Society for Mining, Metallurgy, and Exploration, Retrieved from: books.google.ca/books?isbn=0873351002

[5] Huttagosol, P. and Cameron, R. E., (1992), "A Computer Design of Ultimate Pit Limit by Using Transportation Algorithm", *in Proceedings of 23rd International APCOM Symposium*, © pp. 443-460.

[6] Jamshidi, N., Mehrizi, A. A., Davarzani, H. and Molaei, S. R., (2005), "Applied Guide on Matlab 7.6", © Aabed Publishers, Third ed, Pages 653.

[7] Kennedy, B. A., (1990), "Surface Mining", © Society for Mining, Metallurgy and Exploration, Inc, Littleton, Colorado, Second ed, Pages 1194.

[8] Luenberger, D. G. and Ye, Y., (2008), "Linear and Nonlinear Programming", © Springer Science+Business Media, LLC, Third ed, Pages 546.

8. Appendix 1: Matlab Code for 18 and 120 blocks

Matlab code for 18 and 120 blocks

9. Appendix 2: Instructions on running the MATLAB code

Open Pit Limit Optimization using Linear Programming

Table of Contents

----

1. Introduction

This package is built for the optimization of open pit limit using linear programming. The code was written with Matlab and works with the struct file of the block model. The code checks for the number of blocks in the struct file and generate the objective function and the mining sequence constraint based on this number.

After the constraints matrix is generated, a Matlab function known as *Linprog* is called to solve the resulting optimization problem. *Linprog* uses Simplex Algorithm in solving for the optimal solution. For any new set of blocks, the filename of the struct file in the code can be altered to be able to read the new file or the new file can be renamed to match the name used in the code.

Matlab Code filename: optimb120G.mat
Blocks struct filename: Blocks120
The attached demonstration blocks struct file contains 120 blocks.

2. Minimum requirements

Matlab 7.5.0 (R2007b)

3. Getting Started

Open/load the matlab struct file containing the blocks.
Open the matlab file containing the optimization code.
Ensure that the struct filename is the same as that used in the code on line 6 and 7.
Change line 3, 4, 5 and 30 to the number of blocks in the struct file.
Change line 20 to the elevation of the 1st level of blocks at the bottom.
Change line 26 to the number of levels in the struct file excluding the bottom level.
Change line 84 to the inter-level elevation difference.
Once all these parameters are set, execute the optimization from the run icon in Matlab.

4. Results

The results of the optimization will be output to the Matlab workspace. The variable "b" displays the number of blocks optimized. The variable "x" displays the decision variables; whether to mine or not to mine a given block. The variable "fval" displays the optimized function value.

Optimization of blending process in mine production

Samira Kalantari[1] and Hooman Askari-Nasab

Abstract

There are many important issues in the mine planning and management that can be solved with different optimization techniques, therefore a wide range of optimization problems has been covered in the mining industry. In this project, we will use linear programming method (LP) to optimize the mine production plan. The goal is maximizing the net present value of mine production schedule, while meeting mining and processing capacities constraints. Optimization of the mine production scheduling problem, by linear programming method (LP) facilitates blending strategies that would satisfy the mill grade constraints.

1. Introduction

Over the years, optimization techniques have been applied to improve the efficiency of many industrial systems. Mine planning plays an important role in the overall efficiency of mining operations; in other words, the success of mine production is highly dependent on the mine scheduling and mine planning. There are many important issues in the mine planning and management that can be solved with different optimization techniques, therefore a wide range of optimization problems has been covered in the mining industry. Finding the optimal blend of multiple ore sources in mine production, maximizing the net present value of mining operations, and finding the optimal final pit limits of a mine are some of the main problems in the mine scheduling. An important problem in an open pit mine is maximizing the net present value of the mine production in a way that the final mine product meets all of the desired constraints. In addition, optimal blending of the mine products is a critical stage of mine planning in order to obtain final product economically and meet the requirements of the processing plant (Cacetta, 2007).

1.1. Linear Programming

The aim of mine production planning is to find a plan for the mining process in order to achieve the desired quality and quantity specifications (Chanda, 1995). One of the most common methods in solving optimization problems is linear programming. The function that should be either minimized or maximized is called the objective function. Linear programming is applied to the optimization problems such as maximizing or minimizing an objective function. These constraints may include equalities and inequalities. In this

[1] - MSc. Candidate in Mining Engineering

method, the constraints and decision parameters that are used to optimize the objective function are either linear or assumed to be so. There is a general form for the objective function and various constraints (Ferguson, 2009):

Maximize or minimize $Z = \sum_{i=1}^{n} f_i * x$

Z = the objective function

$f =$ The net present value of the mine production

$x =$ The answer of the optimization problem subjected to:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b1 \\ a_{21}x_2 + a_{22}x_2 + \ldots + a_{2n}x_n \le b2 \\ \ldots. \\ \ldots. \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le bm \end{cases}$$

Where:

a_{ij} (i=1,2,…,m; j=1,2,…,n) are the coefficients and will form the matrix of constraints,

b_i (i=1,2,…,m) are the boundaries.

Or we can write:
$\min f^T x$ such that :

$$\begin{cases} A.x \le b \\ Aeq.x = beq \\ lb \le x \le ub \end{cases}$$

Where matrix A is the matrix of constraints with m rows and n columns,

lb is the lower bound of x,

ub is the upper bound of x (Ferguson, 2009) .

The linear programming model defined above is used to solve the mine production scheduling problem. Like many industries, Linear Programming is widely used to optimize mining problems. One of the advantages of using linear programming to solve mine production optimization is that it can optimize the problem with several numbers of constraints and restrictions (Gershon, 1983).

2. Theoretical framework and models

In this project, the goal is to find the maximum net present value of the mine production schedule over all extraction periods, such that the final product meets the desired constraints including grade and tonnage. The 3-D fixed block model is the most common method of modeling ore bodies (Gershon, 1983). In this model, the whole ore body is divided to some fixed-size blocks.

2.1. Objective function

The objective function of the LP model is to maximize the net present value (Ramazan and Dimitrikapoulos, 2004):

$$Maximize \sum_{t=1}^{p} \sum_{i=1}^{n} C_i^t * x_i^t$$

Where:
p is the number of the periods;

C_i^t is the net present value of the portion of the block which has been extracted;

x_i^t is the portion of block i that has to be blended with another portion of block j, therefore x_i^t is a number between 0 and 1;

n is the total number of blocks to be extracted.

The problem has various constraints that should be considered during the optimization, such as grade blending constraints, processing capacity constraints, and mining capacity constraints (Ramazan and Dimitrikapoulos, 2004).

2.2. Grade blending constraints

There is a maximum desired grade for each material $G_{\max}$ and each time period t in which the average grade of block i, g_i should not exceed that value. Therefore, this constraint is an upper bound constraint (Ramazan and Dimitrikapoulos, 2004). Equation (1) shows the maximum grade constraint.

$$\sum_{i=1}^{n} (g_i - G_{\max}) * O_i * x_i^t \leq 0 \qquad (1)$$

Where O_i is the tonnage of ore in the ith block.

Also, there is a minimum desired grade for each material $G_{\min}$ and each time period t in which the average grade of block i, g_i, should not be less than that value (Ramazan and Dimitrikapoulos, 2004). This is a lower bound constraint which has been shown in equation (2).

$$\sum_{i=1}^{n} (g_i - G_{\min}) * O_i * x_i^t \geq 0 \qquad (2)$$

2.3. Processing capacity constraint

The total tonnage of the ore processed in each period should not exceed the maximum value of the processing capacity. This is an upper bound constraint (Ramazan and Dimitrikapoulos, 2004):

$$\sum_{i=1}^{n}(O_i * x_i^t) \leq PC_{\max} \tag{3}$$

Where

$PC_{\max}$ is the maximum processing capacity in each period.

The total tonnage of the ore processed in each period should not be less than the minimum value of the processing capacity. This is a lower bound constraint:

$$\sum_{i=1}^{n}(O_i * x_i^t) \geq PC_{\min} \tag{4}$$

Where

$PC_{\min}$ is the minimum processing capacity in each period (Ramazan and Dimitrikapoulos, 2004).

2.4. Mining capacity constraints

The total amount of material including ore and waste material in each period should not be greater than the total capacity of the mining process. This is an upper bound constraint:

$$\sum_{i=1}^{n}(O_i + W_i) * x_i^t \leq MC_{\max} \tag{5}$$

Where

W_i is the waste tonnage of the block i;

$MC_{\max}$ is the maximum capacity of the total available equipments.

The total amount of materials including ore and waste in each period should not be less than the minimum mining capacity in each period. Therefore, this is a lower bound constraint:

$$\sum_{i=1}^{n}(O_i + W_i) * x_i^t \geq MC_{\min} \tag{6}$$

Where

$MC_{\min}$ is the minimum of the available mining capacity (Ramazan and Dimitrikapoulos, 2004).

2.5. Reserve constraints

The whole block of i should be processed completely at the end of all the periods, in other words, at the end of the last period, all of the blocks should be processed (Ramazan and Dimitrikapoulos, 2004):

$$\sum_{t=1}^{p} x_i^t = 1 \tag{7}$$

Equation (7) is an equality constraint.

3. Methodology

We will illustrate the formulation of the mathematical model with a small synthetic model containing 18 blocks. The blocks' attributes are: three dimensional coordinates of the block (X, Y, and Z), block tonnage, economic block value (EBV), grade of Phosphor (%P), grade of Sulphur (% S), and magnetic weight recovery of Fe (%MWT), ore tonnes, waste tonnes, ore dollar value, and waste removal costs.

The objective function that has to be maximized is the net present value of the cash flow. We assume scheduling over three periods. Another assumption is that within these periods all of the blocks will be extracted. Therefore, the objective function (f) would have $(\text{number of periods} \times \text{number of blocks})$ rows, and only one column.

Number of periods (t) =3
Number of blocks (i) = 18
f is a matrix of (18,1) elements.

Maximize or minimize $Z = \sum_{i=1}^{n} f_i * x$

Where f is equal to:

$$\frac{(EBV_i)}{(1+i)^t} \tag{8}$$

Where i is the interest rate, and the interest rate is assumed to be equal to 15%. We used the MATLAB optimization toolbox for implementing the optimization problem. MATLAB optimization toolbox has a linear programming solver, the Linprog function. Linprog solves linear programming problems using the revised simplex algorithm. The general format of the objective function in Linprog forms a minimization problem. The objective of the mine scheduling problem is to maximize the net present value; therefore, the objective function should be multiplied in a negative sign.

$\Rightarrow Minimize - f = Maximize + f$

The code for generating the matrix f is in the appendix.

Therefore, for the first period, the first 18 rows of the objective function should be:

$$-\frac{(EBV_i)}{(1+0.15)^1}\ \forall i \in \{1,2,...,18\} \tag{9}$$

The second 18 blocks of the objective function which shows the second period should be equal to:

$$-\frac{(EBV_i)}{(1+0.15)^2}\ \forall i \in \{1,2,...,18\} \tag{10}$$

And finally, the third 18 rows (37 to 54) of the objective function which show the third period should be equal to:

$$-\frac{(EBV_i)}{(1+0.15)^3}\ \forall i \in \{1,2,...,18\} \tag{11}$$

The next step is to generate the matrix of inequality constraints (matrix **A**) which contains the grade, processing capacity, and mining capacity constraints.

3.1. Grade blending constraints

According to equation (1), the main ore is Fe, therefore the G_{max} is the maximum desired grade of Fe in each period, and g_i is the grade of Fe for each block, and O_i is the ore tonnage for each block. As the total number of blocks is 18, and the number of periods is 3, therefore the matrix for this constraint will have 3 rows and columns. In the first row which indicates the first period, the first 18 elements are non-zero elements, but the other 36 elements (19 to 54) should be zero. The same rule applies to the two other rows.

The MATLAB code generating the matrix of grade constraints is provided in appendix. This matrix is called **A** and the vector G_{max} has 3 elements:

G_{max} =[G_{maxi}] $\forall i \in \{1,2,3\}$

Because the inequality equation is in the form of A.x<=b, and the grade constraint should be less than zero, therefore, there is no need to multiply the matrix by a negative sign.

vector G_{max} is:

[0.9, 0.9, 0.9]

This shows that the grade constraint is equal for all of the three periods. The next grade constraint is for the minimum desired grade of Fe in each period $G_{min,}$ which is shown in equation (2). Therefore, this constraint is a matrix with 3 rows (indicating three periods), and 54 columns.

G_{min}=[G_{mini}] $\forall i \in \{1,2,3\}$

matrix G_{min} is assumed to be:

G_{min}= [0, 0, 0]

As this constraint is an inequality which is greater than or equal to zero, the matrix should be multiplied by -1 to match the other constraints. In the first row of this matrix which is indicating period number one, the first 18 elements is shown in equation (12).

$$-(g_i - G_{\min}(1,1)) * O_i \qquad (12)$$

And the other 36 elements (19 to 54) are zero. In the second row, for the period number two the second 18 elements are shown in equation (13).

$$-(g_i - G_{\min}(1,2)) * O_i \qquad (13)$$

And the other 36 elements (1-18, and 37-54) are zero. Equation (14) shows the third 18 elements in the third row of the constraint matrix.

$$-(g_i - G_{\min}(1,3)) * O_i \qquad (14)$$

This matrix is called matrix K. At this stage, to extend the matrix of constraints the 2 matrices (A and K) should build an augmented matrix; the augmented matrix is called matrix **S**.

S= [**A**; **K**]

where ; will concatenate matrix **K** with matrix **A.** Therefore, matrix **S** is a (6, 54) matrix. The code for generating matrix **K** and **S** is provided in appendix.

3.1.1 Grade of phosphor

$$\sum_{i=1}^{n} (g_i - P_{\max}) * O_i * x_i^t \leq 0$$

At this stage, in equation (15), Pmax indicates the maximum desired grade of phosphor in each period, and gi indicates the grade of phosphor for each block. Therefore, the matrix Pmax has 1 row and 3 (number of periods) columns.

The maximum desired grade vector for phosphor is as follows:

Pmax= [0.002%, 0.0025%, 0.0025%]

Which means that grade of phosphor of the blocks being extracted in the first period should not be greater than 0.002%. Similarly, the phosphor grade of the blocks being extracted in the second and third periods should not be greater than 0.0025%.

3.1.2 Grade of sulphur

$$\sum_{i=1}^{n} (g_i - S_{\max}) * O_i * x_i^t \leq 0 \qquad (16)$$

Where in equation (16), S_{max} is the maximum desired grade of Sulphur of each block, in each period.

In this project the matrix S_{max} is as follows:

S_{max}=[0.028%, 0.031%, 0.031%]

Therefore, the Sulphur grade of the blocks being extracted in the first period should not exceed the value of 0.028%; the same rule applies to the second and third periods.

3.2. Processing capacity constraint

$$\sum_{i=1}^{n}(O_i * x_i^t) \le PC_{\max} \tag{17}$$

Where in equation (17), $PC_{\max}$ is the maximum processing capacity in each period.

$PC_{\max}$ =[2,200,000, 2,000,000, 2,000,000]

This matrix is called **Q** and has 3 rows and 54 columns. For this matrix as the left hand side is the ore tonnage of the 18 blocks, the three rows would be the same, but in the first row which is indicating the first period the first 18 elements, in the second row the second 18 elements, and in the third row the third 18 elements are nonzero.

To extend the matrix of constraints the 2 matrices (**S**,**Q**) should build an concatenated matrix. The concatenated matrix is called **V**.

V=[**S**;**Q**]

Where matrix **V** adds the matrix **Q** to the matrix **S**.

The resulting V matrix is (9, 54). The code for generating this matrix is provided in the appendix.

There is a constraint for the minimum processing capacity in each period ($PC_{\min}$):

$$\sum_{i=1}^{n}(O_i * x_i^t) \ge PC_{\min}$$

In this project the vector $PC_{\min}$ is assumed to be:
$PC_{\min}$=[0,0,0]

As the inequality in this case is greater than or equal to zero, the matrix should be multiplied by -1. As the matrix in the left hand side is the ore tonnage of each block, this matrix is the same as matrix **Q**. The difference is the negative sign, and the right hand side. This matrix is called **E**.

The extended matrix of constraints is the concatenated matrix of **V** and **E**. The concatenated matrix is called **C**.

C=[**V**;**E**]

The matrix **E** is (12,54).

The code for generating matrix **E** is provided in the appendix.

3.3. Mining capacity constraint

Equation (5) is used at this stage. The matrix contains the sum of the ore tonnes and waste tonnes of each block, and it has 3 rows and 54 columns. As the summation of the waste and ore tonnage for each block is constant in all of the periods, the first 18 elements of the first row, the second 18 elements of the second row, and the third 18 elements of the third row would be the same. This matrix is called **P**, and has 3 rows and 54 columns.

The concatenated matrix of constraints is called matrix **U**.

U=[**C**;**P**]

MC$_{max}$=MC$_{max(i)}$ $\forall i \in \{1,2,3\}$

In this problem the matrix MC$_{max}$ assumed to be equal to:

[1800000,1800000,1800000]

Matrix **U** has 15 rows and 54 columns. In this stage, the matrix of inequalities (matrix of constraints) is completed. The code for generating matrix **P** is provided in appendix.

3.4. Reserve constraints

This constraint is an equality constraint:

$A_{eq} * x = b_{eq}$

All of the blocks have to be extracted during these three periods; in other words, the summation of x_i during three periods should be equal to one.

$x_1^{\,1} + x_1^2 + x_1^3 = 1$

$x_2^1 + x_2^2 + x_2^3 = 1$

...

...

$x_{18}^1 + x_{18}^2 + x_{18}^3 = 1$

Therefore, for the matrix of reserve constraints, in the first row, the first, 19th and 37th element should be equal to one. For the second row, the second, 20th, and 38th element should be equal to one, similarly this rule applies to the rest of the rows. As a result, this matrix contains 18 (number of the blocks) rows, and 54 (number of the periods*number of the blocks) columns.

On the right hand side, b$_{eq}$ is a unit vector with 18 rows and 1 column. The code for generating matrix **A**$_{eq}$ is provided in appendix .

3.5. Normalizing the matrices

In order to solve this problem using Linprog, the scale of the matrices elements should be in the same order, therefore, both matrices f and U should be normalized. To normalize vector f, since this vector is a column vector, the norm of the matrix should be calculated, and each element should be divided by the matrix norm.

F= f./norm(f)

Where ./ indicates element wise division.

In addition, the matrix of constraints should be normalized, to do so, the norm of each row should be calculated, and afterwards, each element should be divided by the corresponding norm. Finally, the matrix of constraints (**U**) should be normalized.

4. Implementation

4.1. Optimization of the problem for 18 blocks

In order to use Linprog to solve the optimization problem, there should be upper bound and lower bounds for the answer. In other words, there should be limits for x_i^t so the answer would be in the desired range. Since x_i^t indicates the portion of the i^th^ block which has to be mined in the period t, this answer couldn't be lower than zero, also, it couldn't be greater than one. As a result, the lower bound would be a zeros vector with 54 rows and 1 column. Similarly, the upper bound should be unit vector with the same size as the vector x (answer) with 54 rows and just 1 column.

Lower bound (lb) = zeros (54,1)

Upper bound (ub) = ones (54,1)

To solve this optimization problem, the MATLAB Linprog has been used. The general code for using Linprog to solve this problem is as follows:

[X, fval]= Linprog(f,U3,b,Aeq,Beq,lb,ub)

And the answers for this optimization problem are provided in table 1.

Table 1: Linprog answers for optimizing the 18 blocks.

Period Number	1	2	3
x1	0.6665	0.3335	0.0000
x2	1.0000	0.0000	0.0000
x3	0.0716	0.2634	0.6651
x4	0.4273	0.5727	0.0000
x5	0.0000	1.0000	0.0000
x6	0.0000	0.7668	0.2332
x7	0.0000	1.0000	0.0000
x8	1.0000	0.0000	0.0000
x9	0.0716	0.2634	0.6651
x10	0.0716	0.2634	0.6651
x11	0.0716	0.2634	0.6651
x12	0.0716	0.2634	0.6651
x13	0.0716	0.2634	0.6651
x14	0.0716	0.2634	0.6651
x15	0.0716	0.2634	0.6651
x16	0.0716	0.2634	0.6651
x17	0.0716	0.2634	0.6651
x18	1.0000	0.0000	0.0000

These values show the portion of each block which has to be mined in each particular period, so that all of the constraints will be satisfied. The objective function value (fval) is as follows:

Fval=-0.8939

Since the Linprog has minimized the objective function, the fval should be multiplied by -1:

Fval*(-1) = 0.08939

In addition, since the original function (f) had been normalized, it should be multiplied by norm of the matrix f:

Fval*norm (f) =0.08939*(1.6010*10^8) =1.431100 ($million)

4.2. Average grade of Fe in each period

To find the average grade of each period, equation (18) should be applied.

$$\frac{\sum x_i^t * O_i * g_i}{\sum x_i^t * O_i} \tag{18}$$

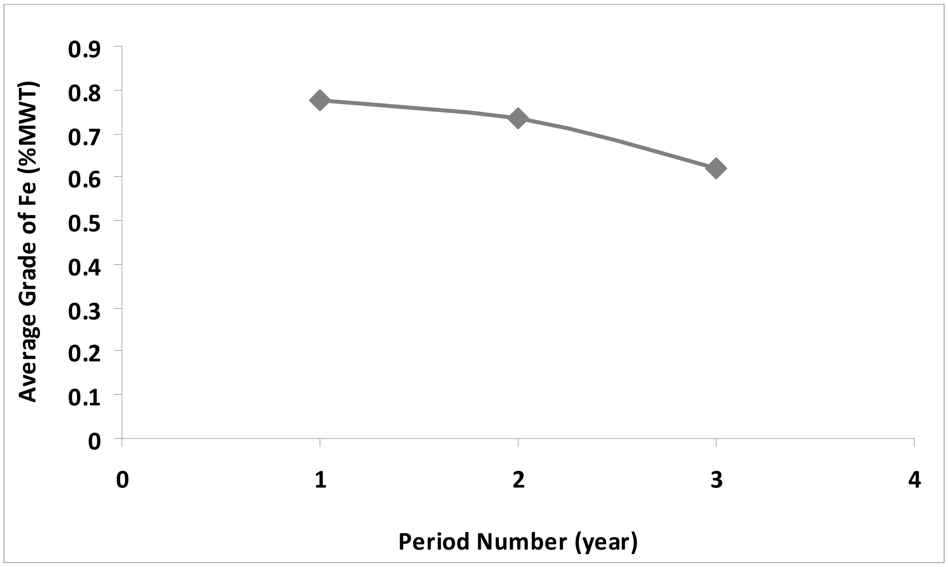


Figure 1: Average grade of Fe (%) in each period

4.3. Average grade of phosphor and sulphur in each period

Table 2 shows the average grade of Phosphor and Sulphur in each period after the optimization has been done, and the results are shown in figure 2 and figure 3 respectively.

Table 2: Average grade of Phosphor and Sulphur in each period.

Average grade (%)	Period #1	Period #2	Period #3
Phosphor	0.00199	0.00192	0.0016
Sulphur	0.02480	0.02220	0.01590

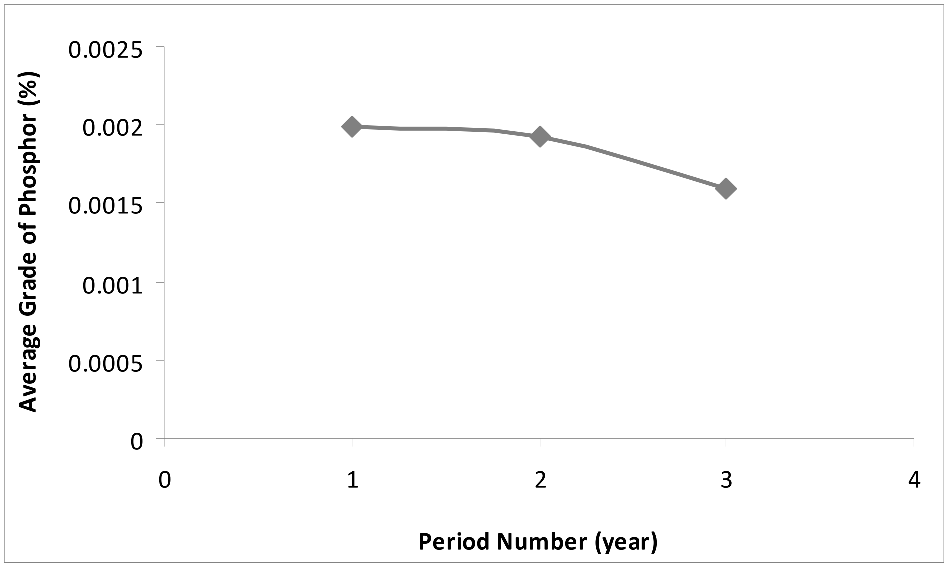


Figure 2: Average grade of Phosphor (%) in each period

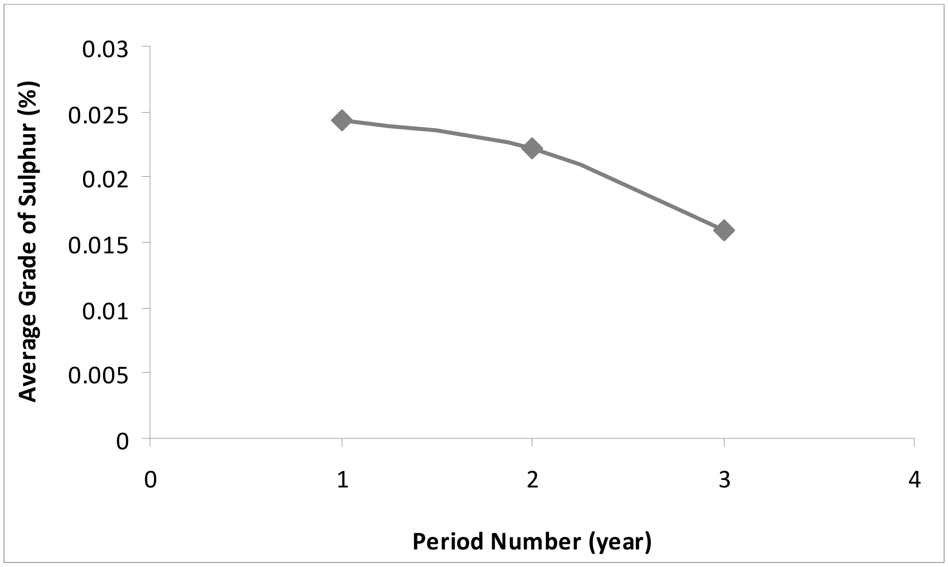


Figure 3: Average grade of Sulphur in each period

4.4. Summation of the ore and waste tonnes in each period

Figure 4 shows that the summation of tonnage of waste mined and the tonnage of ore processed in each period. The summation of ore and waste does not exceed 11,000,000 tonnes, which is the maximum mining capacity for each period. In other words, figure 4 shows that the minimum and maximum mining capacity for each period is satisfied.

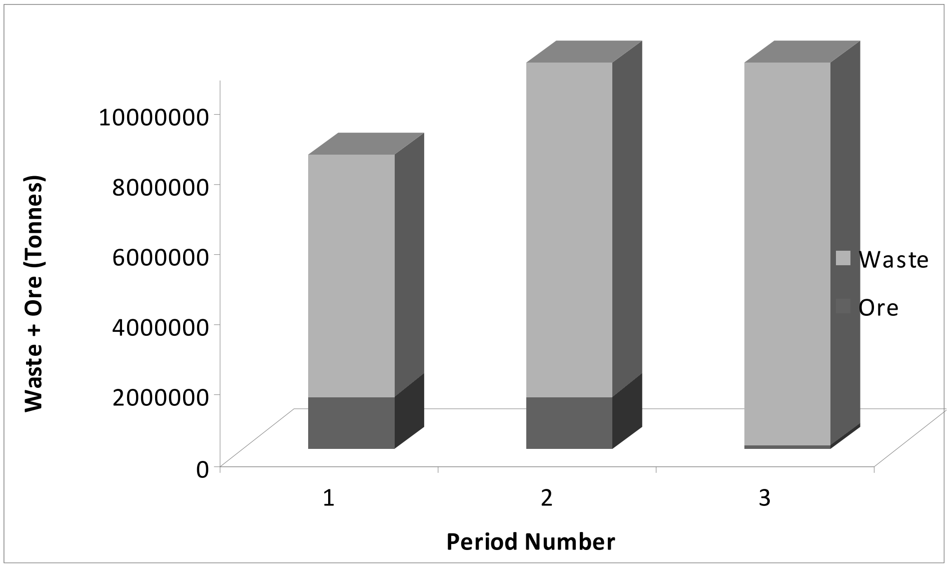


Figure 4: Summation of waste and ore tonnage in each period

4.5. Net present value (NPV)

Figure 5 shows that the maximum cash flow is achieved in the first period in order to maximize the total cash flow in all of the periods. In the second period, the cash flow decreases, and finally in the third period, the cash flow will be negative, which is the the result of the reserve constraints. In the reserve constraint we assumed that we want all the blocks to be extracted within these three periods, therefore in the third period the cash flow becomes negative.

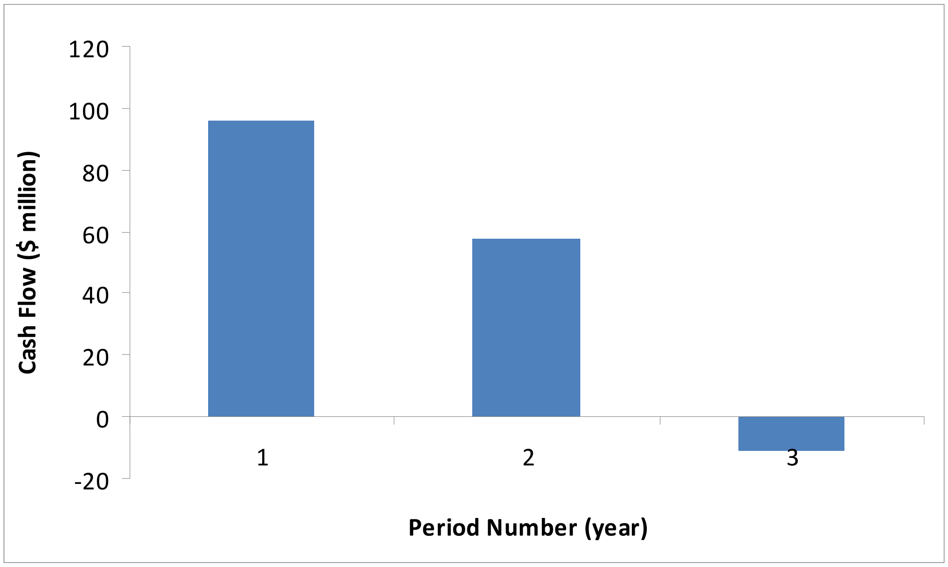


Figure 5: Cash flow in each period

5. Illustrative example

5.1. Optimization of the problem for 120 blocks

For the data with different number of blocks the general procedure will be the same, only the size of the matrices will change with respect to number of periods. We have examined a scheduling case by 120 blocks, over ten periods; therefore the size of all of the matrices will change. The code for solving the optimization problem for 120 blocks is provided in appendix G. With 120 blocks, and 10 periods, the matrix of objective function will have 1 column and 1200 rows.

The matrix of constraints in this problem has 60 rows and 1200 column, the number of rows indicates that there are 6 constraints for 10 periods, and the number of columns shows that 120 blocks will be extracted in 10 periods. The problem has been solved using MATLAB Linprog, and the final answer for x values is in appendix.

5.2. Results of optimization for 120 blocks

Figure 6 shows the average grade of ore in each period, from this figure it can be seen that the average grade of Fe is decreasing as the number of periods increases, it shows that in order to obtain the maximum cash flow, the blocks with the higher grade of Fe should be mined in the first periods.

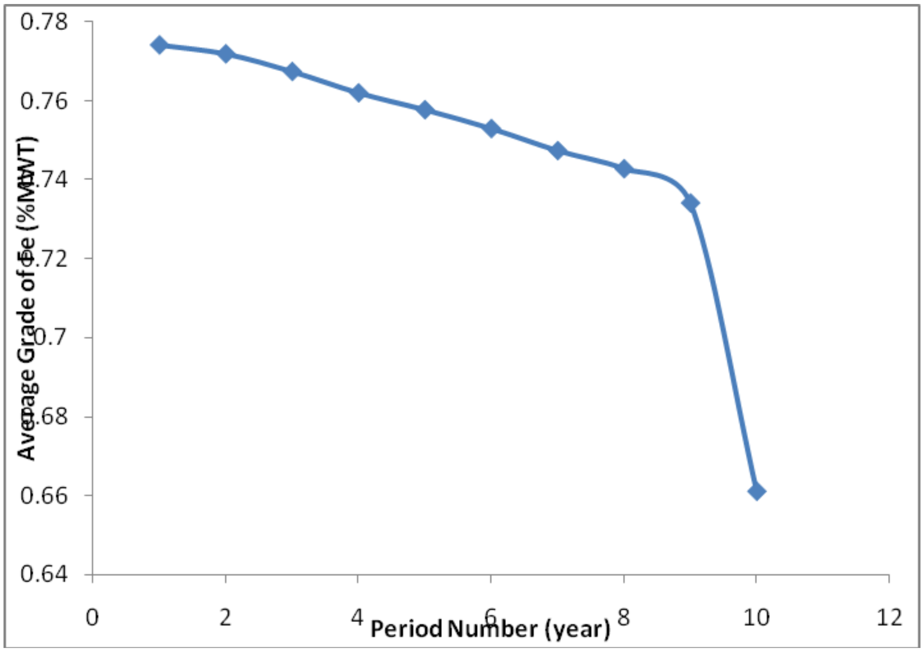


Figure 6: Average grade of Fe (%)

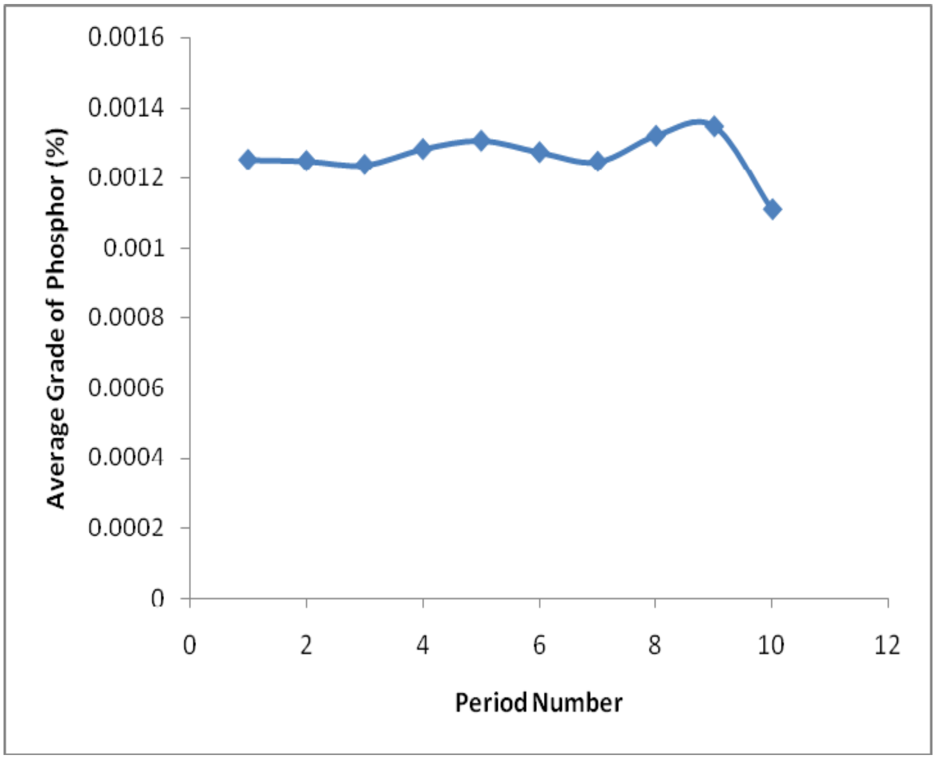


Figure 7: Average grade of Phosphor (%)

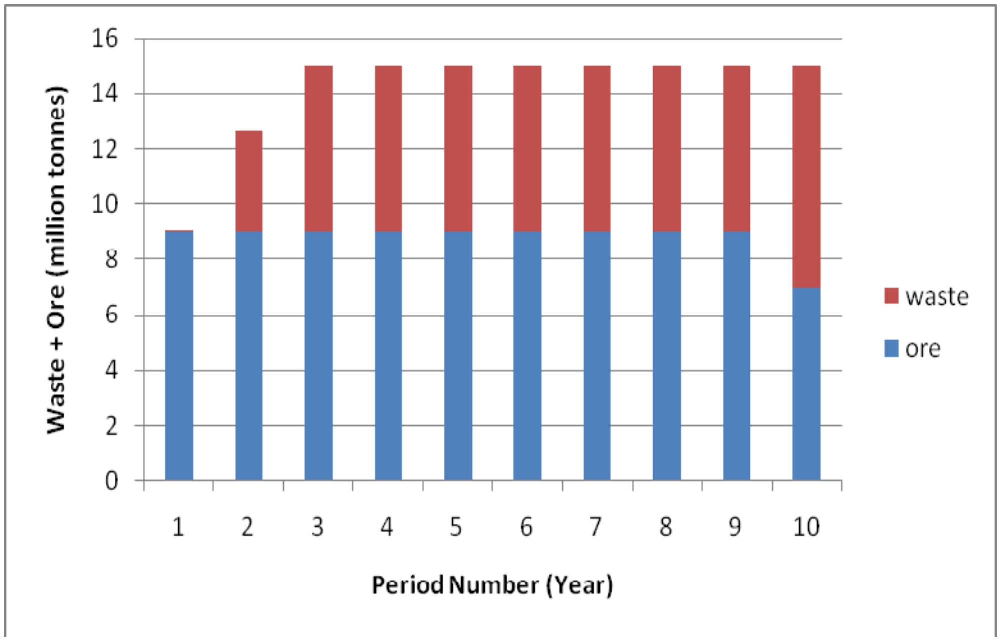


Figure 8: Summation of waste and ore tonnage

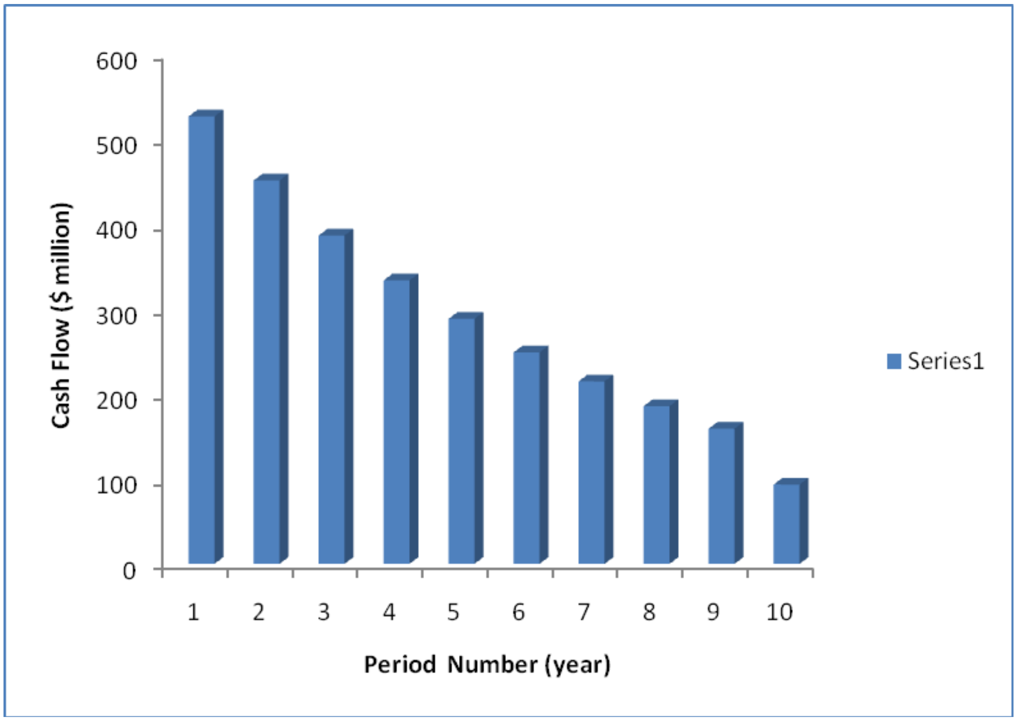


Figure 9: Cash flow

Figure 9 shows that the maximum cash flow are from the first period, and it decreases when the number of period increases. Therefore, in the last period we have the smallest cash flow.

6. Conclusions & future work

There are many important issues in the mine planning and management that can be solved with different optimization techniques, therefore a wide range of optimization problems has been covered in the mining industry.

As it can be seen from the results of optimization of two different sets of data in this project, the amount of ore that has to be extracted is greater in the first periods comparing to the last ones. Since the goal is maximizing the net present value this result is reasonable. In order to prevent extracting a big amount of waste in the last periods, the reserve constraint should be considered.

7. References

[1] Cacetta, L., (2007), "Handbook of operations research in natural resources", Vol. 99, © Pages 561.

[2] Chanda, E., K, C, (1995), "Optimal blending of mine production using goal programming and interactive graphic systems", *International journal of surface mining, reclamation and environment*, Vol. 9, pp. 203-208.

[3] Ferguson, T., S, (2009), "Linear programmin: a concise introduction,"

[4] Gershon, M., E, (1983), "Optimal mine production scheduling: evaluation of large scale mathematical programming approaches", *international journal of mining engineering*, Vol. 1, pp. 315-329.

[5] Ramazan, S. and Dimitrikapoulos, P., (2004), "Recent applications of operation research and efficient MIP formulations in open pit mining", *Society for mining, metallurgy and exploration*, Vol. 316, pp. 73-78.

8. Appendix

MATLAB code for the optimization problem

In order to run the codes:

1. Click on the above link, and open the folder "paper1".
2. The code for optimization of 18 blocks is the file opt18.m, in order to run this code:
   - Copy the opt18.m file to your MATLAB directory
   - Import the file blocks18 into MATLAB
   - Run the code

3. The code for optimization of 120 blocks is in the file new120.m, in order to run this code:
   - Copy the new120.m file to your MATLAB directory
   - Import the file block120
   - Run the code.

Optimizing block extraction sequence with MIP method and investigating the effect of road condition on truck cycle time

Yashar Pourrahimian[1]and Hooman Askari-Nasab

Abstract

Mine long-term plans define the complex strategy of displacement of ore, waste, overburden, and tailings over the mine life. The objectives of long-term mine plans are to manage and maximize the future cash flows and to minimize the environmental footprint of mining operations. Among the operation tasks, the haulage is the most expensive operation. We have used a mixed integer programming formulation to schedule the order of extraction of a mining bench in an iron ore mine. The objective function was to maximize the net present value of the operation, while meeting the qualitative and quantitative production constraints. Also, we have used simulation to investigate the affect of road conditions on truck cycle time.

1. Introduction

The operation tasks in open-pit mines consist of drilling, blasting, loading, haulage and general services. Among them, the haulage is the most expensive operation that occupies more than 50% of the total operation cost in open-pit mines. Therefore, minimizing the haulage cost can be one of the most critical factors in ore production. The shovel-truck haulage system is common in open-pit mines due to the flexibility of the fleet. The haulage cost of shovel-truck system is dependent on the productivity of an operating truck which can be represented by the average truck cycle time.

1.1. Introduction to linear programming (LP) and integer programming (IP)

A major element of mine planning is the optimization of long-term production schedule. The aim is to maximize the overall discounted net revenue from a mine within operational constraints such as mining slope, grade blending, ore production and mining capacity. Integer programming (IP) and linear programming (LP) mathematical models are considered to be powerful tools in optimizing mine schedules, and there have been major efforts in applying them to mining projects (Ramazan *et al.*, 2004).

Linear programming (LP): Linear programming uses a mathematical model to describe the problem of concern. The adjective linear means that all the mathematical functions in this model are required to be linear functions. Linear programming involves the planning of activities to obtain an optimal result, a result that reaches the specified goal best (according to the mathematical model) among all feasible alternatives(Hillier, 2005).

Integer programming (IP): In many practical problems, the decision variables actually make sense only if they have integer values. If requiring integer values is the only way in

[1] PhD student, School of Mining, University of Alberta

which a problem deviates from a linear programming formulation, then it is an integer programming (IP) problem. The mathematical model for integer programming is the linear programming model with the one additional restriction that the variables must have integer values. If only some of the variables are required to have integer values (so the divisibility assumption holds for the rest), this model is referred to as mixed integer programming (MIP). With just two choices, we can represent such decisions by decision variables that are restricted to just two values, say 0 and 1. Thus, the j^{th} yes-or-no decision would be represented by, say, x_j such that

$$x_j = \begin{cases} 1 & \text{If decision } j \text{ is yes.} \\ 0 & \text{If decision } j \text{ is no.} \end{cases}$$

Such variables are called binary variables (or 0–1 variables). Consequently, IP problems that contain only binary variables sometimes are called binary integer programming (BIP) problems (or 0–1 integer programming problems) (Hillier, 2005).

2. An abstract description of open pit mining

Ore bodies in open pit mines are represented by block models (figure 1). A block model divides ore and waste blocks adjacent to each other. The model may have several hundred thousand blocks depending on the size of the orebody and the size of the blocks. The size of the blocks is a function of the equipment used and the blasting pattern practiced. The average ore content of each block is calculated using geostatistical methods.

Open pit mining is a mineral extraction method by which the ore body is accessed by opening a large stretch of ground to expose the ore to air. Mining begins with a small pit in the surface, and then proceeds to a larger pit, which encloses the small pit, and the process continues until a final pit is reached (figure2). A mining sequence is obtained from a series of nested pits (figure3).

The time it will take to mine all the pits in the sequence will define the mine life, and the boundaries of the last pit in the sequence will determine the ultimate pit limits.

The blocks with average grade less than the cutoff grade are considered as waste blocks, and sometimes it is necessary to mine them in order to reach the ore blocks. The material mined from waste blocks is sent to the waste dump. On the other hand, some ore blocks may have to be left without mining because too many waste blocks must be mined to reach those blocks.

As seen in figure 3, mining in open pits starts from top and proceeds towards the bottom. Pit 1 is mined first, and subsequently Pit 2 is reached by mining the incremental blocks between Pit 1 and 2. In mining engineering terminology, it is said that Pit 2 is reached by a pushback from Pit 1. The most essential problem in long-term production planning is the determination of the set of pushbacks, that is, the mining sequence, to maximize the net present value (NPV) of the project. The results of the long term production planning are used as guides for the short- term production planning which may be for a quarter, a month or a week.

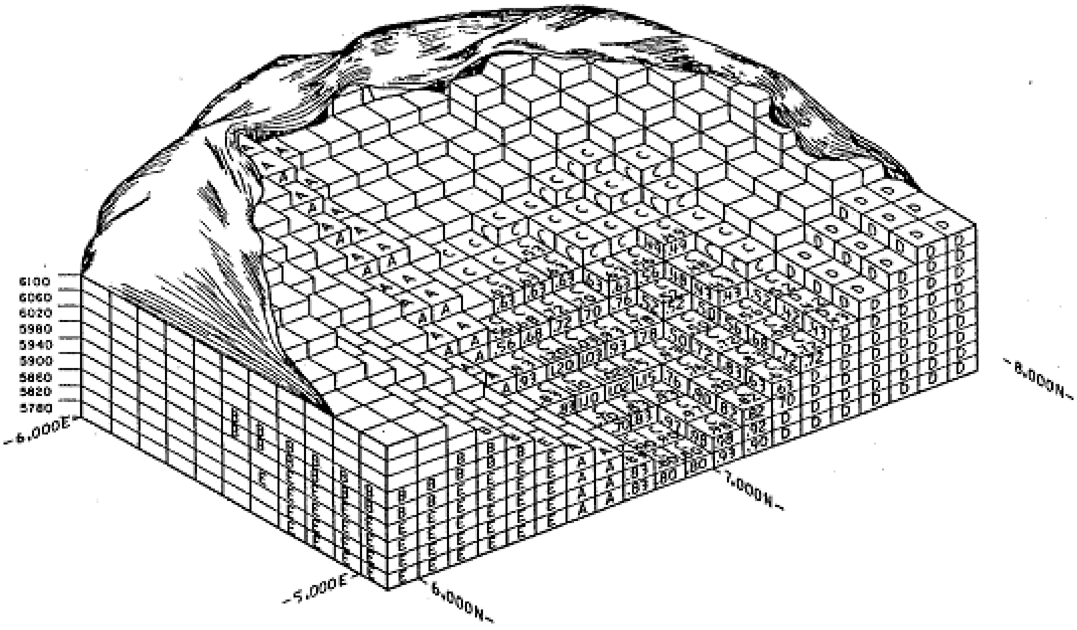


Fig 1. Diagrammatic view of a 3-D block matrix containing can orebody (Hustrulid *et al.*, 2006)

Fig 2. Sungun copper mine pit, Iran

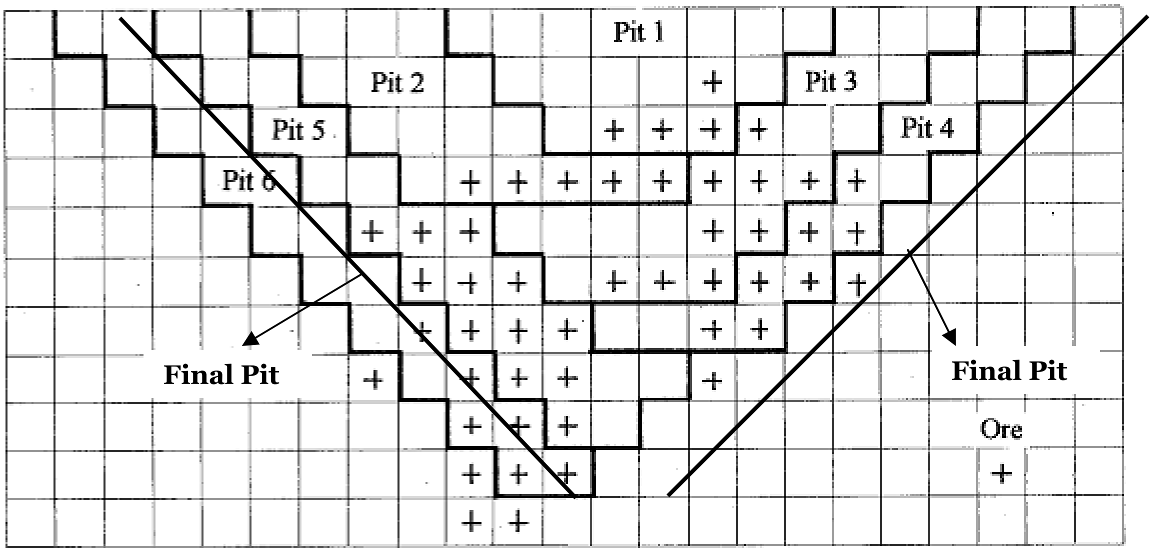


Fig 3. Two- dimensional view of nested pits and final pits
(Pits 1 to 6 are nested pits) (Sevim *et al.*, 1998)

3. MIP formulation for multi-period long-term production scheduling model

MIP models are generally used to maximize the overall discounted economic value, or net present value (NPV), of a mining project. Equation (1) was used as the objective function.

$$\text{Maximize } \sum_{t=1}^{T}\sum_{n=1}^{N}\frac{BEV_n^t}{(1+i)^t}\times X_n^t = \text{Minimize}[-\sum_{t=1}^{T}\sum_{n=1}^{N}\frac{BEV_n^t}{(1+i)^t}\times X_n^t] \qquad (1)$$

Where T is the maximum number of scheduling periods, N is the total number of blocks to be scheduled, BEV_n^t is block economic value of block n in period t, i is interest rate and X_n^t is a binary variable, equal to 1 if the block n is to be mined in period t, otherwise 0.

3.1. Grade blending constraints

The average grade of the material sent to the mill has to be less than or equal to a certain grade value, G_{max} , for each period, t (Ramazan *et al.*, 2004).

$$\sum_{n=1}^{N}(g_n - G_{\max})\times Ot_n \times X_n^t \le 0 \qquad (2)$$

Where g_n is the average grade of block n, and Ot_n is the ore tonnage in block n.

The average grade of the material sent to the mill has to be greater than or equal to a certain grade value, G_{min} , for each period, t.

$$\sum_{n=1}^{N}(g_n - G_{\min}) \times Ot_n \times X_n^t \geq 0 \tag{3}$$

3.2. Reserve constraints

Reserve constraints are constructed for each of the blocks to state that all the blocks in the model considered have to be mined once (Ramazan *et al.*, 2004).

$$\sum_{t=1}^{T} X_n^t = 1 \tag{4}$$

3.3. Processing capacity constraints

The total tonnage of ore processed cannot be more than the processing capacity $(PC_{\max})$ in any period, t (Ramazan *et al.*, 2004).

$$\sum_{n=1}^{N}(Ot_n \times X_n^t) \leq PC_{\max} \tag{5}$$

The total tonnage of ore processed cannot be less than a certain amount $(PC_{\min})$ in any period, t.

$$\sum_{n=1}^{N}(Ot_n \times X_n^t) \geq PC_{\min} \tag{6}$$

3.4. Mining capacity

The total amount of material (waste and ore) to be mined cannot be more than the total available equipment capacity $(MC_{\max})$ for each period, t (Ramazan *et al.*, 2004).

$$\sum_{n=1}^{N}(Ot_n + W_n) \times X_n^t \leq MC_{\max} \tag{7}$$

Where W_n is the tonnage of waste material in block n .

To force the MIP model to produce balanced waste production throughout the periods, a lower bound $(MC_{\min})$ may need to be implemented as follows:

$$\sum_{n=1}^{N}(Ot_n + W_n)\times X_n^t \geq MC_{\min} \tag{8}$$

The number of binary variables required for the MIP model is equal to the number of blocks multiplied by the total periods to be scheduled, as can be seen in the formulations above.

4. Illustration of an example

In this section, a scheduling optimization example with 415 blocks has been explained. All necessary steps for problem setting have been explained in detail.

The goal is extraction of all 415 blocks in five periods while maximizing the net present value as represented by equation (1).

Step 1. Analysis of blocks information

Block model is created with GEMS6.1 and Whittle4.1 software. For this suppose the drill- holes data are used as input of Gems then the block model is created. After that, information is saved in Matlab file format.

These 415 blocks are a segment of an orebody block model of a real iron ore deposit. Information of each block is as follows:

1x415 struct array with fields:

XI, YI, ZI:	Index of each block in the block model
X,Y, Z:	Coordinates of the block
MCAF:	Mining cost adjustment factor
PCAF:	Processing cost adjustment factor
gradeS, gradeP, gradeMWT:	Grade of Sulphur, Phosphor and Fe
OreTonnes:	Total tonnage of ore in the block
WasteTonnes :	Total tonnage of waste in the block
OreValue:	Value of ore in the block
WasteCosts:	Costs of waste in the block
BlockTonnage:	Total tonnage of the block(waste + ore)
EBV:	Economic block value
P, S, MWT:	Tonnage of Phosphor, Sulphur and Fe in the block

All blocks were analyzed and necessary graphs and statistic analysis carried out using Matlab. Figure 4 shows grade distribution histogram. According to this histogram, maximum value of grade $(G_{\max})$ is equal to 0.8 percent.

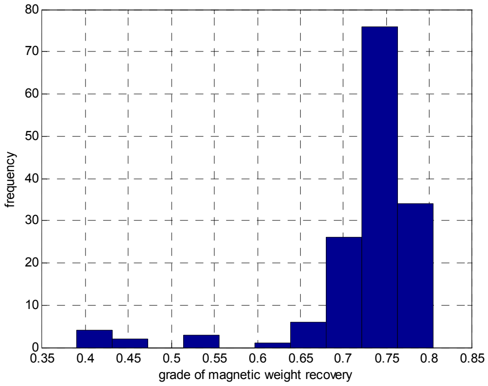


Fig 4. Grade histogram for ore blocks

Step 2. Setting up the problem for solving with Matlab

The Matlab **bintprog** function solves binary integer programming problems of the form:

Minimize $f^{/}.x$ such that $\begin{array}{l} A.x \leq b \\ Aeq.x = beq \end{array}$

Where f, b, and beq are vectors, A and Aeq are matrices, and the solution x is required to be a binary integer vector—that is, its values can only take on the values 0 or 1.

To solve this problem in Matlab we have to use following function:

x = bintprog(f,A,b, Aeq,beq)

To set the problem in format of Matlab bintprog function the following steps have to be followed:

Step 3. Creating matrix f

The function that we want to maximize with interest rate equal 15% is as follows:

$$\sum_{t=1}^{T}\sum_{n=1}^{N}\frac{BEV_n^t}{(1+i)^t}\times X_n^t \Rightarrow \sum_{t=1}^{5}\sum_{n=1}^{415}\frac{BEV_n^t}{(1+i)^t}\times X_n^t$$

The expanded formats of function for each period are as follows:

Function for the first period: $(\frac{BEV_1^1}{(1+0.15)^1}x_1^1 + \frac{BEV_2^1}{(1+0.15)^1}x_2^1 + ... + \frac{BEV_{415}^1}{(1+0.15)^1}x_{415}^1)$

Function for the second period: $(\frac{BEV_1^2}{(1+0.15)^2}x_1^2 + \frac{BEV_2^2}{(1+0.15)^2}x_2^2 + ... + \frac{BEV_{415}^2}{(1+0.15)^2}x_{415}^2)$

Function for the third period: $(\frac{BEV_1^3}{(1+0.15)^3}x_1^3+\frac{BEV_2^3}{(1+0.15)^3}x_2^3+...+\frac{BEV_{415}^3}{(1+0.15)^3}x_{415}^3)$

Function for the fourth period: $(\frac{BEV_1^4}{(1+0.15)^4}x_1^4+\frac{BEV_2^4}{(1+0.15)^4}x_2^4+...+\frac{BEV_{415}^4}{(1+0.15)^4}x_{415}^4)$

Function for the fifth period: $(\frac{BEV_1^5}{(1+0.15)^5}x_1^5+\frac{BEV_2^5}{(1+0.15)^5}x_2^5+...+\frac{BEV_{415}^5}{(1+0.15)^5}x_{415}^5)$

In this case, we want to maximize the objective function; therefore, we have to multiply the function by a negative sign.

Where, f is a column matrix with N×T elements that N is number of blocks and T is number of periods.

$$f=\begin{bmatrix}-\frac{BEV_1^1}{(1.15)^1} & . & . & . & -\frac{BEV_{415}^1}{(1.15)^1} & -\frac{BEV_1^2}{(1.15)^2} & . & . & . & -\frac{BEV_{415}^2}{(1.15)^2} & . & . & . & . & -\frac{BEV_{415}^5}{(1.15)^5}\end{bmatrix}^T$$

Step 4. *Creating* [A],{x}, {b}, [Aeq] and {beq}

4.1. Creating coefficients matrix of the linear inequality constraints [A]

We want to maximize f subject to $[A].\{x\}\le\{b\}$, where [ ] represents a matrix and { } represents a vector. A is the matrix containing the coefficients of the linear inequality constraints, b is the vector corresponding to the right-hand side of the linear inequality constraints and x is solution.

Matrix A has m rows and n columns that m and n given by m=NC×T and n=N×T. Where, NC is number of inequality constraints, T is number of periods that blocks are extracted in and N is number of blocks.

Matrix A is created by following inequality constraint equations:

Constraint 1(A1): $\sum_{n=1}^{N}(g_n-G_{\max})\times Ot_n\times X_n^t\le 0$

Constraint 2(A2): $\sum_{n=1}^{N}(g_n-G_{\min})\times Ot_n\times X_n^t\ge 0$

Constraint 3(A3): $\sum_{n=1}^{N}(Ot_n\times X_n^t)\le PC_{\max}$

Constraint 4(A4): $\sum_{n=1}^{N}(Ot_n\times X_n^t)\ge PC_{\min}$

Constraint 5(A5): $\sum_{n=1}^{N}(Ot_n+W_n)\times X_n^t\le MC_{\max}$

Constraint 6(A6): $\sum_{n=1}^{N}(Ot_n + W_n) \times X_n^t \geq MC_{\min}$

Since, constraints must be written in $[A].\{x\} \leq \{b\}$ form, conditions 2, 4, and 6 are written as follows:

$$\sum_{n=1}^{N}(G_{\min} - g_n) \times Ot_n \times X_n^t \leq 0$$

$$\sum_{n=1}^{N} - (Ot_n \times X_n^t) \leq -PC_{\min}$$

$$\sum_{n=1}^{N} - (Ot_n + W_n) \times X_n^t \leq -MC_{\min}$$

Table 1 shows summary of inequality constraints and short description about them.

Table 1. constructing matrix A

$(g_n - G_{\max}) \times Ot_n$	This constraint is exerted from first row to Tth row. First, value of this equation is calculated for each block, and then, they are replaced in row one period 1, row 2 period 2 and so to Tth row.
$(G_{\min} - g_n) \times Ot_n$	This constraint is exerted from (T+1)th row to (2T) th row. Value of this equation is calculated for each block, and then, is replaced in row (T+1) period 1, row (T+2) period 2 and so to (2T) th row.
Ot_n	Ore tonnage of each block is replaced from (2T+1) th row to (3T) th row into the related period.
$-Ot_n$	Ore tonnage of each block is multiplied by (-1) and then is replaced from (3T+1) th row to (4T) th row into the related period.
$(Ot_n + W_n)$	Total tonnage of each block (ore+waste) is replaced from (4T+1) th row to (5T) th row into the related period.
$-(Ot_n + W_n)$	Total tonnage of each block (ore+waste) is multiplied by (-1) and then is replaced from (5T+1) th row to (6T) th row into the related period.

When the values of equation are replaced in related period of each row, other elements in those rows have to be zero. Figure 5 shows how matrix A is created. AC_n indicates value of constraint C for block n. For instance, A1$_1$ is the value of constraint 1 for block 1 and A2$_{411}$ is the value of constraint 2 for block 411. The first number is number of constraints and second one number of blocks.

4.1.1 Creating vector x

Vector x with elements $x_j \{j = 1,2,3,...,2075\}$ is the solution of the problem. This vector has (T×N) elements. The following relationship governs among elements of vector x.

$$\left.\begin{array}{l}
\text{Period 1 : } x_1, x_2, ..., x_N \\
\text{Period 2 : } x_{N+1}, x_{N+2}, ..., x_{2N} \\
\text{Period 3 : } x_{2N+1}, x_{2N+2}, ..., x_{3N} \\
. \\
. \\
. \\
\text{Period T: } x_{[(T-1)\times N]+1}, x_{(T-1)\times N]+2}, ..., x_{T\times N}
\end{array}\right\}
\quad
\begin{array}{l}
x_1 = x_{N+1} = ... = x_{[(T-1)\times N]+1} \\
x_2 = x_{N+2} = ... = x_{[(T-1)\times N]+2} \\
. \\
. \\
. \\
x_N = x_{2N} = ... = x_{TN}
\end{array}$$

$$\begin{array}{ccccccccccccccccccccccccc|c}
\multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad}^{\begin{matrix}\textbf{Period 1}\\ \textbf{1 - 415}\end{matrix}}} & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad}^{\begin{matrix}\textbf{Period 2}\\ \textbf{416 - 830}\end{matrix}}} & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad}^{\begin{matrix}\textbf{Period 3}\\ \textbf{831 - 1245}\end{matrix}}} & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad}^{\begin{matrix}\textbf{Period 4}\\ \textbf{1246 - 1660}\end{matrix}}} & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad}^{\begin{matrix}\textbf{Period 5}\\ \textbf{1661 - 2075}\end{matrix}}} & \\
A1_1 & . & . & . & A1_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & A1_1 & . & . & . & A1_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A1_1 & . & . & . & A1_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textbf{Period 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A1_1 & . & . & . & A1_{415} & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A1_1 & . & . & . & A1_{415} & \\
\hdashline
A2_1 & . & . & . & A2_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & A2_1 & . & . & . & A2_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A2_1 & . & . & . & A2_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textbf{Period 2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A2_1 & . & . & . & A2_{415} & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A2_1 & . & . & . & A2_{415} & \\
\hdashline
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A6_1 & . & . & . & A6_{415} & {}_{30\times 2075}
\end{array}$$

Dimension of matrix $A = (NC\times T)\times(N\times T) = (6\times 5)\times(415\times 5) = 30\times 2075$

Fig 5. Method of coefficient matrix creation

4.1.2 Creating vector b

b is a vector containing the constants of the linear inequality constraints. Dimension of this vector depends on dimensions of matrix A and vector x.

$$[A]_{(NC\times T)\times(N\times T)}\cdot\{x\}_{(N\times T)\times 1}=\{b\}_{(NC\times T)\times 1}$$

In this case dimension of b is equal to 30×1.

4.1.3 Coefficients matrix of the linear equality constraints [Aeq] and vector{beq}

In addition to linear inequality constraints, there are some equality constraints that we cannot replace them into the matrix A. In this case, we know each block can be mined once during the mine life.

Condition 7(A7): $\sum_{t=1}^{T} X_n^t = 1$

$x_1 + x_{N+1} + ... + x_{[(T-1)\times N]+1} = 1$

$x_2 + x_{N+2} + ... + x_{[(T-1)\times N]+2} = 1$

.

$x_N + x_{2N} + ... + x_{TN} = 1$

To create matrix Aeq the related coefficient of each block have to be replaced by 1. Matrix Aeq is a N×(T×N) matrix. beq is the vector corresponding to the right-hand side of the linear equality constraints and x is solution.

In this case at the following matrix the first row indicates the coefficients of block number 1 during the scheduling periods (five periods). x_1, x_{416}, x_{831} , x_{1246} and x_{1661} are solution of problem for block number 1. If x_1=1 thus block 1 is extracted in period 1 and x_{416}=x_{831}= x_{1246}=x_{1661}=0, if x_{416}=1 thus block 1 is extracted in period 2 and x_1=x_{831}= x_{1246}=x_{1661}=0, and etc. Figure 6 shows how matrix Aeq is created. Therefore, the following condition must be governed because each block can be mined once. For instance, for block number one we can write:

$$1\times x_1 + 1\times x_{416} + 1\times x_{831} + 1\times x_{1246} + 1\times x_{1661} = 1$$

After creating all necessary matrices and vector this problem was solved using Matlab and Tomlab toolbox and the answer was as follows:

bintprog (CPLEX): bintprog converged to a solution X.

Table 2 shows the summary of important information for each period. Tonnages of ore and waste that are extracted in each period are shown in figure 7.

Figures 8 and figure 9 show the average grade and value of function for each period, respectively.

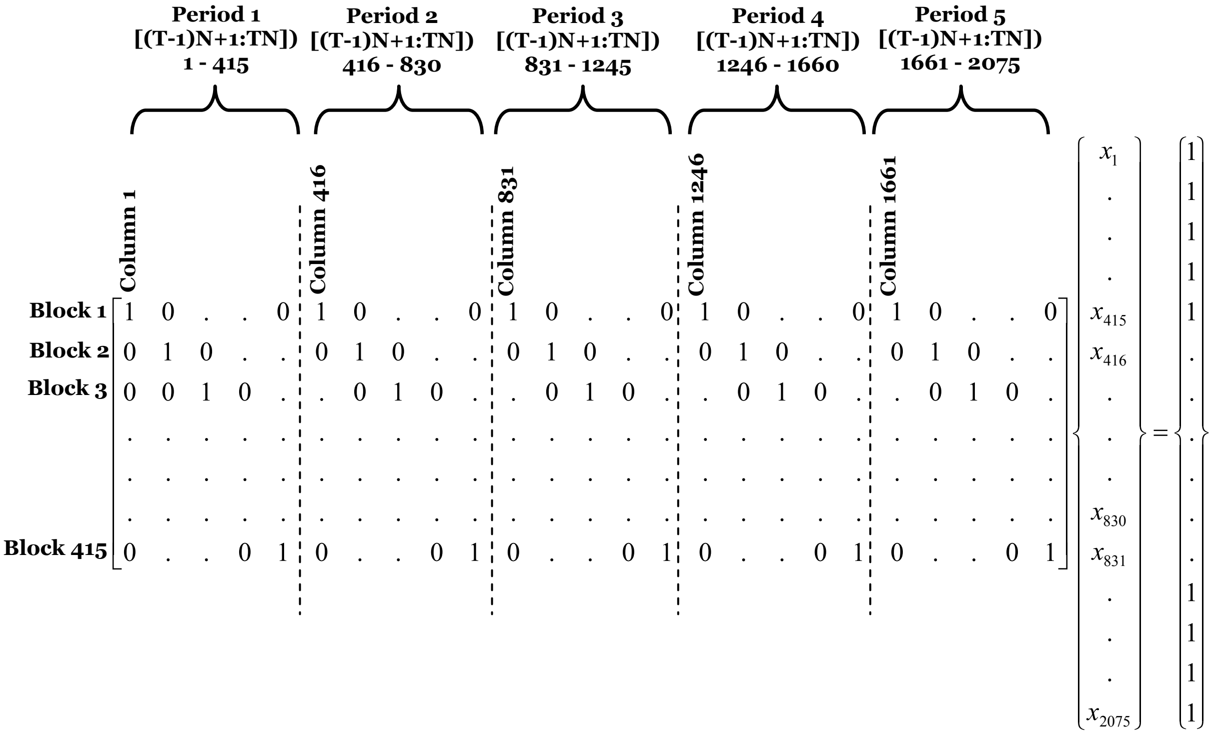


Fig 6. Method of Aeq matrix creation

Table 2. Summary of obtained answers for each period.

	Period 1	Period 2	Period 3	Period 4	Period 5
Waste (t)	1660332	2484400	2498464	2501276	3750976
Ore (t)	2496000	2496000	2496000	2496000	1244066
Waste+Ore (t)	4156332	4980400	4994464	4997276	4995042
$f = \sum \frac{BEV_n}{(1+i)^t}$	1.4325×10^8	1.2010×10^8	1.0121×10^8	0.8538×10^8	0.2673×10^8
$\sum_{t=1}^{5}\sum_{n=1}^{415} \frac{BEV_n^t}{(1+i)^t}$			4.7668×10^8		

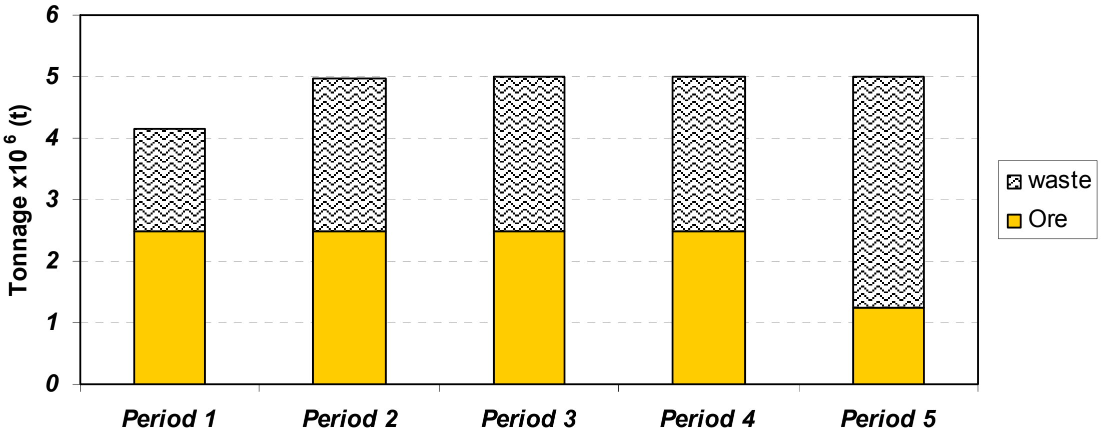


Fig 7. Amount of ore and waste in each period

To calculate average grade of each period equation (9) was used:

$$g_{ave}^{t} = \frac{\sum (g_n^t \times O_n^t)}{\sum O_n^t} \qquad (9)$$

Where, g_{ave}^{t} is average grade of period t, g_n^t grade of block number n that is extracted in period t and O_n^t ore tonnage of block number n that is extracted in period t.

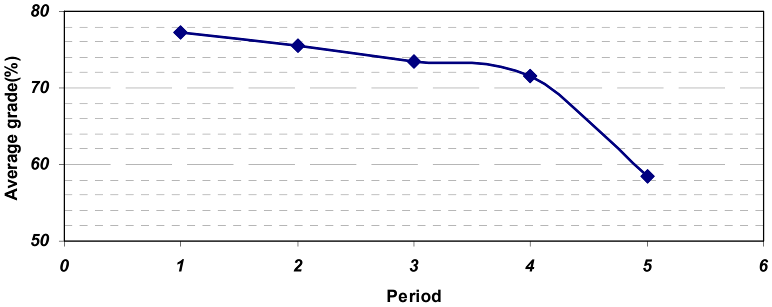


Fig 8. Average grade of Fe% in each period

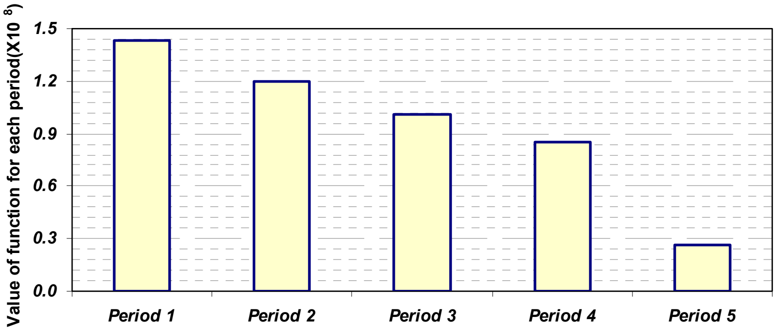


Fig 9. Annual discounted cash flows

5. Problem definition of the transportation problem

Figure 10 shows plan view of the working bench and roads that have been used for simulation. Distance between working bench as starting point and destinations are divided into two parts. The first part is from each block to the starting point of main road and the second part is from the starting point of main road to destinations. The main road is 7 km and the coordinates of the starting point of the main road are x=96000, y=600740. Distance between each block and destination is given by:

$$D(m) = 7000 + d_{B-S} \tag{10}$$

Where, D is the total distance and d_{B-S} is the distance between each block and the main road starting point.

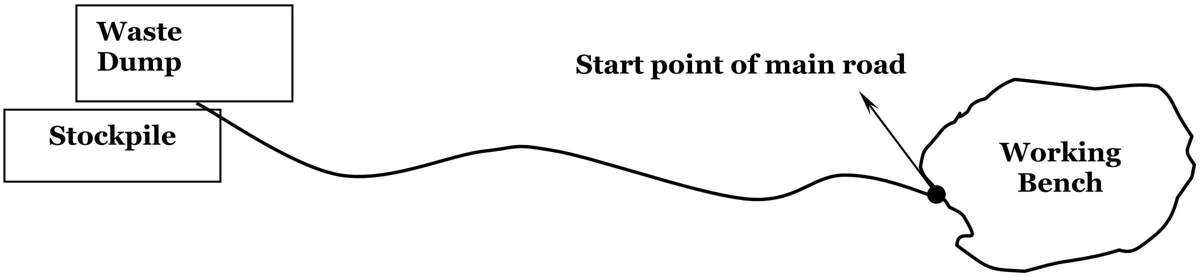


Fig 10. Plan view of working area

6. Simulation of transportation system with Simulink

Average truck cycle time is given by equation (11)

$$T_{Truck} = ST_{loading} + L_t + T_{t-fill} + ST_{unloading} + T_{unloading} + T_{t-empty} + AD_t \tag{11}$$

Where, T_{Truck} is the truck cycle time, $ST_{loading}$ is the spot time at loading, L_t is the loading time, T_{t-fill} is the travel time to dump or stockpile (fill), $ST_{unloading}$ is the spot time at unloading , $T_{unloading}$ is the unloading time, $T_{t-empty}$ is the travel time moving to working bench (empty) and AD_t is the average delay time including both waits and delays.

To estimate loading time we can use equation (12) :

$$L_t = \frac{t_{capacity} \times SH_{t-cyc}}{SH_{capacity}} \tag{12}$$

Where, L_t is the loading time of truck, $t_{capacity}$ is the truck capacity, SH_{t-cyc} is the shovel cycle time and $SH_{capacity}$ is the shovel capacity.

In this case, we can use equations (13) and (14) as constitutive relationships for the main road:

$$T_{t-fill} = \frac{x}{V_{t-fill}} \tag{13}$$

$$T_{t-empty} = \frac{x}{V_{t-empty}} \tag{14}$$

Where, x is the distance, V_{t-fill} is the velocity of truck after loading and $V_{t-empty}$ is the velocity of truck after unloading. The assumption is that in the working area the velocity of truck because of extraction is 85% less than velocity on the main road, therefore equations (13) and (14) can be rewritten as follows:

$$T_{t-fill} = \frac{x}{V_{t-fill}} + \frac{x_w}{0.15 \times V_{t-fill}} = \frac{0.15x + x_w}{0.15 \times V_{t-fill}}$$

$$T_{t-empty} = \frac{x}{V_{t-empty}} + \frac{x_w}{0.15 \times V_{t-empty}} = \frac{0.15x + x_w}{0.15 \times V_{t-empty}}$$

$$T_{t-fill} + T_{t-empty} = \frac{0.15x + x_w}{0.15 \times V_{t-fill}} + \frac{0.15x + x_w}{0.15 \times V_{t-empty}} = \frac{V_{t-empty}(0.15x + x_w) + V_{t-fill}(0.15x + x_w)}{0.15 \times V_{t-fill} \times V_{t-empty}}$$

$$= \frac{(V_{t-empty} + V_{t-fill})(0.15x + x_w)}{0.15 \times V_{t-fill} \times V_{t-empty}}$$

Where, x_w is the distance between each block and starting point of main road. Using the constitutive relationships we can rewrite equation (11) as follows:

$$T_{Truck} = ST_{loading} + \frac{t_{capacity} \times SH_{t-cyc}}{SH_{capacity}} + \frac{(V_{t-empty} + V_{t-fill})(0.15x + x_w)}{0.15 \times V_{t-fill} \times V_{t-empty}} + ST_{unloading} + T_{unloading} + AD_t$$

A Simulink simulation model was built to capture the truck trave cycle time based on the above equation. Figure 11 shows the model.

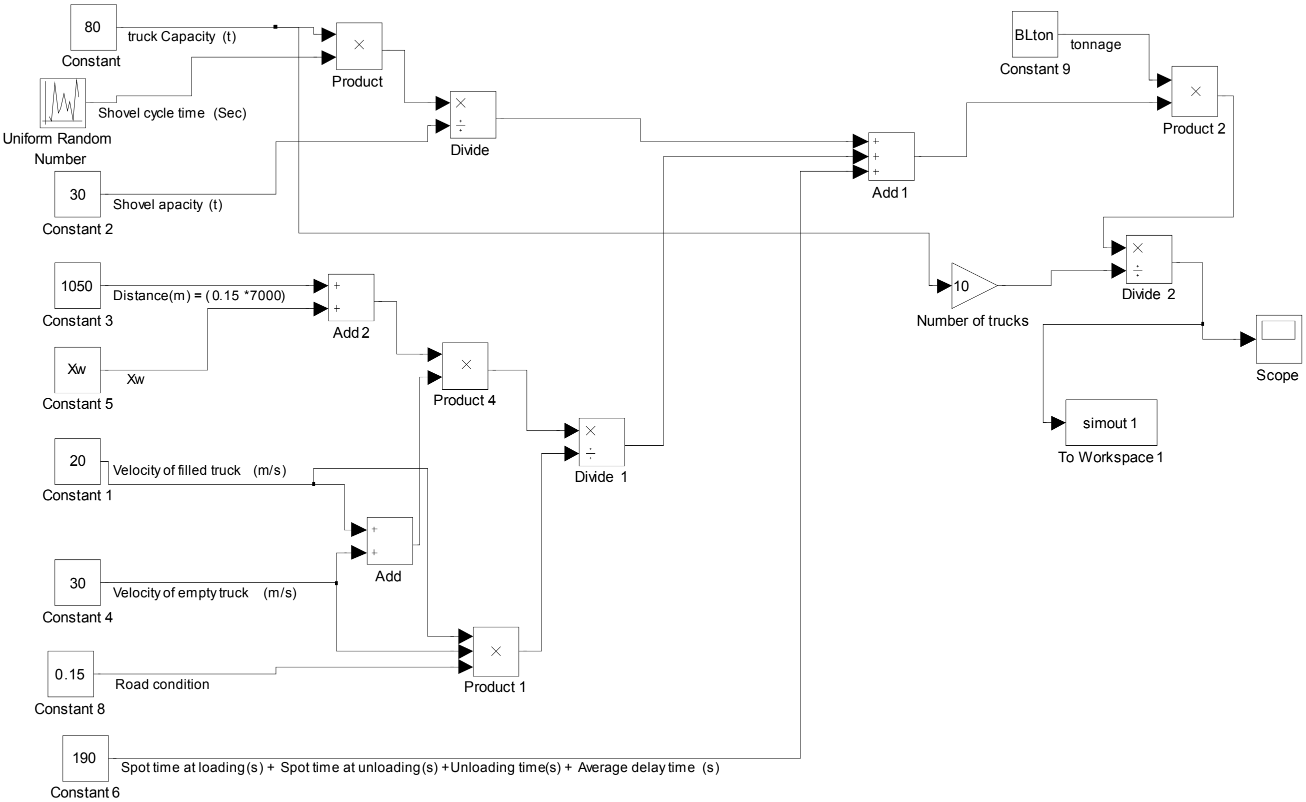


Fig 11. Simulink model for sensitivity analysis

Simulation was carried out for three different road conditions. Table 3 indicates truck velocity in fill and empty situation for each condition.

We assumed that there were 10 available trucks in the mine. Distance between each block and starting point of main road is calculated using Matlab code and is inserted as a matrix from X_w entrance. Tonnage of each block that has been calculated using Matlab code is inserted from BLton entrance.

Table 3. Velocity of truck for different conditions

Condition	V_{t-fill} (m/s)	$V_{t-empty}$ (m/s)
1	20	30
2	14	25
3	10	18

There are two working shifts, day shift and night shift, which each shift is 8 hours and working days during a year is 350 days.

Figure 12 shows the necessary working days for blocks in each period under different conditions. Sum of the necessary days for each condition per period have been summarized in table 4.

Table 4. Duration of each period and project for different conditions

	Period 1 (day)	Period 2 (day)	Period 3 (day)	Period 4 (day)	Period 5 (day)	Duration of project (day)
Velocity condition1	142	172	170	171	172	826
Velocity condition2	181	219	216	218	219	1054
Velocity condition3	242	294	290	292	294	1411

The results show duration of project for conditions 1, 2 and 3 will be more than 2, 3 and 4 years, respectively. We want to find the effect of maintenance of road in working area on duration of project. For this purpose, the mine maintenance group increased the quality of ground for transportation in working area to 50% of main road quality using two dozers.

Necessary days for extraction of blocks of each period under different conditions after maintenance are shown in figure 13. Table 5 shows summary of project duration when quality of ground for transportation was increased in working area.

7. Conclusion

In the present project, I tried to explain how we could use mixed integer formulation for open pit mines production scheduling using Matlab. After finding the blocks extraction sequence, effect of road maintenance on duration of project has been considered. For this purpose, the truck velocity was considered as a function of road conditions.

The results show that deterioration of the road conditions has a significant affect on project duration. The project time increased from two years under velocity condition one to four years under velocity condition two.

In mines, most of the road problems are within the working areas where the extraction and blasting are taken place. Therefore, two different kinds of road conditions in working areas were considered. The results indicate when the quality of road in working area increases because of good maintenance, project duration shows a decrease of 32%.

On the other hand, many mines are paying more attention to tire care programs. For the surface mining business, tires represent the largest portion of a haul truck's hourly operating expenses. Good haul roads lead to reduced fuel consumption, higher vehicle speed, longer tire life, and more comfortable and safer riding.

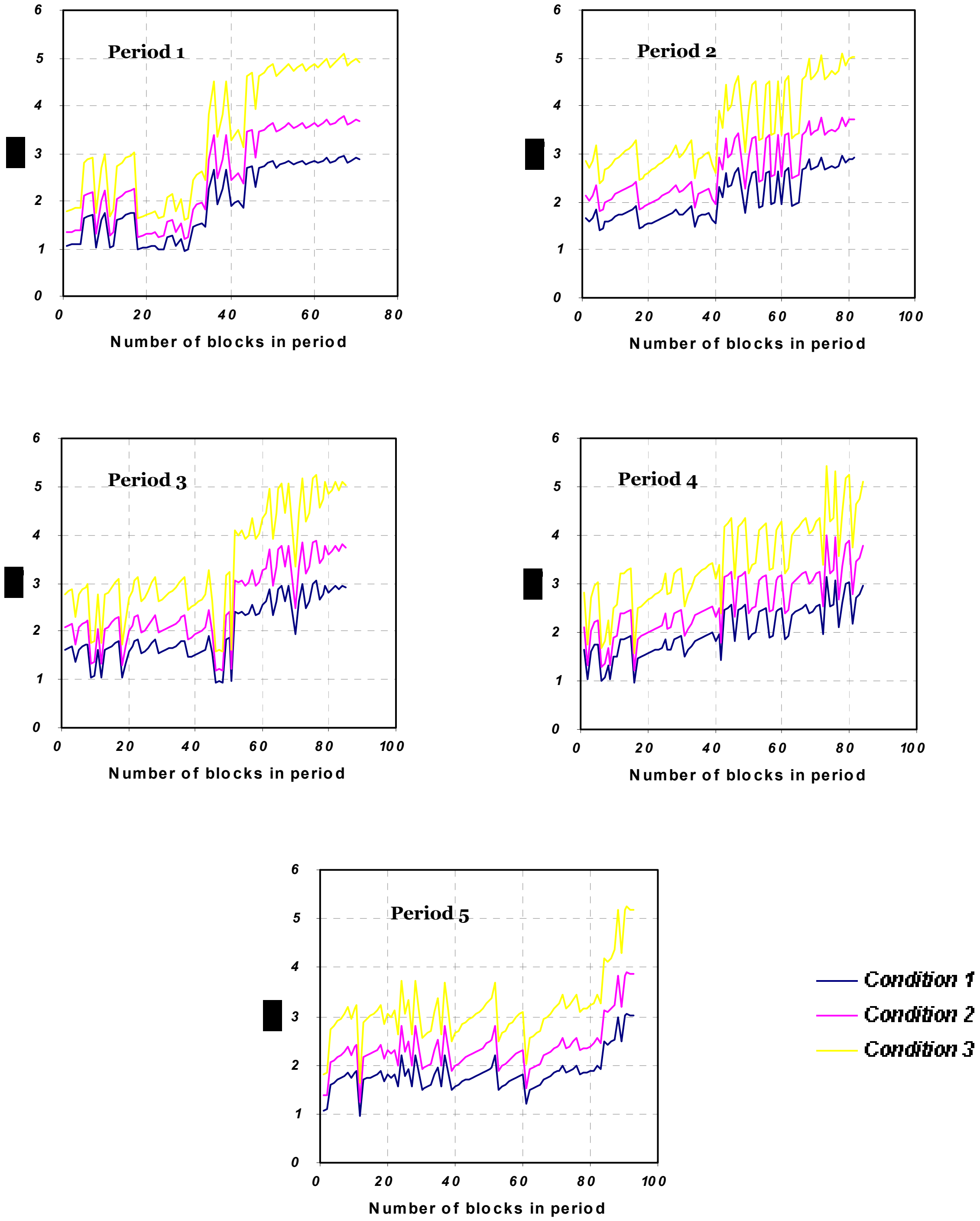


Fig 12. Necessary days for transportation of each block from working bench to destinations per period under different conditions.

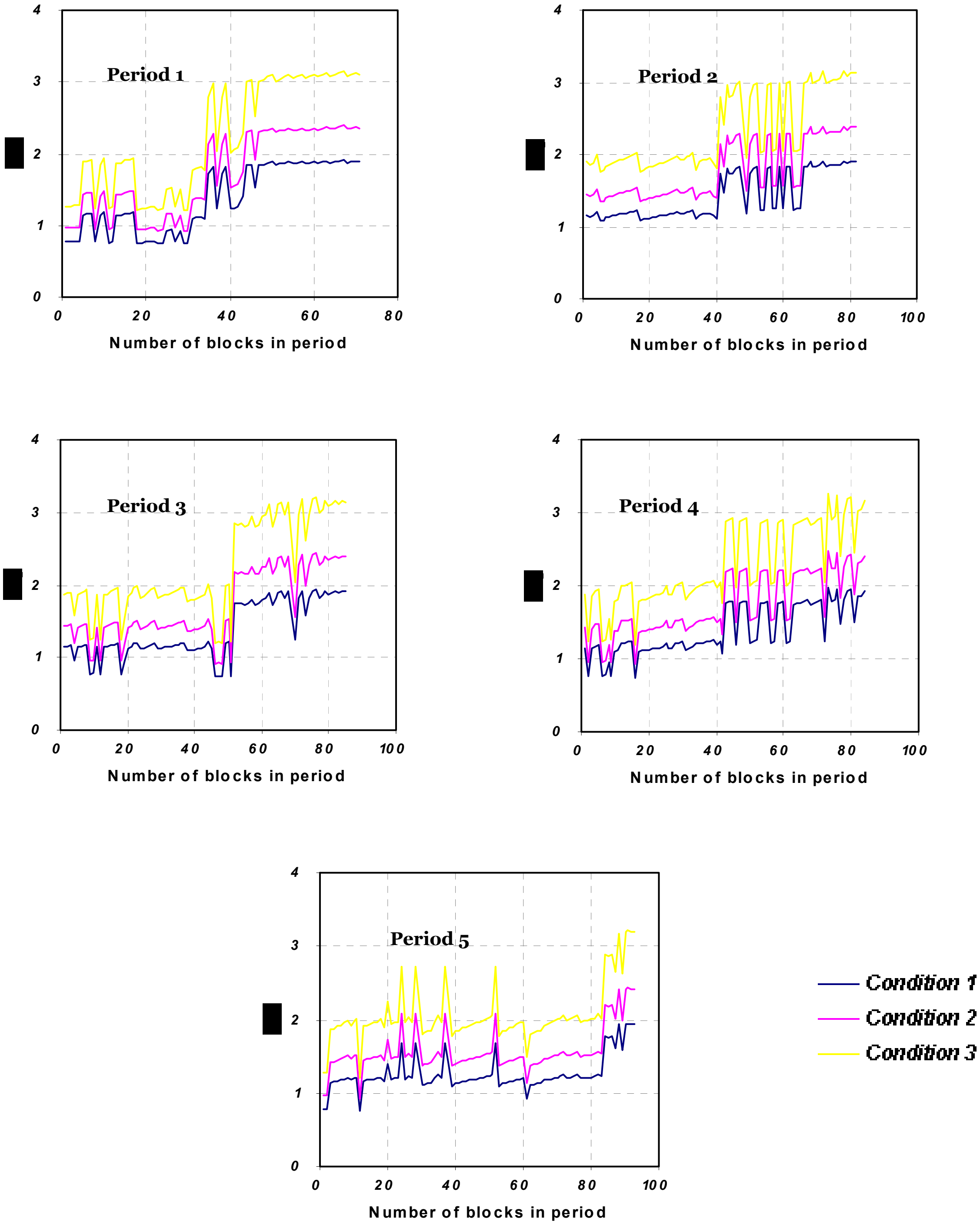


Fig 13. Necessary days for transportation of each block from working bench to destinations per period under different conditions when quality of ground in working area increases to 50% of main road quality.

Table 5. Duration of each period and project for different conditions

	period1 (day)	period2 (day)	period3 (day)	period4 (day)	period5 (day)	duration of project (day)
Velocity condition1	97	117	117	117	118	566
Velocity condition2	122	147	146	146	147	708
Velocity condition3	159	192	191	192	193	927

Difference between duration of project for two different road conditions is illustrated in figure 14.

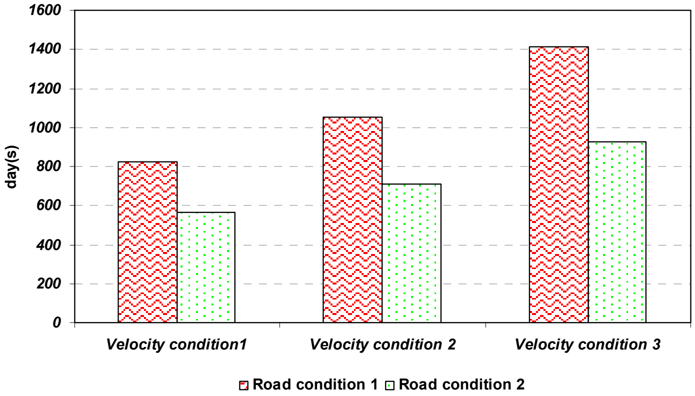


Fig 14. Duration of project for two different road conditions under different velocity conditions.

8. References

[1] Hillier, F. S., (2005), "Introduction to operation research", © McGraw-Hill, Pages 1061.

[2] Hustrulid, W. and Kuchta, M., (2006), "Open pit mine planning and design", © Taylor & Francis/BALKEMA, Pages 735.

[3] Ramazan, S. and Dimitrakopoulos, R., (2004), "Recent applications of operations research and efficient MIP formulation in open pit mining", *Transactions of the Society for Mining,Metallurgy and Exploration*, Vol. 316, pp. 73-78.

[4] Ramazan, S. and Dimitrakopoulos, R., (2004), "Traditional and new MIP models for production scheduling with in-situ grade variability", *International Journal of Surface Mining, Reclamation and Environment*, Vol. 18, pp. 85-98.

[5] Sevim, H. and Lei, D., (1998), "The problem of production planning in open pit mines", *INFOR*, Vol. 36, 1/2, pp. 1-12.

9. Appendix

Link to: MATLAB code of MIP model

To run above Matlab code you need Matlab software and Tomlab toolbox. Then to run follow the below steps:

1. Copy three files: Blocks.m, MIP_solver_415.m and viewmatrix.m in a same folder
2. Run Matlab
3. Select c:\tomlab as current directory
4. Inside Matlab type startup and press enter (>> startup)
5. Change current directory to the folder that you copied three files inside it in step 1
6. Inside Matlab load Blocks.m >> load Blocks
7. Open MIP_solver_415.m and then run it.

A review of open pit mine production scheduling and a guide for using Excel Solver in modeling MIP problems

Yashar Pourrahimian[1] *and Hooman Askari-Nasab*

Abstract

Optimization of production scheduling is important for managing the substantial cash flows inherent in open pit mining ventures. Mixed integer programming (MIP) models are used for production scheduling of open pit mines. In this method, formulations are based on binary variables for mining blocks. In this paper the uncertainty-based methods to optimizing open pit mine design and scheduling have been reviewed and then integer programming (IP) and mixed integer programming (MIP) have been discussed. Finally, a simple MIP model has been solved with Excel Solver and the results have been presented.

1. Introduction

Optimization and mathematical formulation have been exploited to solve long-term production scheduling problems since the 1960s (Ramazan *et al.*, 2003). There are two mathematical optimization approaches to solve these kinds of problem: deterministic and uncertainty based approaches. The uncertainty-based approach to optimizing open pit mine design and scheduling was developed in the 1990s (Golamnejad *et al.*, 2006).

Ravenscroft discussed risk analysis in mine production scheduling (Golamnejad *et al.*, 2006). This method can show only the impact of grade uncertainty on production scheduling using the alternative scenarios of the orebody, which are provided by conditional simulation. Dowd proposed a framework for risk assessment in open pit mines. He also considered some other random variables such as commodity price, mining costs, processing costs, etc (Dowd, 1994). Denby and Schofield proposed an algorithm, which considers ore grade variance in open pit design and scheduling using the Genetic algorithm. They used a multi-objective optimization method: maximizing value and minimizing risk (Denby *et al.*, 1995).

Dimitrakopoulos and Ramazan discussed a linear programming (LP) approach that considered grade uncertainty, equipment access and mobility constraints. This formulation was based on expected ore block grades and the probabilities of different element grades being above required cut-offs, both derived from simulated orebody models. Gody and Dimitrakopoulos presented an algorithm, which addresses the generation of an optimal condition under uncertainty. First, they generated production schedules on each simulated orebody and then they combined the mining sequences to produce a single schedule that

[1] PhD student, School of Mining, University of Alberta

minimizes the chance of deviating from the production target. This was done using the Simulated Annealing Meta-Heuristic method (Godoy *et al.*, 2004).

Ramazan (Ramazan *et al.*, 2004) suggested an MIP model that accommodates grade uncertainty. In this method, after obtaining simulated orebody models, scheduling patterns on each model is generated using traditional MIP formulation (with the objective of NPV maximization). Then the excavation probability of each block in a given time period is calculated. The block with probability between zero and one, are considered in a new optimization model.

Caccetta and Hill solved a binary integer programming (BIP) formulation with an efficient branching scheme and briefly outlined some aspects of a specialized branch-and-cut algorithm (Caccetta *et al.*, 2003). Golamnejad (Golamnejad *et al.*, 2006) presented a chance constrained binary integer programming model. The model integrates ore grade uncertainty explicitly and economic and mining considerations to generate an optimal life of mine production schedule to meet the required targets with a high level of confidence and low risk.

Boland (Boland *et al.*, 2006) extend the BIP formulation of Caccetta and Hill to include more than two ore attributes and develop knapsack cover inequalities that significantly decrease the computational requirements to obtain the optimal integer solution.

Fricke (Fricke, 2006; Boland *et al.*, 2009) described an MIP formulation, using ideas from the formulation of Menabde et al., which is a generalisation of the BIP formulation of Caccetta and Hill and Boland. The computational experiments conducted by Fricke show that this generalization is particularly effective.

Mixed integer programming (MIP) has become a common approach for optimizing production schedules of open pit mines. However, MIP has been found to be limited by: (a) feasibility in generating optimal solutions with practical mining schedules; and (b) inability to deal with in-situ variability of orebodies. In long-term production scheduling of open pit mines, MIP models are usually constructed to maximize the overall net present value (NPV) of the mining project (Ramazan *et al.*, 2004).

A key limitation in past MIP models has been difficulty in solving large problems, as these require a substantial number of binary variables. The attempted LP-based models often generate fractional mining of blocks, leading to the design becoming infeasible and/or non-optimal. Ramazan (Ramazan *et al.*, 2004) proposed a new method based on the fundamental tree concept, and substantially decreases the number of binary variables in MIP formulations for long-term production scheduling. Ramazan(Ramazan *et al.*, 2004) presented alternative methodologies useful in reducing the number of binary variables in MIP models. Although the above methods significantly decrease the number of binary variables required and enhance the application of MIP in large mineral deposits, in-situ orebody variability is not considered and all inputs are considered without uncertainty.

2. Theoretical framework and models

2.1. Introduction to LP and IP programming

A major element of mine planning is the optimization of production scheduling. The aim is to maximize the overall discounted economic value of a mine within operational constraints such as mining slope, grade blending, ore production and mining capacity. Integer programming (IP) and linear programming (LP) mathematical models are considered to be powerful tools in optimizing mine scheduling, and there have been major efforts in applying them to mining projects (Ramazan *et al.*, 2004).

2.1.1 Linear programming (LP)

Linear programming uses a mathematical model to describe the problem of concern. The adjective linear means that all the mathematical functions in this model are required to be linear functions. The word programming does not refer here to computer programming; rather, it is essentially a synonym for planning. Thus, linear programming involves the planning of activities to obtain an optimal result, a result that reaches the specified goal best (according to the mathematical model) among all feasible alternatives (Hillier, 2005).

2.1.2 Integer programming (IP)

In many practical problems, the decision variables actually make sense only if they have integer values. If requiring integer values is the only way in which a problem deviates from a linear programming formulation, then it is an integer programming (IP) problem. The mathematical model for integer programming is the linear programming model with one additional restriction that the variables must have integer values. If only some of the variables are required to have integer values (so the divisibility assumption holds for the rest), this model is referred to as mixed integer programming (MIP). With just two choices, we can represent such decisions by decision variables that are restricted to just two values, say 0 and 1. Thus, the j^{th} yes-or-no decision would be represented by, say, x_j such that

$$x_j = \begin{cases} 1 & \text{If decision } j \text{ is yes.} \\ 0 & \text{If decision } j \text{ is no.} \end{cases}$$

Such variables are called binary variables (or 0–1 variables). Consequently, IP problems that contain only binary variables sometimes are called binary integer programming (BIP) problems (or 0–1 integer programming problems) (Hillier, 2005).

3. MIP formulation for extraction sequencing

MIP models are generally used to maximize the overall discounted economic value, or net present value (NPV), of a mining project. For this purpose and production scheduling for a bench of mine the following objective function can be used.

$$\text{Maximize} \sum_{t=1}^{T}\sum_{n=1}^{N}\frac{BEV_n^t}{(1+i)^t}\times X_n^t \qquad (2)$$

Where T is the maximum number of scheduling periods, N is the total number of blocks to be scheduled, BEV_n^t is the block economic value of block n in period t, i is interest rate and X_n^t is a binary variable, equal to 1 if the block n is to be mined in period t, otherwise 0.

3.1. Grade blending constraints

The average grade of material sent to the mill has to be less than or equal to a certain grade value, G_{max} , for each period, t (Ramazan *et al.*, 2004).

$$\sum_{n=1}^{N}(g_n - G_{\max})\times Ot_n \times X_n^t \le 0 \qquad (3)$$

Where g_n is the average grade of block n, and Ot_n is the ore tonnage in block n.

The average grade of the material sent to the mill has to be greater than or equal to a certain grade value, G_{min} , for each period, t (Ramazan *et al.*, 2004).

$$\sum_{n=1}^{N}(g_n - G_{\min})\times Ot_n \times X_n^t \ge 0 \qquad (4)$$

3.2. Reserve constraints

Reserve constraints are constructed for each of the blocks to state that all the blocks in the model considered have to be mined once (Ramazan *et al.*, 2004).

$$\sum_{t=1}^{T} X_n^t = 1 \qquad (5)$$

3.3. Processing capacity constraints

The total tonnage of ore processed cannot be more than the processing capacity $(PC_{\max})$ in any period, t (Ramazan *et al.*, 2004).

$$\sum_{n=1}^{N}(Ot_n \times X_n^t) \le PC_{\max} \qquad (6)$$

The total tonnage of ore processed cannot be less than a certain amount $(PC_{\min})$ in any period, t (Ramazan *et al.*, 2004).

$$\sum_{n=1}^{N}(Ot_n \times X_n^t) \geq PC_{\min} \tag{7}$$

3.4. Mining capacity

The total amount of material (waste and ore) to be mined cannot be more than the total available equipment capacity $(MC_{\max})$ for each period, t (Ramazan *et al.*, 2004).

$$\sum_{n=1}^{N}(Ot_n + W_n)\times X_n^t \leq MC_{\max} \tag{8}$$

Where W_n is the tonnage of waste material in block n.

To force the MIP mod el to produce balanced waste production throughout the periods, a lower bound $(MC_{\min})$ may need to be implemented as follows (Ramazan *et al.*, 2004):

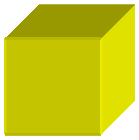
$$\sum_{n=1}^{N}(Ot_n + W_n)\times X_n^t \geq MC_{\min} \tag{9}$$

Note: In this formulation mine equipment mobility has not been considered.

4. Illustrative example

A plan view of 24 blocks and average grade of ore in each block are shown in Figures 1 and 2, respectively. In calculating the values per ton(st) of blocks, it is assumed that the product price is \$250/t, ore and waste mining cost are \$25/t, processing cost is \$40/t, the recovery factor is 70% and the specific weight of ore and waste are 2.7 and 2.4 t/m^3 respectively.

The goal is to schedule the extraction of blocks in three periods. The objective is to maximize the summation of discounted economic block values, while extracting all the 24 blocks. Table 1, shows each block economic value calculated by Equation (10).



Dimension of each block=15 m × 15 m × 15 m

BL1	BL2	BL3	BL4	BL5	BL6
BL7	BL8	BL9	BL10	BL11	BL12
BL13	BL14	BL15	BL16	BL17	BL18
BL19	BL20	BL21	BL22	BL23	BL24

Fig 1. Plan view is showing blocks configuration

0.87	waste	0.64	0.67	0.52	waste
0.51	0.70	0.55	0.73	0.67	0.59
0.95	0.78	0.84	0.71	waste	0.62
waste	0.65	waste	0.62	0.71	0.74

Fig 2. Plan view is showing the average grade (MWT %) of each block.
MWT (magnetic weight recovery)

$$BEV_n = (Ot_n \times g_n \times P \times R) - (Ot_n \times C_p) - [BT_n \times C_m] \qquad (10)$$

Where Ot_n is the total amount of ore in the block n (t), g_n is the average grade of n^{th} block (%), P is the price of product ($/t), R is the proportion of the product recovered by processing the ore (%), C_p is the cost of the processing ($/t), Wtn is the total amount of waste in the block n (t), BT_n is the block tonnage that is equal to (Ot_n + Wt_n) and C_m is the cost of the mining for ore and waste ($/t).

To solve this problem, Excel Solver was used. Solver is a free add-in package for Excel. A more powerful version – Premium Solver – is commercially available for optimization on an industrial scale. Solver allows us to specify restrictions, or constraints, for cell values. For example, we can specify that certain cell values must be equal to, higher than, or lower than a given number. In fact, we can specify up to 500 constraints in a Solver problem, consisting of up to two constraints with up to 200 changing cells per constraint, plus 100 additional constraints. Actually, when the Solver Options dialog box's Assume Linear Model check box is selected, there is theoretically no limit to the number of constraints.

The number of binary variables required for the MIP model is equal to the number of blocks in the model (in this case=24) multiplied by the total periods (in this case=3) to be scheduled, as can be seen in the formulations above.

For mining capacity constraints, following boundaries were used:

$$1^{st}, 2^{nd}, 3^{rd} \ years \qquad 65{,}000(t) \le Mining\ capacity \le 75{,}000(t)$$

Processing capacity that was used as constraint for all periods was same and it was between 55,000 (t) and 60,000 (t).

To use Excel to solve LP problems the Solver add-in must be included. Typically this feature is not installed by default when Excel is first setup on your hard disk. To add this facility to your Tools menu you need to carry out the following steps:

1. Select the menu option Tools → Add-Ins (this will take few moments to load the necessary file).

2. From the dialogue box presented check the box for Solver Add-In.

Table 1. Economic Block Value (volume of each block is equal to 15×15×15 (m^3))

BL No.	Grade (%)	waste (st)	Ore(st)	Cost ($)	Revenue($)	EBV×10^8 ($)
BL 1	87	0	9112.5	592312.5	138,145,500.00	1.38
BL 2	waste	8100	0.0	202500.0	-202,500.00	-0.002
BL 3	64	0	9112.5	592312.5	101,467,687.50	1.01
BL 4	67	0	9112.5	592312.5	106,251,750.00	1.06
BL 5	52	0	9112.5	592312.5	82,331,437.50	0.82
BL 6	waste	8100	0.0	202500.0	-202,500.00	-0.002
BL 7	51	0	9112.5	592312.5	80,736,750.00	0.81
BL 8	70	0	9112.5	592312.5	111,035,812.50	1.11
BL 9	55	0	9112.5	592312.5	87,115,500.00	0.87
BL 10	73	0	9112.5	592312.5	115,819,875.00	1.16
BL 11	67	0	9112.5	592312.5	106,251,750.00	1.06
BL 12	59	0	9112.5	592312.5	93,494,250.00	0.93
BL 13	95	0	9112.5	592312.5	150,903,000.00	1.51
BL 14	78	0	9112.5	592312.5	123,793,312.50	1.24
BL 15	84	0	9112.5	592312.5	133,361,437.50	1.33
BL 16	71	0	9112.5	592312.5	112,630,500.00	1.13
BL 17	waste	8100	0.0	202500.0	-202,500.00	-0.002
BL 18	62	0	9112.5	592312.5	98,278,312.50	0.98
BL 19	waste	8100	0.0	202500.0	-202,500.00	-0.002
BL 20	65	0	9112.5	592312.5	103,062,375.00	1.03
BL 21	waste	8100	0.0	202500.0	-202,500.00	-0.002
BL 22	62	0	9112.5	592312.5	98,278,312.50	0.98
BL 23	71	0	9112.5	592312.5	112,630,500.00	1.13
BL 24	74	0	9112.5	592312.5	117,414,562.50	1.17

3. On clicking OK, you will then be able to access the Solver option from the new menu option Tools → Solver

An optimization model has three parts: the target cell, the changing cells, and the constraints.

- Target cell: The target cell represents the objective or goal. We want to either minimize or maximize the target cell. In our example of a mine production scheduling, we want to maximize the discounted cash flow.
- Changing cells: The changing cells are the spreadsheet cells that we can change or adjust to optimize the target cell. In the other word, these cells are solution of the problem.
- Constraints: The constraints are restrictions you place on the changing cells. Software must adhere to these constraints as it tries to change the adjustable cells to meet the objective.
- Model: A model is the set of target cell, all changing cells and any constraints for the current problem that you want to solve.

To solve this problem, first the solution section was setup as illustrated in figure 3. The cells C3:E26 are changing cells or solution of the problem. In column F, each cell from F3 to F26 is equal to sum of solutions in related row. For instance cell F3=sum (C3:E3) and cell F25=sum (C25:E25). Sum of solutions for each period is written in row 28 and under each period in front of total. For instance cell C28=sum (C3:C26).

Figure 4 shows how constraints of grade blending have been set up in Excel. There are two different parts. Cells H3:J26 for equation (4) and cells K3:M26 for equation (3). In this problem, according to grades in table 1, $G_{\min} = 0$ and $G_{\max} = 0.95$. Value of each cell H3 through to J26 is calculated as follows:

H11=(B11-0)×G11×C11, I11=(B11-0) ×G11×D11 , J11=(B11-0) ×G11×E11

For equation (4), value of each cell K3 through to M26 is calculated as follows:

K18=(B18-0.95) ×G18×C18, L18=(B18-0.95) ×G18×D18, M18=(B18-0.95) ×G18×E18

The cells H28 to M28 are equal to sum of cell values from row 3 to row 26 above each of them. The cells H28 through to J28 indicate equation (4) in periods 1, 2 and 3 and the cells K28 through to M28 indicate equation (3) in periods 1, 2 and 3.

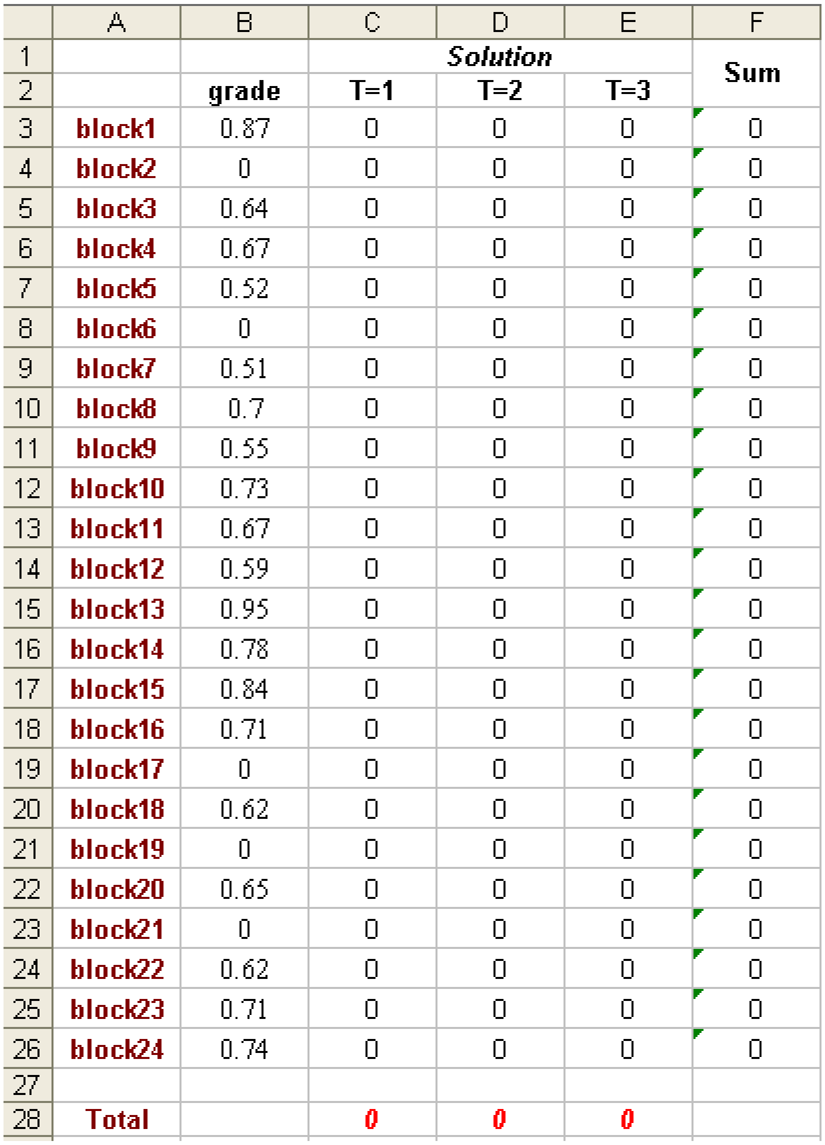
Figure 5 shows how constraints of mining capacity and processing capacity have been set up in Excel. There are two different parts. Cells S3:U26 for equations (8) and (9) and cells W3:Y26 for equations (6) and (7). Value of each cell S3 through to U26 is calculated as follows:

S14=R14 ×C14, T14=R14 ×D14, U14=R14×E11

Value of each cell W3 through to Y26 is calculated as follows:

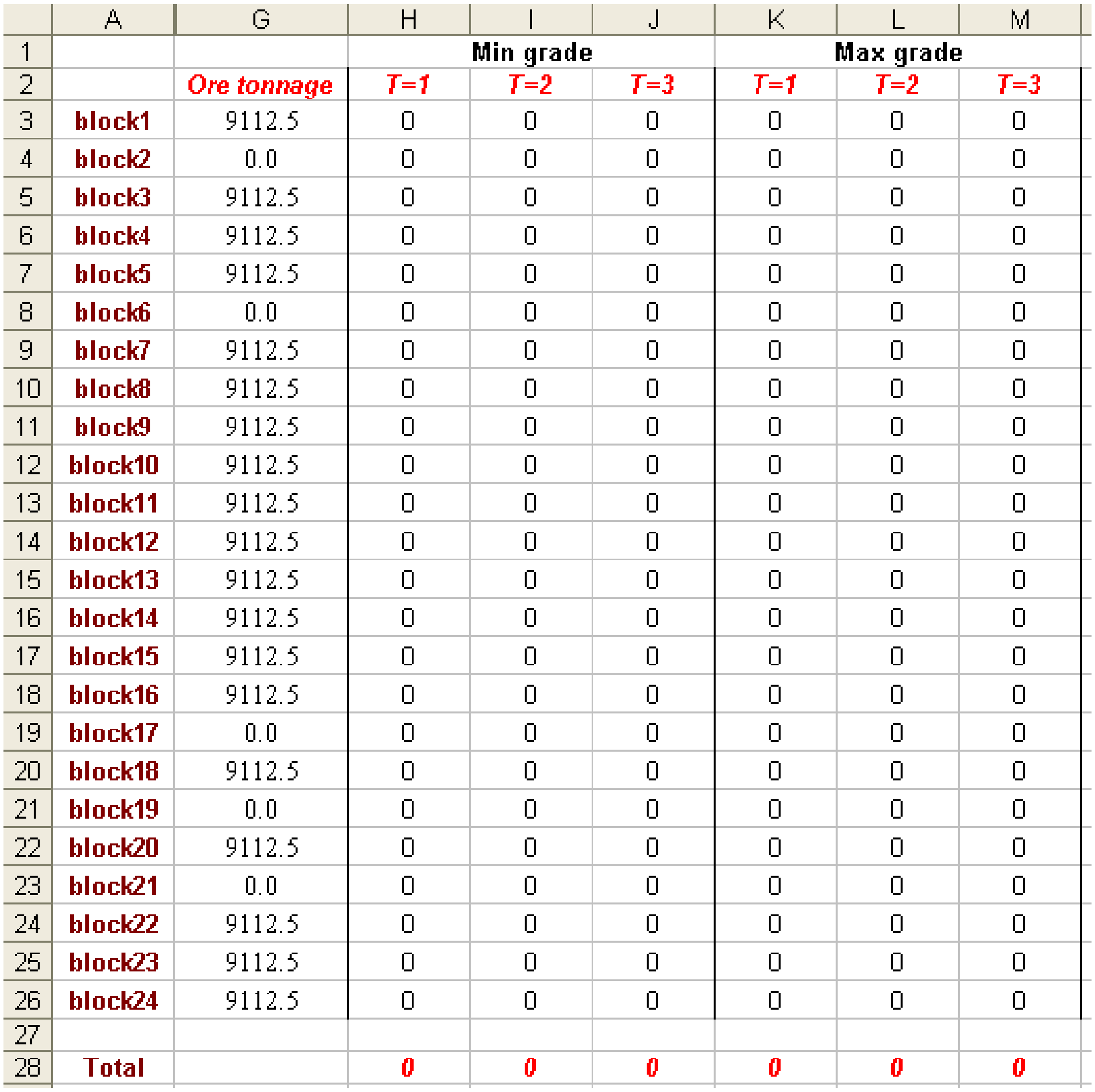
W7=V7×C7, X7=V7 ×D7, Y7=V7×E7

The cells S28 through to Y28 are equal to sum of cell values from row 3 to row 26 above each of them. The cells S28 through to U28 indicate equations (8) and (9) in periods 1, 2 and 3 and the cells W28 through to Y28 indicate equations (6) and (7) in periods 1, 2 and 3.



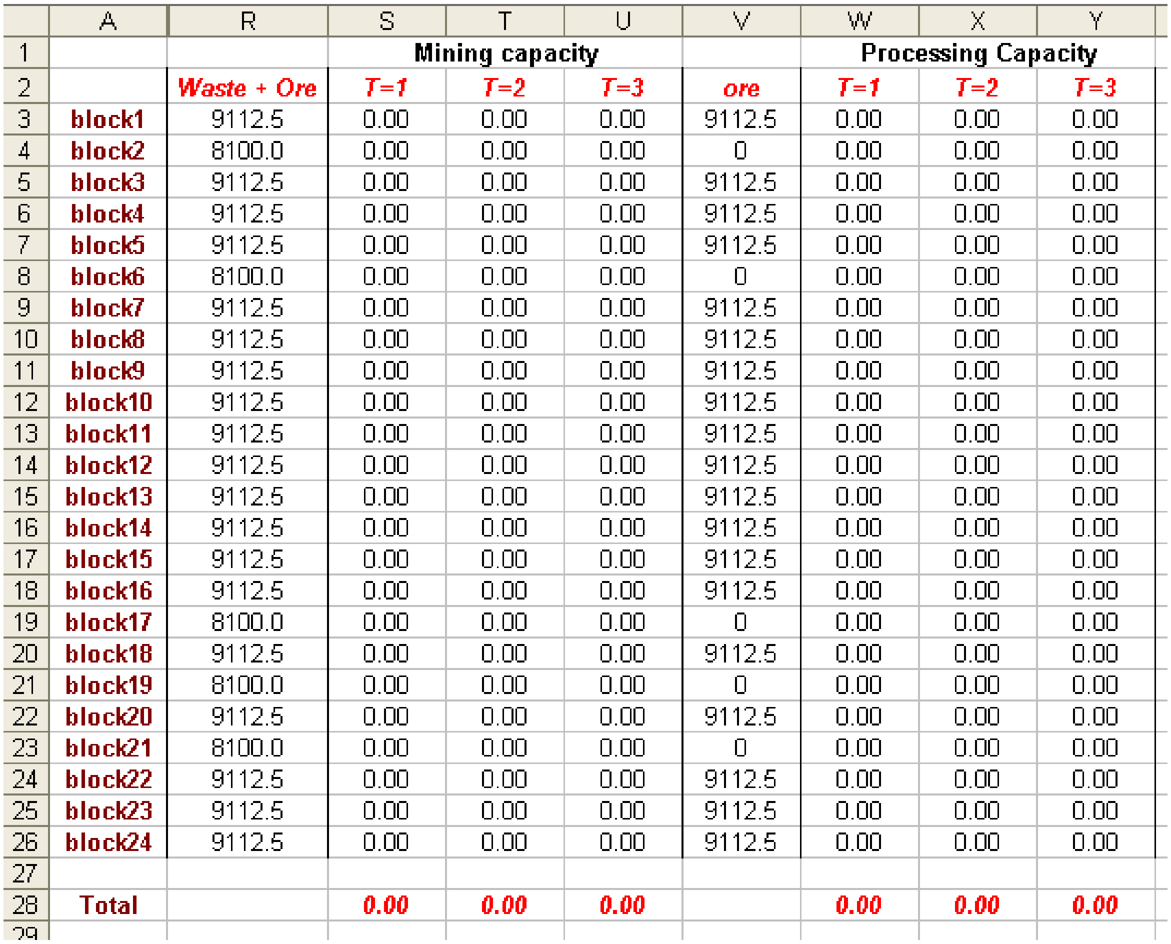
	A	B	C	D	E	F
1			Solution			Sum
2		grade	T=1	T=2	T=3	
3	block1	0.87	0	0	0	0
4	block2	0	0	0	0	0
5	block3	0.64	0	0	0	0
6	block4	0.67	0	0	0	0
7	block5	0.52	0	0	0	0
8	block6	0	0	0	0	0
9	block7	0.51	0	0	0	0
10	block8	0.7	0	0	0	0
11	block9	0.55	0	0	0	0
12	block10	0.73	0	0	0	0
13	block11	0.67	0	0	0	0
14	block12	0.59	0	0	0	0
15	block13	0.95	0	0	0	0
16	block14	0.78	0	0	0	0
17	block15	0.84	0	0	0	0
18	block16	0.71	0	0	0	0
19	block17	0	0	0	0	0
20	block18	0.62	0	0	0	0
21	block19	0	0	0	0	0
22	block20	0.65	0	0	0	0
23	block21	0	0	0	0	0
24	block22	0.62	0	0	0	0
25	block23	0.71	0	0	0	0
26	block24	0.74	0	0	0	0
27						
28	Total		0	0	0	

Fig 3. The part of worksheet that was used for setting up solution of problem



	A	G	H	I	J	K	L	M
1				Min grade			Max grade	
2		Ore tonnage	T=1	T=2	T=3	T=1	T=2	T=3
3	block1	9112.5	0	0	0	0	0	0
4	block2	0.0	0	0	0	0	0	0
5	block3	9112.5	0	0	0	0	0	0
6	block4	9112.5	0	0	0	0	0	0
7	block5	9112.5	0	0	0	0	0	0
8	block6	0.0	0	0	0	0	0	0
9	block7	9112.5	0	0	0	0	0	0
10	block8	9112.5	0	0	0	0	0	0
11	block9	9112.5	0	0	0	0	0	0
12	block10	9112.5	0	0	0	0	0	0
13	block11	9112.5	0	0	0	0	0	0
14	block12	9112.5	0	0	0	0	0	0
15	block13	9112.5	0	0	0	0	0	0
16	block14	9112.5	0	0	0	0	0	0
17	block15	9112.5	0	0	0	0	0	0
18	block16	9112.5	0	0	0	0	0	0
19	block17	0.0	0	0	0	0	0	0
20	block18	9112.5	0	0	0	0	0	0
21	block19	0.0	0	0	0	0	0	0
22	block20	9112.5	0	0	0	0	0	0
23	block21	0.0	0	0	0	0	0	0
24	block22	9112.5	0	0	0	0	0	0
25	block23	9112.5	0	0	0	0	0	0
26	block24	9112.5	0	0	0	0	0	0
27								
28	Total		0	0	0	0	0	0

Fig 4. The part of worksheet that was used for setting up constraints (3) and (4)



	A	R	S	T	U	V	W	X	Y
1			Mining capacity				Processing Capacity		
2		*Waste + Ore*	*T=1*	*T=2*	*T=3*	*ore*	*T=1*	*T=2*	*T=3*
3	block1	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
4	block2	8100.0	0.00	0.00	0.00	0	0.00	0.00	0.00
5	block3	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
6	block4	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
7	block5	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
8	block6	8100.0	0.00	0.00	0.00	0	0.00	0.00	0.00
9	block7	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
10	block8	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
11	block9	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
12	block10	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
13	block11	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
14	block12	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
15	block13	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
16	block14	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
17	block15	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
18	block16	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
19	block17	8100.0	0.00	0.00	0.00	0	0.00	0.00	0.00
20	block18	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
21	block19	8100.0	0.00	0.00	0.00	0	0.00	0.00	0.00
22	block20	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
23	block21	8100.0	0.00	0.00	0.00	0	0.00	0.00	0.00
24	block22	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
25	block23	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
26	block24	9112.5	0.00	0.00	0.00	9112.5	0.00	0.00	0.00
27									
28	Total		*0.00*	*0.00*	*0.00*		*0.00*	*0.00*	*0.00*
29									

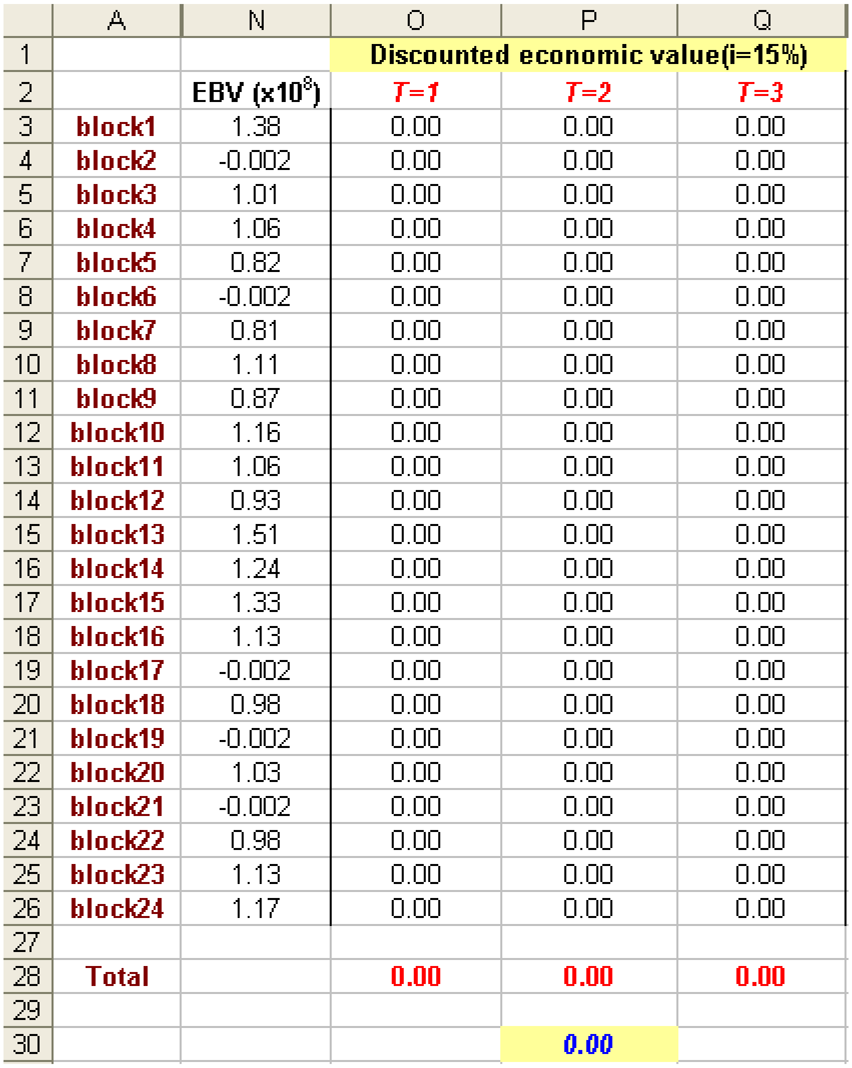
Fig 5. The part of worksheet that was used for setting up constraints (6) to (9)

Setting up the objective function has been illustrated in figure 6. Value of cells N3:Q26 are calculated using equation (11) for each block according to the period of extraction.

$$\frac{BEV_n^t}{(1+i)^t} \times X_n^t \qquad (11)$$

Where BEV_n^t is the block economic value of block n in period t, i is interest rate and X_n^t is a binary variable, equal to 1 if the block n is to be mined in period t, otherwise 0.

The cells O28 through to Q28 are equal to sum of cell values from row 3 to row 26 above each of them. The cell P30 is target cell that must be maximized and it is equal to sum of cells O28 through Q28.



	A	N	O	P	Q
1			Discounted economic value(i=15%)		
2		EBV ($\times 10^8$)	T=1	T=2	T=3
3	block1	1.38	0.00	0.00	0.00
4	block2	-0.002	0.00	0.00	0.00
5	block3	1.01	0.00	0.00	0.00
6	block4	1.06	0.00	0.00	0.00
7	block5	0.82	0.00	0.00	0.00
8	block6	-0.002	0.00	0.00	0.00
9	block7	0.81	0.00	0.00	0.00
10	block8	1.11	0.00	0.00	0.00
11	block9	0.87	0.00	0.00	0.00
12	block10	1.16	0.00	0.00	0.00
13	block11	1.06	0.00	0.00	0.00
14	block12	0.93	0.00	0.00	0.00
15	block13	1.51	0.00	0.00	0.00
16	block14	1.24	0.00	0.00	0.00
17	block15	1.33	0.00	0.00	0.00
18	block16	1.13	0.00	0.00	0.00
19	block17	-0.002	0.00	0.00	0.00
20	block18	0.98	0.00	0.00	0.00
21	block19	-0.002	0.00	0.00	0.00
22	block20	1.03	0.00	0.00	0.00
23	block21	-0.002	0.00	0.00	0.00
24	block22	0.98	0.00	0.00	0.00
25	block23	1.13	0.00	0.00	0.00
26	block24	1.17	0.00	0.00	0.00
27					
28	Total		0.00	0.00	0.00
29					
30				0.00	

Fig 6. The part of worksheet that was used for setting up objective function

To use Excel Solver with the objective of discounted cash flow maximization of a mine production schedule, the following steps must be followed:

1.Click **Tools** → **Solver**

2. Click **Reset All**, and then click OK to clear Solver's existing settings.

3. Click the **Set Target Cell** box, and then click cell P30

4.Click **Max**

5. Click the **By Changing Cells** box, and then select cells C3 through E26

6. Click **Add**

7. Click the **Cell Reference** box, and then select cells C3 through E26

8. In the operator list between the Cell Reference and constraint boxes, select **bin**.

9. Click **Add**.

10. Click the **Cell reference** box, and then select cell F3

11. In the operator list , select =
12.click the **Constraint** box, and then type the number 1
(**Note**: Why one? because each block can be mined once)
13. Repeat steps 9 to 12 for cells F4 through F26
14.Click **Add**
15. Click the **Cell reference** box, and then select cell C28
16. In the operator list , select =
17. click the **Constraint** box, and then type the number 8
(**Note**: Why 8? because we want to extract 8 blocks in each period)
18. Repeat steps 14 to 17 for cells D28 and E28
19. Click **Add**.
20. Click the **Cell reference** box, and then select cell H28
21. In the operator list , select >=
22.Click the **Constraint** box, and then type the number 0 (refer to equation (4))
23. Repeat steps 19 to 22 for cells I28 and J28
24. Click **Add**.
25. Click the **Cell reference** box, and then select cell K28
26. In the operator list , select <=
27.Click the **Constraint** box, and then type the number 0 (refer to equation (3))
28. Repeat steps 24 to 27 for cells L28 and M28
29. Click **Add**.
30. Click the **Cell reference** box, and then select cell S28
31. In the operator list , select <=
32.Click the **Constraint** box, and then type the number 75,000 (Maximum mining capacity)
33. Click **Add**.
34. Click the **Cell reference** box, and then select cell S28
35. In the operator list , select >=
36. Click the **Constraint** box, and then type the number 65,000 (Minimum mining capacity)
37. Repeat steps 29 to 36 for cells T28 and U28
38. Click **Add**.
39. Click the **Cell reference** box, and then select cell W28
40. In the operator list , select <=
41.Click the **Constraint** box, and then type the number 60,000 (Maximum processing capacity)
42. Click **Add**.
43. Click the **Cell reference** box, and then select cell W28
44. In the operator list , select >=
45. Click the **Constraint** box, and then type the number 55,000 (Minimum processing capacity)
46. Repeat steps 38 to 45 for cells X28 and Y28

47.Click **OK**

48. Click **Option**

49. Select the **Assume Linear Model** and **Use Automatic Scaling** check boxes, and then click OK.

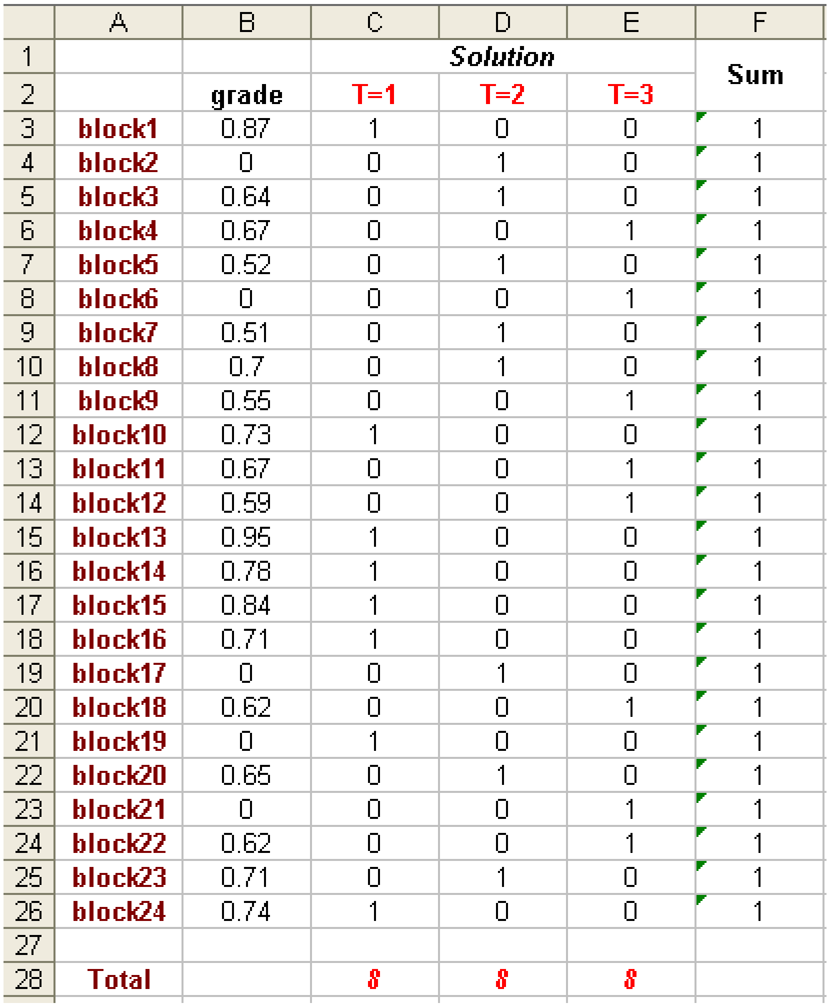
50. Click **Solve**

5. Results and Discussion

The production scheduling results obtained by applying the optimization formulation in equation (2) and constraints are shown in figure 7.

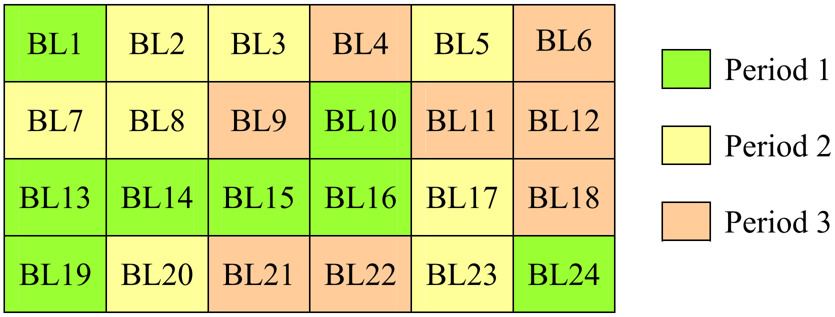
Figure 8 shows the optimal schedule for three scheduled periods. Summary of results have been presented in table 2. Comparisons between tonnages of ore and waste that are extracted in each period have been shown in figure 9. It is assumed that maximum and minimum mining capacities are 65,000 ts and 75,000 ts, respectively. For processing capacity, maximum and minimum capacities were assumed 55,000 ts and 60,000 ts, respectively. The obtained solution satisfies these conditions. This matter has been shown in figure 10.

.



	A	B	C	D	E	F
1			*Solution*			**Sum**
2		**grade**	**T=1**	**T=2**	**T=3**	
3	**block1**	0.87	1	0	0	1
4	**block2**	0	0	1	0	1
5	**block3**	0.64	0	1	0	1
6	**block4**	0.67	0	0	1	1
7	**block5**	0.52	0	1	0	1
8	**block6**	0	0	0	1	1
9	**block7**	0.51	0	1	0	1
10	**block8**	0.7	0	1	0	1
11	**block9**	0.55	0	0	1	1
12	**block10**	0.73	1	0	0	1
13	**block11**	0.67	0	0	1	1
14	**block12**	0.59	0	0	1	1
15	**block13**	0.95	1	0	0	1
16	**block14**	0.78	1	0	0	1
17	**block15**	0.84	1	0	0	1
18	**block16**	0.71	1	0	0	1
19	**block17**	0	0	1	0	1
20	**block18**	0.62	0	0	1	1
21	**block19**	0	1	0	0	1
22	**block20**	0.65	0	1	0	1
23	**block21**	0	0	0	1	1
24	**block22**	0.62	0	0	1	1
25	**block23**	0.71	0	1	0	1
26	**block24**	0.74	1	0	0	1
27						
28	**Total**		***8***	***8***	***8***	

Fig 7. Solution of the problem that was obtained using Excel solver



🟩 BL1	🟨 BL2	🟨 BL3	🟧 BL4	🟨 BL5	🟧 BL6
🟨 BL7	🟨 BL8	🟧 BL9	🟩 BL10	🟨 BL11	🟧 BL12
🟩 BL13	🟩 BL14	🟩 BL15	🟩 BL16	🟨 BL17	🟧 BL18
🟩 BL19	🟨 BL20	🟧 BL21	🟧 BL22	🟨 BL23	🟩 BL24

🟩 Period 1

🟨 Period 2

🟧 Period 3

Fig 8. Block extraction sequence

Table 2. Summary of obtained results for each period using Excel Solver

	Period 1	Period 2	Period 3
Blocks Number	1,10,13,14,15 16, 19, 24	2,3,5,7,8, 17,20,23	4,6,9,11,12, 18,21,22
Waste (t)	8100	16200	16200
Ore (t)	63787.5	54675	54675
Waste+Ore (t)	71887.5	70875	70875
Average grade	0.8	0.62	0.62
$f = \sum \frac{BEV_n}{(1+i)^t}$	7.348×10^8	4.493×10^8	4.785×10^8
$\sum_{t=1}^{3}\sum_{n=1}^{24}\frac{BEV_n^t}{(1+i)^t}$		16.626×10^8	

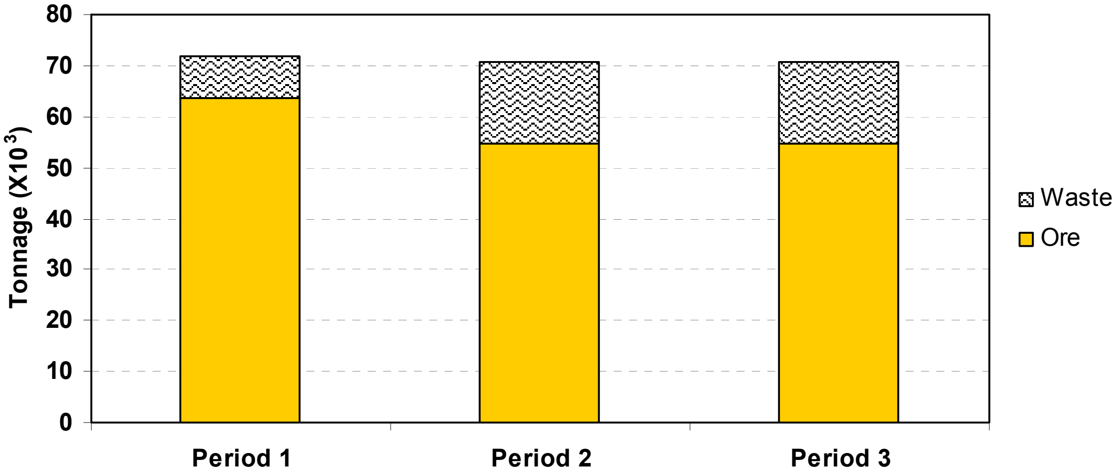


Fig 9. Amount of ore and waste in each period

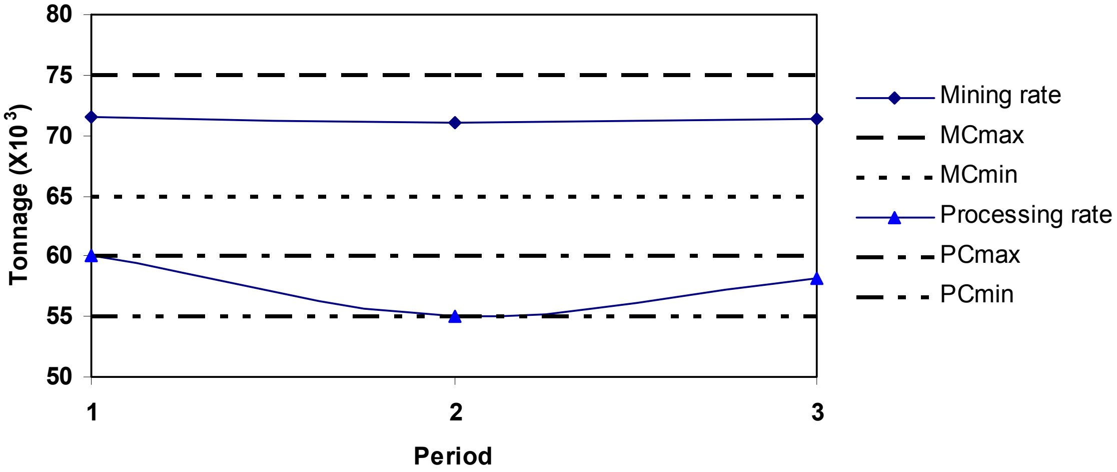


Fig 10. Relationship between PCmax, PCmin, MCmax, MCmin and obtained solution

6. Conclusions

This paper was a general review of open pit mining scheduling problem definition. Several optimization methods have appeared in the literature, but among of those optimization techniques, mixed integer programming is recognized as having significant potential for optimizing production planning in open pit mines with the objective of maximizing the discounted cash flow. Review of previous studies about MIP shows in open pit mines with many numbers of block, modeling and formulation of production planning need too many binary variables and this makes it very difficult or impossible to solve with the current state of hardware and software.

A step by step guideline for using the Excel Solver for solving MIP models was documented in conjunction with an illustrative example. The results show that we can use the Excel Solver software for a bench by bench short-term production scheduling.

7. References

[1] Boland, N., Dumitrescu, I., Froyland, G., and Gleixner, A. M., (2009), "LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity", *Computers & Operations Research*, Vol. 36, 4, pp. 1064-1089.

[2] Boland, N., Frick, C., and Froyland, G., (2006), "A strengthened formulation for the open pit mine production scheduling problem", Optimization online: an eprint site for the optimization community, © Retrieved from: http//:optimization-online.org/DB_HTML/2007/03/1624.html

[3] Caccetta, L. and Hill, S., (2003), "An application of branch and cut to open pit mine scheduling", *Journal of Global optimization*, Vol. 27, pp. 349-365.

[4] Denby, D. B. and Schofield, D., (1995), "Inclusion of risk assessment in open pit design and scheduling", *Trans. Instn Min. Metall.(Sec.A: Min Industry)*, Vol. 104, pp. A67-A71.

[5] Dowd, P. A., (1994), "Risk assessment in reserve estimationand open pit planning", *Sec.A: Min. Industry, The Institution of mining and Metallurgy*, Vol. 103, pp. A148-A154.

[6] Fricke, C., (2006), "Application of integer programming in open pit mining",Thesis, © The university of Melbourne,

[7] Godoy, M. C. and Dimitrakopoulos, R., (2004), "Managing risk and waste mining in long-term production scheduling of open pit mine", *in Proceedings of SME Transaction*, © pp. 43-50.

[8] Golamnejad, J., Osanloo, M., and Karimi, B., (2006), "A chance-constrainted programming approach for open pit long-term production scheduling in stochastic enviroments", *Journal of the South African institute of Mining and Metallurgy*, Vol. 106, pp. 105-114.

[9] Hillier, F. S., (2005), "Introduction to operation research", © McGraw-Hill, Pages 1061.

[10] Ramazan, S. and Dimitrakopoulos, R., (2003), "Production scheduling optimization in Nickel laterite deposit: MIP and LP application and in the presence of orebody variability mine", *in Proceedings of MPES2003*, © pp. 1-5.

[11] Ramazan, S. and Dimitrakopoulos, R., (2004), "Recent applications of operations research and efficient MIP formulation in open pit mining", *Transactions of the Society for Mining,Metallurgy and Exploration*, Vol. 316, pp. 73-78.

[12] Ramazan, S. and Dimitrakopoulos, R., (2004), "Traditional and new MIP models for production scheduling with in-situ grade variability", *International Journal of Surface Mining, Reclamation and Environment*, Vol. 18, pp. 85-98.

McMurray oil sands pit design using Gemcom and Whittle

Samira Kalantari

1. Introduction

Mine planning plays an important role in the overall efficiency of mining operations; in other words, the success of mine production is highly dependant on the mine scheduling and mine planning. General mine planning (GMP) software such as Gemcom are useful tools to increase mine productivity. We used Gemcom and Whittle to design the optimum final pit from the drillhole information which we imported into Gemcom. In this project the data is coming from the Fort McMurray oil sands in northern Alberta. From the three oil sands deposit in Alberta, only the Athabasca deposit is economically minable using surface mining techniques (Boratynec, 2003).

In the Fort McMurray oil sands, the clear water formation, a highly consolidated clay, overlies the oil sands and must be stripped before exposing the oil sands for mining. The Mcmurray formation overlies Devonian carbonates of the Beaver hill Lake.

The subdivision for the McMurray formation includes the Lower, Middle and Upper McMurray formation.

- LKM overlies Devonian carbonates and is comprised of gravel, coarse sand, slit and clay,
- MKM overlies the Lower McMurray fluvial deposits and is comprised of a fitting-upward sedimentary package with clean blocky sands,
- UKM is the topmost unconformity separating McMurray succession from the overlying Wabiskaw (Hein et al., 2000).

2. Methodology

2.1. Importing into Gemcom

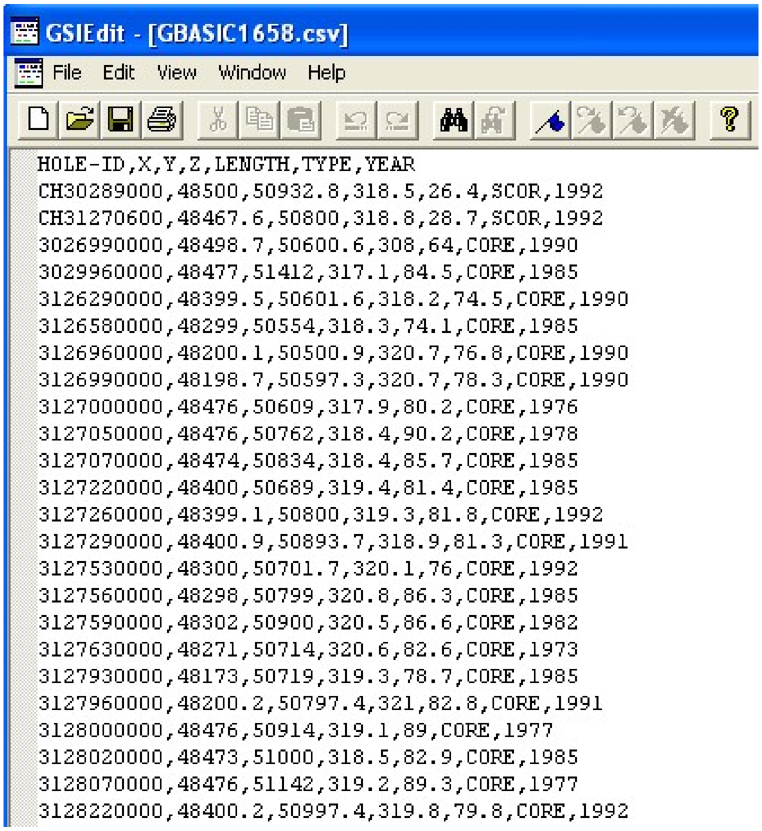
The first step is to import the drillholes information into Gemcom. We have oil sands drillhole data with hole-id, location X, Y, and Z coordinates of the drillholes, the length, type and the year of the drillholes in the Header file. The lithology file contains rock-type and rock-codes and the intervals. As it was discussed before, there are different rock-types in the Mcmurray oil sands deposits such as CWF (Clearwater Formation), PLU, LKM (Lower Mcmurray), MKM (the Middle Mcmurray), UKM (the Upper Mcmurray) and DW (Devonian Carbonates). Each rock-type has the corresponding rock-code assigned to it, for example in our data the code for the LKM is 1, for MKM is 5, and for the UKM is 15, etc.

The information about the Mcmurray rock-types and rock-codes is provided in the lithology file.

The assay file contains the information about the grade of bitumen in the specific intervals within the length of the drillholes. In the survey file the azimuth and dip of each drillhole is provided.

The gfine file has the hole-ids, the intervals and the fines percent in each interval and the gsed file contains the borehole identifier, the thickness and hardness of the siltstone. In order to import these files into Gemcom, their format should be changed to CSV.

We imported this information as CSV (comma delimited) files into Gemcom. Figure 1 shows the first lines of the header file which is imported to Gemcom.



```
HOLE-ID,X,Y,Z,LENGTH,TYPE,YEAR
CH30289000,48500,50932.8,318.5,26.4,SCOR,1992
CH31270600,48467.6,50800,318.8,28.7,SCOR,1992
3026990000,48498.7,50600.6,308,64,CORE,1990
3029960000,48477,51412,317.1,84.5,CORE,1985
3126290000,48399.5,50601.6,318.2,74.5,CORE,1990
3126580000,48299,50554,318.3,74.1,CORE,1985
3126960000,48200.1,50500.9,320.7,76.8,CORE,1990
3126990000,48198.7,50597.3,320.7,78.3,CORE,1990
3127000000,48476,50609,317.9,80.2,CORE,1976
3127050000,48476,50762,318.4,90.2,CORE,1978
3127070000,48474,50834,318.4,85.7,CORE,1985
3127220000,48400,50689,319.4,81.4,CORE,1985
3127260000,48399.1,50800,319.3,81.8,CORE,1992
3127290000,48400.9,50893.7,318.9,81.3,CORE,1991
3127530000,48300,50701.7,320.1,76,CORE,1992
3127560000,48298,50799,320.8,86.3,CORE,1985
3127590000,48302,50900,320.5,86.6,CORE,1982
3127630000,48271,50714,320.6,82.6,CORE,1973
3127930000,48173,50719,319.3,78.7,CORE,1985
3127960000,48200.2,50797.4,321,82.8,CORE,1991
3128000000,48476,50914,319.1,89,CORE,1977
3128020000,48473,51000,318.5,82.9,CORE,1985
3128070000,48476,51142,319.2,89.3,CORE,1977
3128220000,48400.2,50997.4,319.8,79.8,CORE,1992
```

Figure 1- Header file imported to Gemcom

After importing the data into Gemcom, we should create display profiles to see the traces. The display profiles can be created regarding to the grade of bitumen, fines percent, rock-type and rock-code. To display the grade of oil, for each grade range we assigned a specific color, for example we defined the pink color for the grade range between 0 and 5, while for the grade range between 5 and 10 the defined color was light blue.

To display the drillholes according to different rock-types we defined different colors to different rock-codes, the same rule applies when displaying drillholes according to the fine percent. Figure 2 shows the drillholes according to the different rock-types within the length of the drillholes.

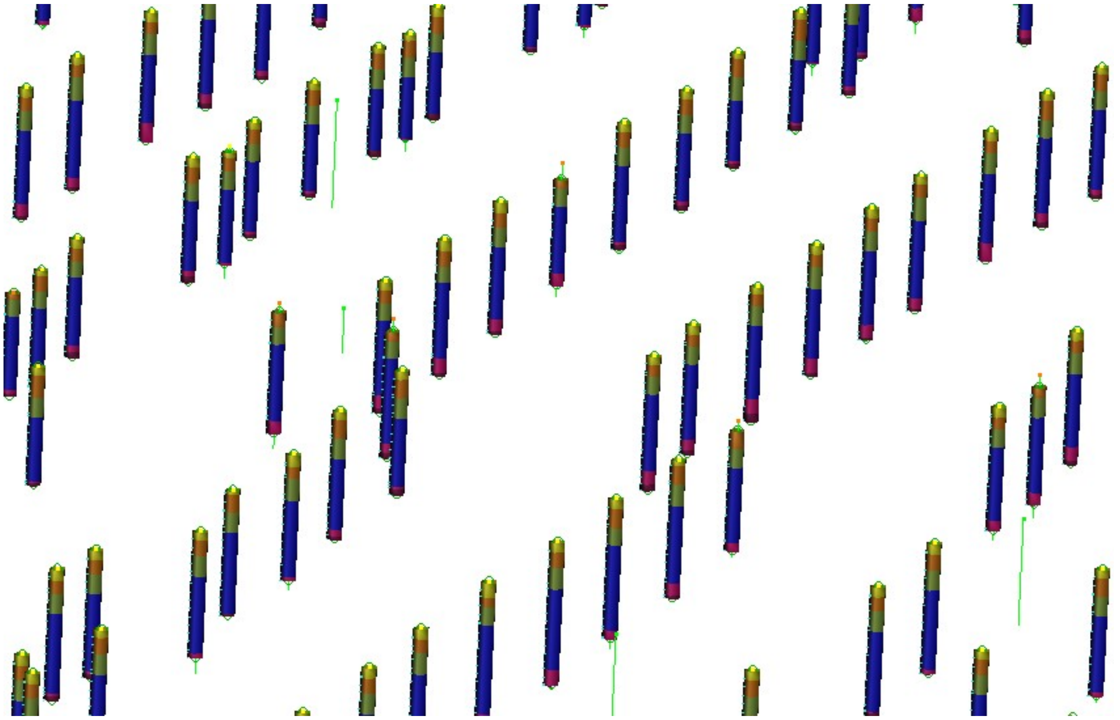


Figure 2- Displaying drillholes according to rock-type

As it can be seen from figure 1, different colors within the length of each drillholes shows the corresponding rock-type for that length, figure 3 shows a sample drillhole and different rock-types on it.

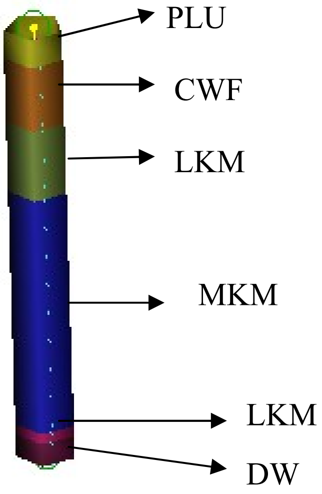


Figure 3- A sample drillhole with different rock-types displaying on it

2.2. Drillhole compositing

Drillhole composites enable grades (or any other values) to be combined or averaged over different ranges. First we did the compositing for the lithology, by rock-codes, and we chose grouping similar values as the method for compositing the rock-codes. The composite rock-codes file is called comp-lith (Gemcome Labs, 2008).

In the next stage, we did the compositing for the assay file, in other words; the compositing was done for the grades. The destination field is Oil and the source table should be from the assays, and the corresponding source field should be oil% (taken from the source table). For preparing the comp-grade, the method for creating composite intervals was length within intervals from another table and we used the comp-lith table in this part. The idea behind this is to increase the accuracy of compositing as it will assign the grade of bitumen and fines percent to the fixed length that we defined. In order to do the compositing for the grade of bitumen, the comp-lith table was used to assign the grade to the fixed length of 5.0 meters. The table for compositing is comp-grade while the controlling table is the comp-lith and the composite length is 5.0 m.

For creating the composite table for the gfine and gsed we do the same thing as we did for the comp-grade table, we used the comp-gsed table with comp-lith as controlling table and the composite length is 5.0 m. The same rule applies to creating the composite table for the gfine.

2.2.1 Data analysis

After compositing, we did the analysis for the data with creating the histograms of bitumen percent in each rock-type.

1. Bitumen Percent

    a) Histogram

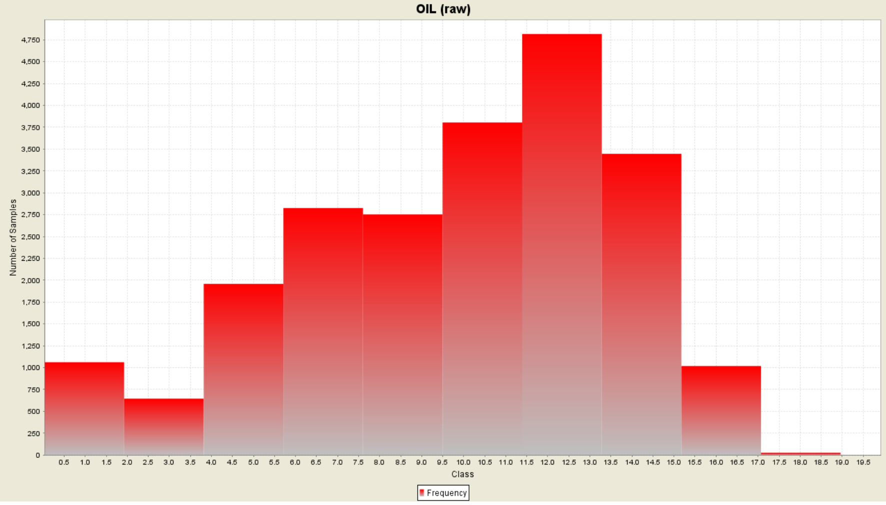


Figure 4- Histogram of bitumen content

Table 1- Descriptive stats for bitumen percent.

NUMBER OF SAMPLES	MEAN	VARIANCE	STD.	MEDIAN(%)
22355	9.82	14.81	3.84	10.46

b) Trend Analysis

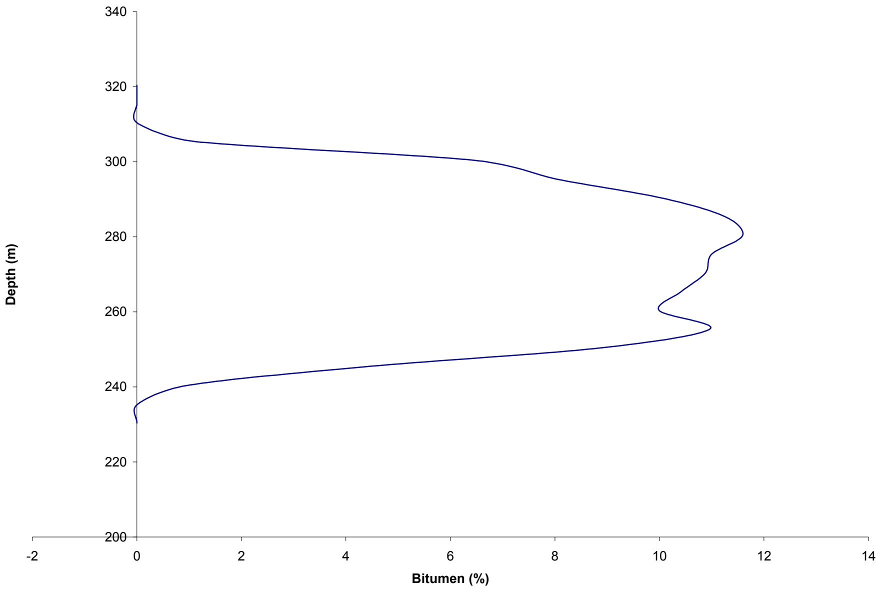


Figure 5- Trend analysis for bitumen content in ore

2. Bitumen percent in LKM

c) Histogram

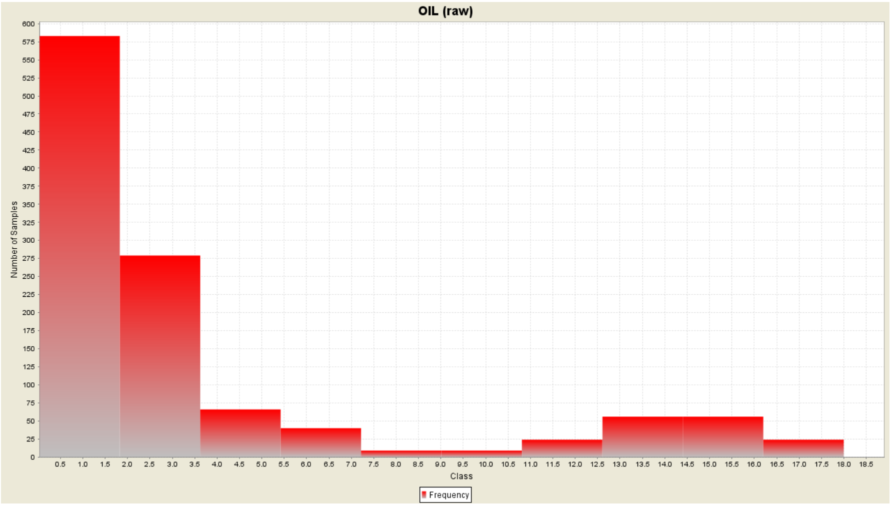


Figure 5- Bitumen percent in LKM

Table 2- Bitumen percent in LKM.

NUMBER OF SAMPLES	MEAN	VARIANCE	STD.	MEDIAN
1146	3.66	22.07	4.70	1.81

3. Bitumen percent in MKM

d) Histogram

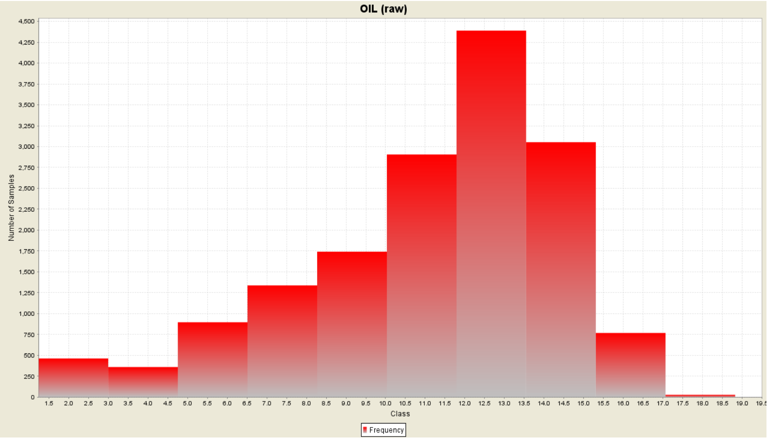


Figure 6- Bitumen percent in Middle Mcmurray (MKM).

Table 3- Bitumen percent in MKM.

NUMBER OF SAMPLES	MEAN	VARIANCE	STD.	MEDIAN
15941	11.31	10.68	3.268	11.91

4. Bitumen percent in UKM

e) Histogram

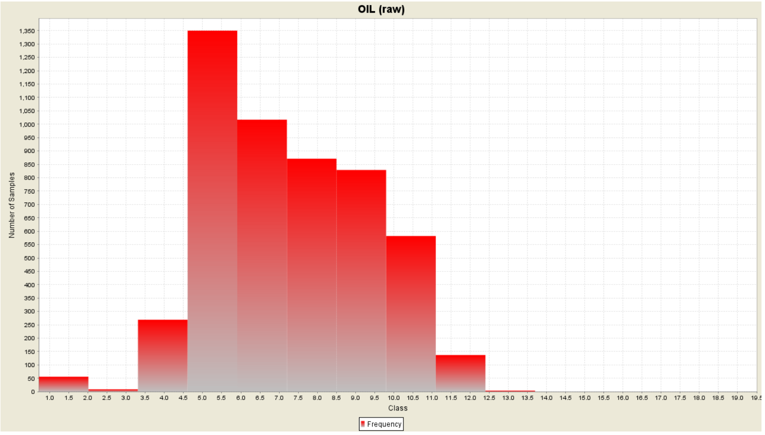


Figure 7- Histogram of bitumen percent in Upper Mcmurray (UKM).

Table 4- Bitumen percent in UKM.

NUMBER OF SAMPLES	MEAN	VARIANCE	STD.	MEDIAN
5124	7.278	4.40	2.09	7.07

2.3. Seam modeling

In working with the oil sands data instead of modeling the solids by outlining polygons, we modeled the interfaces between the layers. In computer models, seams are taken to be continuous over the region and the interfaces between the layers are modeled instead of outlining polygons. First the surfaces of each rock-type were created and afterwards we created the solid between each two surface. The topmost solid is PLU, and the next solids respectively from top to the bottom are CWF, UKM, MKM, LKM and Devonian Carbonates.

Figure 8 shows the solid of MKM.

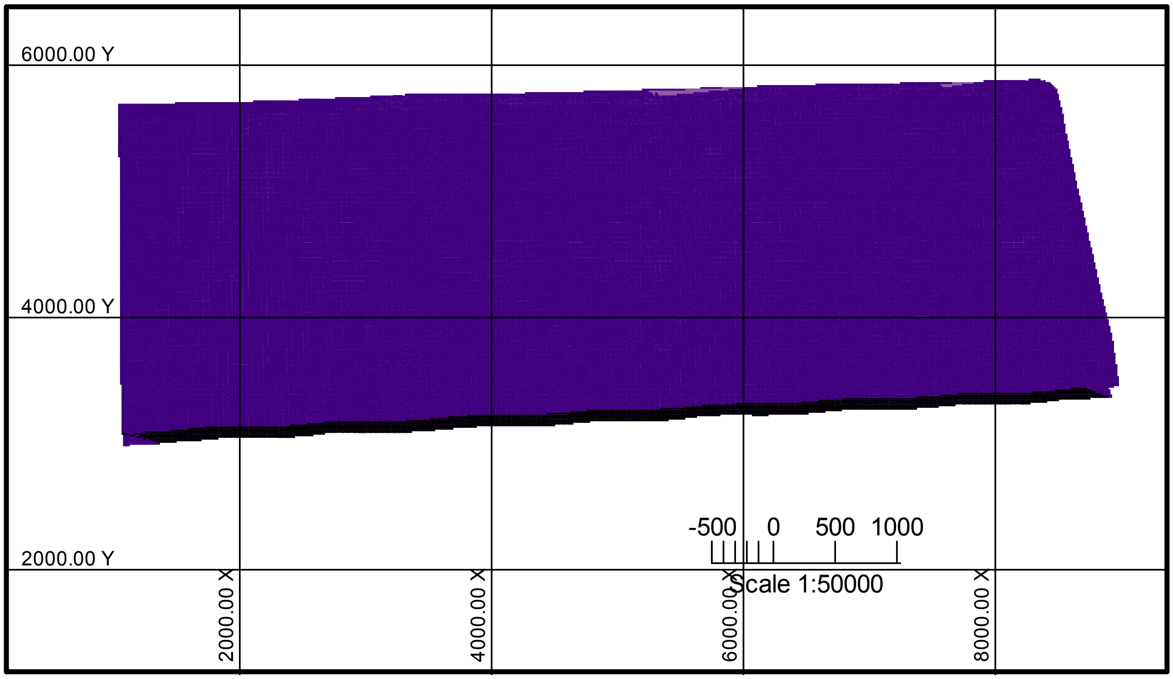


Figure 8- MKM solid.

2.4. Block model

The next step after creating the solids is to build the block model. In order to model ore bodies we use three-dimensional arrays of rectangular blocks where each block is the representative of one characteristic of the volume of material such as rock-type, grade of bitumen, fines percent, density and economic value. We use block modeling for pit design and mine planning. The size of the block that we used was 50m×50m×15m and according to the area we used 195 columns, 85 number of rows and 7 levels for the block model. First of all we should initialize rock-types to waste, in other words we should initialize rock-type as if all the blocks are waste material, and then we define the blocks above the topographic surface (which is on top of the PLU surface) as air blocks. Afterwards we update the rock-types with the solids which we created in the seam modeling.

After updating rock-types from the solids, we should enter the rock densities. As we did for rock-type, we should initialize density with zero, and then it should be updated from rock-type model box.

In previous steps we assigned rock-type and density to the blocks, the grade model should be calculated now. The grade model could be calculated using estimation methods such as inverse distance or Geostatistical techniques. We did the interpolation of the blocks using Inverse Distance Squared to calculate the grade model. Figure 9 shows the block model after the interpolation.

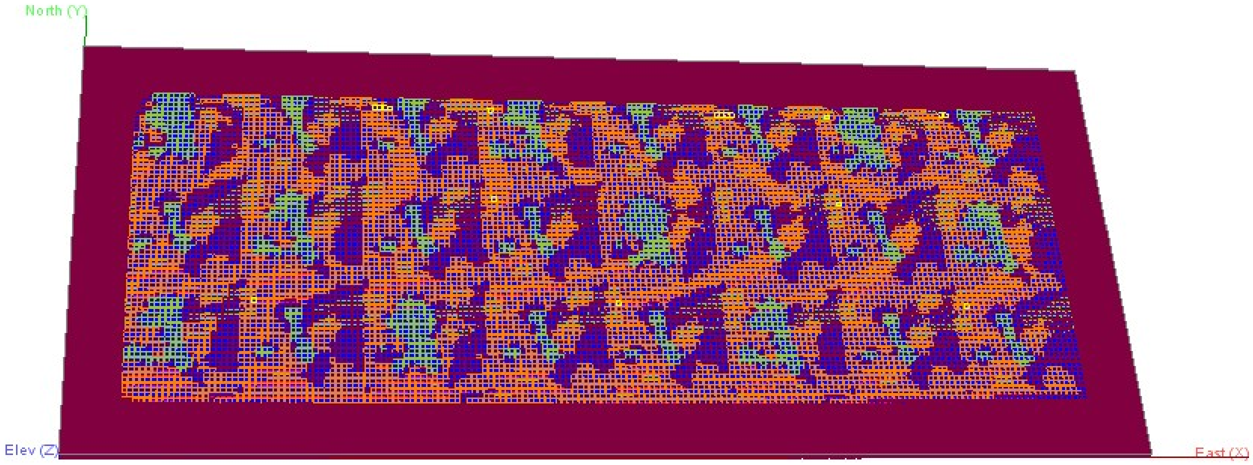


Figure 9- Block model after interpolation.

3. Pit optimization

After the interpolation for the blocks we should send the data to Whittle. Whittle is popular pit optimization and strategic planning software. Using Whittle we can optimize the mine according to the mine life, NPV, etc. After exporting data from Gemcom to Whittle, we used Whittle to optimize the pit according to the mine life. Table 5 shows the economic information for the oil sands. After setting up the pit information, we can run the model to optimize mine life. We can use pit by pit design to check the result of pit optimization.

Table 5- Economic information for oil sands.

Mining reference cost	$4.60
Mining recovery	0.88
Processing costs	$5.03
Processing recovery	0.95
Selling Price ($/Bit %mass)	$2.81
Pit slope	14

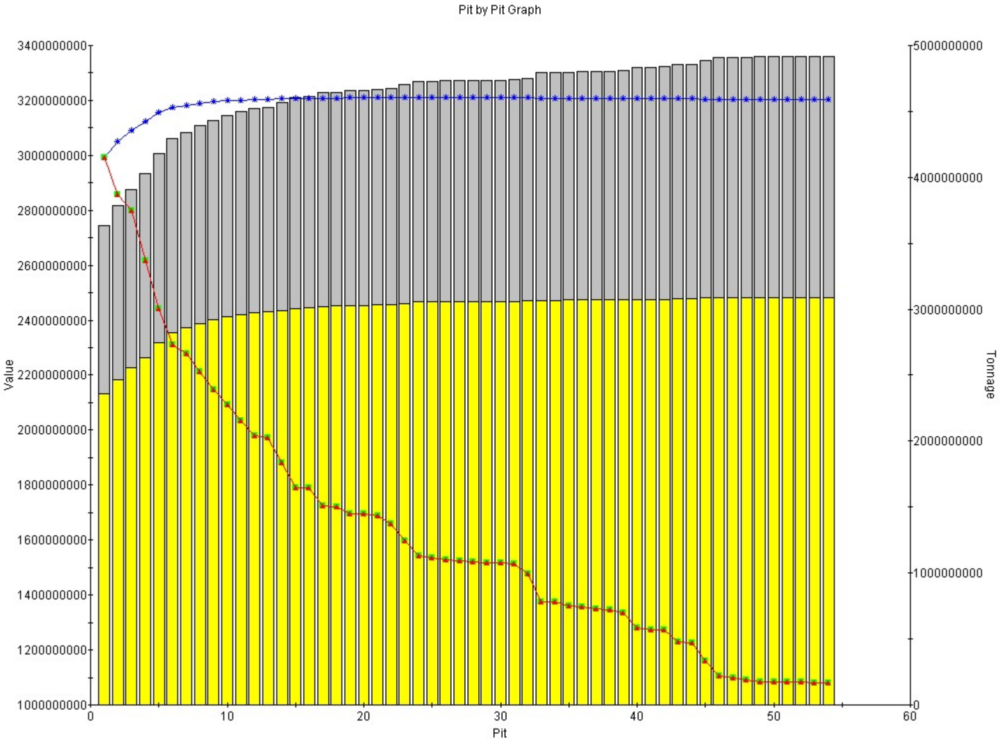


Figure 10– Pit by pit graph.

According to figure 9 we can find out that the red line shows the discounted open pit value for worst case while the blue line represents the discounted open pit value for best case. The yellow region represents the tonnage input to processing for best case while the grey region shows the tonnage of waste rock. According to the pit by pit graph we have 54 pits, and the pit with highest cash flow is pit 26 where the open pit cash flow in the best case is $3,209 million and the mine life years specified is 39.44 years. The tonnage input to processing for the best case in pit 26 is 3,057 million tonnes and the waste tonnage for this pit is specified to be 1,675 million tonnes. Figure 11 shows the new schedule graph.

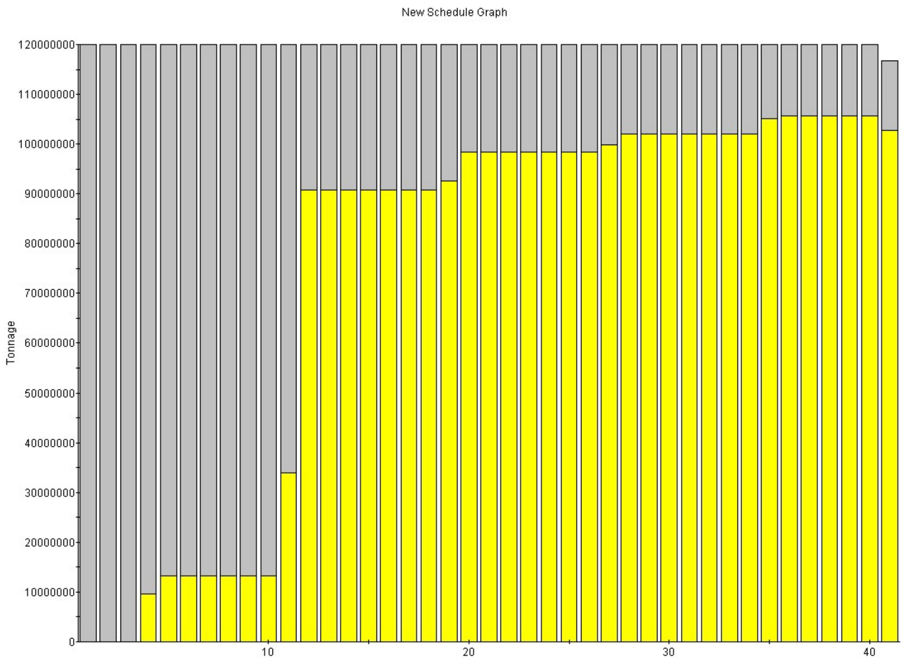


Figure 11– The new schedule graph.

We exported pit 26 to Gemcom, figure 12 shows pit 26 which was imported into Gemcom.

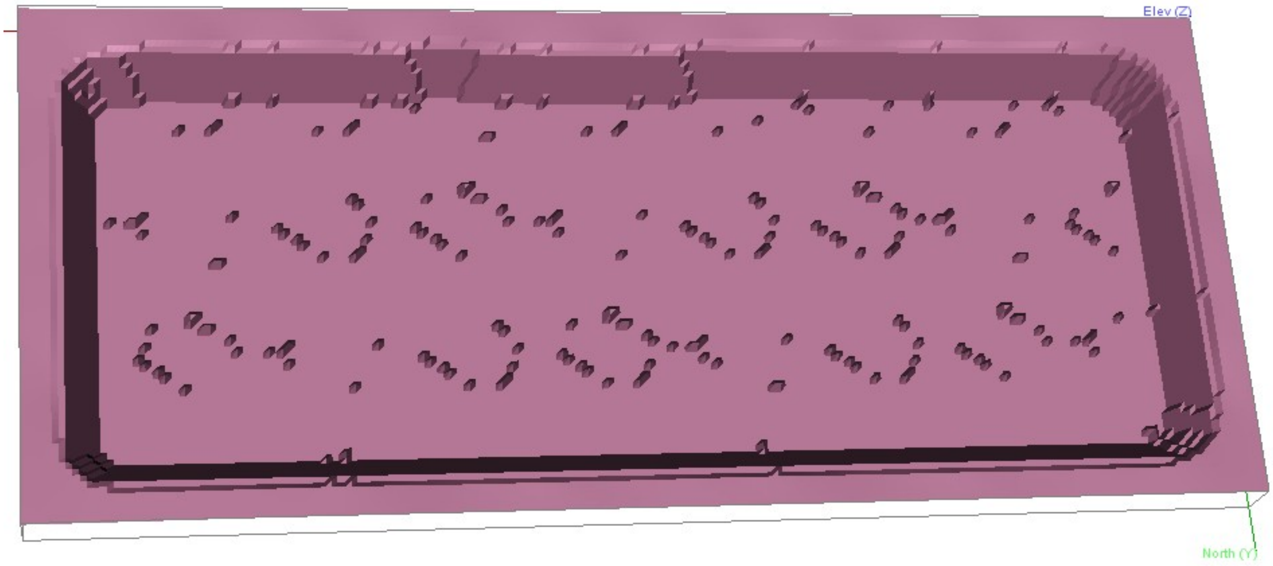


Figure 12- Pit 26 imported from Whittle.

4. Volumetric report

At this stage we created the volumetric report for the pit which we have imported it form Whittle. The volumetric reports provide the information about the amount of the material on a bench-by-bench basis, therefore information such as volume of air, waste and ore, tonnage of waste and ore and quantity of bitumen production can be obtained from the volumetric reports.

Table 6- Volumetric report for pit 26.

Report 1: Incremental

PLANE	EXCAVATION	ROCKGROUP	Volume	Density	Tonnage	%BIT	%BIT_P
			M**3 x 1000	T per M**3	T x 1000	Grade	Product
Lev_1	MINED	AIR	208,917.589	0.000	0.000	0.00	0.0
		ORE	0.000	0.000	0.000	0.00	0.0
		WASTE	659.230	1.800	1,186.614	0.00	0.0
		Total	**209,576.818**	**0.006**	**1,186.614**	**0.00**	**0.0**
	FILLED	AIR	9.522	1.000	9.522	0.00	0.0
		ORE	0.000	0.000	0.000	0.00	0.0
		WASTE	0.000	0.000	0.000	0.00	0.0
		Total	**9.522**	**1.000**	**9.522**	**0.00**	**0.0**
	Total		**209,586.340**	**0.006**	**1,196.136**	**0.00**	**0.0**
Lev_2	MINED	AIR	0.000	0.000	0.000	0.00	0.0
		ORE	53,699.992	2.100	112,769.977	6.54	737,060,418.1
		WASTE	416,420.656	1.800	749,557.161	0.00	0.0
		Total	**470,120.648**	**1.834**	**862,327.138**	**0.85**	**737,060,418.1**
	Total		**470,120.648**	**1.834**	**862,327.138**	**0.85**	**737,060,418.1**
Lev_3	MINED	AIR	0.000	0.000	0.000	0.00	0.0
		ORE	395,174.959	2.100	829,867.377	7.75	6,427,741,098.8
		WASTE	56,287.489	1.800	101,317.478	0.00	0.0
		Total	**451,462.448**	**2.063**	**931,184.854**	**6.90**	**6,427,741,098.8**
	Total		**451,462.448**	**2.063**	**931,184.854**	**6.90**	**6,427,741,098.8**
Lev_4	MINED	AIR	0.000	0.000	0.000	0.00	0.0
		ORE	415,895.96	2.100	873,381	10.4	9,104,47

			8		.493	2	0,190.6
		WASTE	17,255.904	1.800	31,060.627	0.00	0.0
		Total	433,151.872	2.088	904,442.120	10.07	9,104,470,190.6
	Total		433,151.872	2.088	904,442.120	10.07	9,104,470,190.6
Lev_5	MINED	AIR	0.000	0.000	0.000	0.00	0.0
		ORE	413,630.027	2.100	868,623.016	9.29	8,070,784,650.8
		WASTE	1,843.038	1.800	3,317.469	0.00	0.0
		Total	415,473.065	2.099	871,940.485	9.26	8,070,784,650.8
	FILLED	AIR	0.000	0.000	0.000	0.00	0.0
		ORE	0.000	0.000	0.000	0.00	0.0
		WASTE	0.000	0.000	0.000	0.00	0.0
		Total	0.000	0.000	0.000	0.00	0.0
	Total		415,473.065	2.099	871,940.485	9.26	8,070,784,650.8
Lev_6	MINED	AIR	0.000	0.000	0.000	0.00	0.0
		ORE	366,920.288	2.100	770,532.570	8.49	6,544,188,351.7
		WASTE	7,217.041	1.800	12,990.674	0.00	0.0
		Total	374,137.329	2.094	783,523.244	8.35	6,544,188,351.7
	FILLED	AIR	0.000	0.000	0.000	0.00	0.0
		ORE	0.000	0.000	0.000	0.00	0.0
		WASTE	0.000	0.000	0.000	0.00	0.0
		Total	0.000	0.000	0.000	0.00	0.0
	Total		374,137.329	2.094	783,523.244	8.35	6,544,188,351.7
Lev_7	FILLED	AIR	0.000	0.000	0.000	0.00	0.0
		ORE	0.000	0.000	0.000	0.00	0.0
		WASTE	0.000	0.000	0.000	0.00	0.0
		Total	0.000	0.000	0.000	0.00	0.0
	Total		0.000	0.000	0.000	0.00	0.0
Total			2,353,931.703	1.850	4,354,613.978	7.09	30,884,244,709.8

5. Pit design

The final step is to design the imported pit (pit 26). Figure 13 shows the final design of pit 26.

Table 7- Pit parameters.

Bench height (m)	15.0
Berm width (m)	12.0
Batter angle (degree)	17.3
Pit slope angle (degree)	14.0
Ramp start width (m)	20
Ramp grade (%)	8

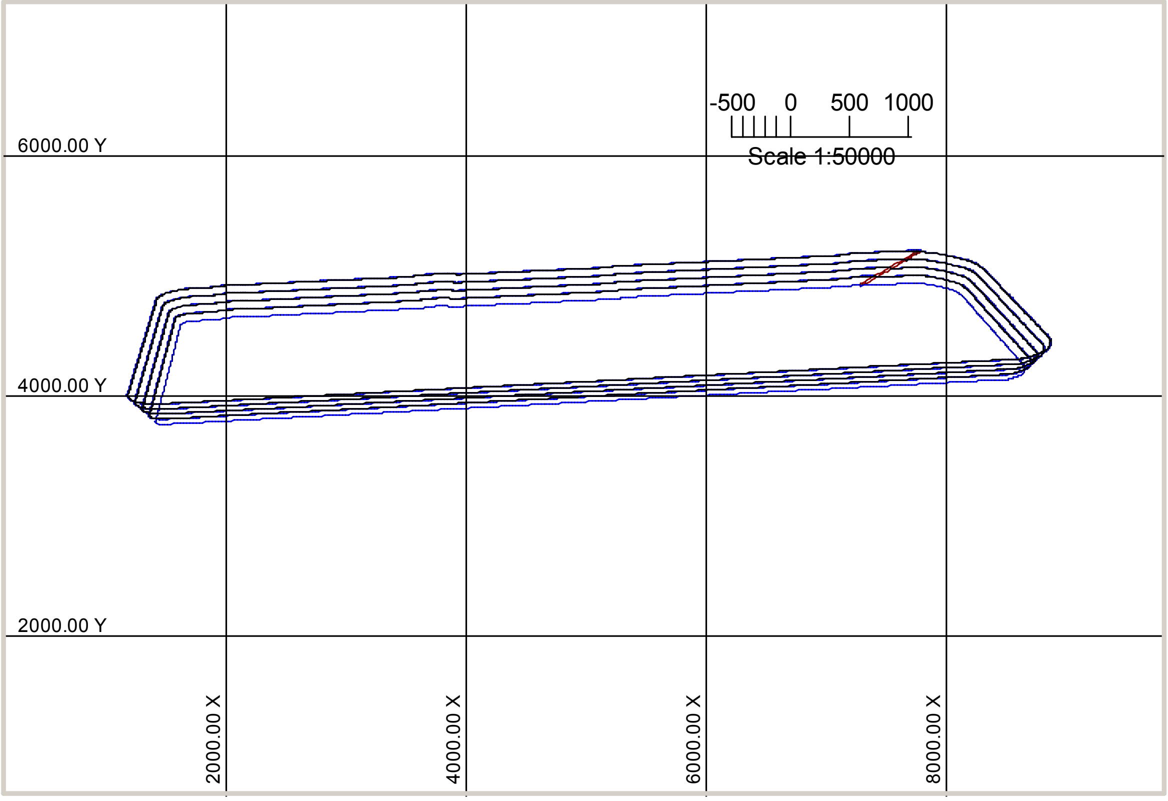


Figure 13- The ultimate pit design.

6. References

[1] Boratynec, D., J, (2003), "Fundamentals of Rapid Dewatering of Composite Tailings", Thesis, © University of Alberta, Edmonton, Alberta, Pages 267.

[2] Gemcome Labs, I. S., (2008). Vancouver, B. C.

[3] Hein, F., J, Cotterill, D., K , and Berhane, H., (2000), "An Atlas of Lithofacies of the McMurray Formation", © Alberta Geological Survey, Edmonton, Alberta, 23-34.

Orebody modeling, optimization and pit design using GEMS and Whittle

Eugene Ben-Awuah

1. Introduction

In mine planning, orebody modeling is a very significant step in the planning process. The model serves as the main backbone that drives the activities of the mine throughout its life. It is used for assessing the cash flow of the entire mining operation and then further exploited in subsequent mine planning processes such as optimization, pit design, production scheduling, waste management, equipment selection and plant design.

The source data for the orebody modeling is the information from the drill holes logged from the exploration of the orebody. It is also required that the modeler understands the basic geology of the formation of the orebody and its country or host rock. The mineral deposit under consideration here is oil sand in the McMurray formation.

2. Oil sand deposit

There are five main soils or rock types associated with this oil sand deposits namely:

1) Muskeg/Peat 2) Pleistocene Unit 3) Clearwater Formation 4) McMurray Formation and 5) Devonian carbonates. The oil bearing rock type is the McMurray formation (MMF) which is also made up of three rock types. These are the Upper McMurray (UKM), the Middle McMurray (MKM) and the Lower McMurray (LKM). The main element of interest is bitumen which exists in various grades across the formation. Details of the five rock types associated with the oil sand deposit are:

1. Muskeg/Peat – this is the topmost overburden material that contains the seeds and roots of native plants and is used for the topmost layer of the reclaimed land. Before mining, this layer is removed and stockpiled and later used for reclamation works.
2. Pleistocene Unit (PLU) and Clearwater Formation (CWF) – The next rock profile is the PLU followed by the CWF. These are considered as waste rocks lying above the bitumen bearing McMurray formation. Materials from these profiles are used for road and dyke construction in the mine depending on the soil properties and its mineral content.
3. McMurray Formation (UKM, MKM, LKM) – the bulk of the bitumen and gas reserves are contained with in the McMurray interval in the oil sand area. The McMurray formation rests with profound unconformity on the Devonian carbonates and is unconformably overlain by the Clearwater formation. The McMurray formation ranges between 0 – 130m from Devonian highs to bitumount basin. The LKM is comprised of gravel, coarse sand, silt and clay with siderite as cement or spherulites within ferrigunous paleosols. The UKM and MKM comprises of micaceous, fine-to-medium-grained sand, silt and clay, with rare

siderite as cement and intraclasts and pyrite nodules up to 10cm in diameter (Hein, Cotterill et al., 2000).

4. Devonian Carbonates (DVN) - This is the rock type which lies beneath the McMurray formation and is made up of numerous limestone outcrops. It marks the end of the oil sand deposit on a vertical profile.

To obtain details of the oil sand deposit, it is required that a detailed exploration programme is undertaken where drilling is carried out and the resulting data logged for further analysis and modeling. Figure 1 shows a sketch of the vertical soil profile of the oil sand deposit.

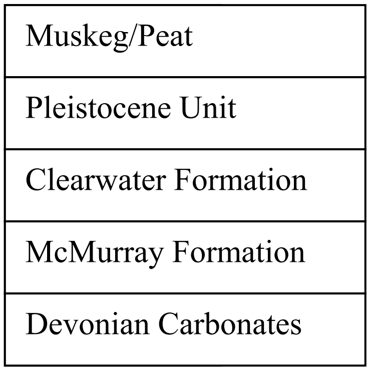


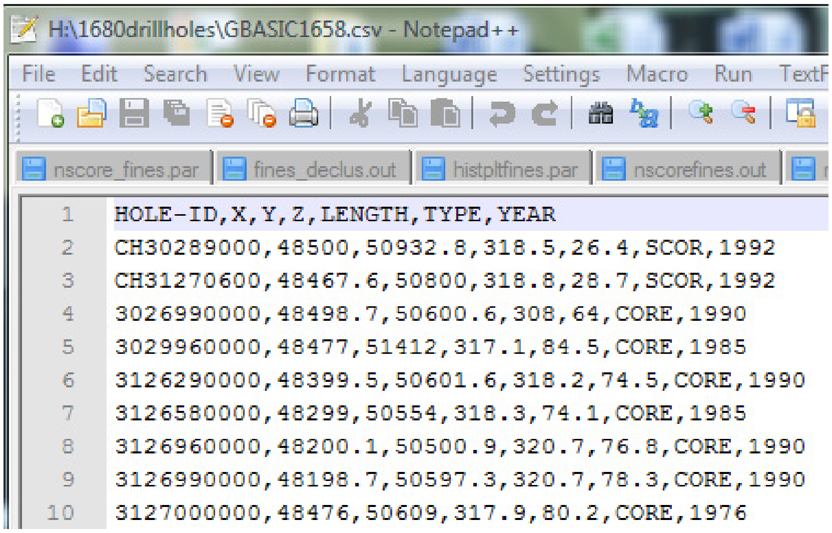
Figure 1: Sketch of the vertical soil profile of the oil sand deposit

3. Drillhole data

The number of drill holes used for this project is 1818. The drillhole data logged from the exploration is generally divided into four basic files and auxiliary files which vary in number. These are saved in different files with 'csv' format which are imported to GEMS for analysis. The basic files are mainly:

- Header file – this contains information on the borehole identification names (hole-id), the 3D coordinates, the length of the borehole, the type and the year drilled.
- Survey file - this contains information on the borehole orientation down the hole. It has the distance from the ground surface, the azimuth and the dip of the hole as it goes down.
- Lithology file – this contains information on the various rock types along the drillhole. It has the hole-id, the 'from', 'to' fields and the rock code representing the rock type within that segment of the drillhole.
- Assay file – this contains information on the grade of the element of interest to us and in our case, its bitumen. This file has the hole-id, the 'from', 'to' fields and the percentage bitumen content for each segment of the drillhole.

Figure 1 is a sample of the header file that was imported into GEMS for analysis. Note that the first line contains a description of the data and all the files contain the hole-id, which is the primary key. You must also know the type of data (real, integer, alphanumeric) and the order of the data.

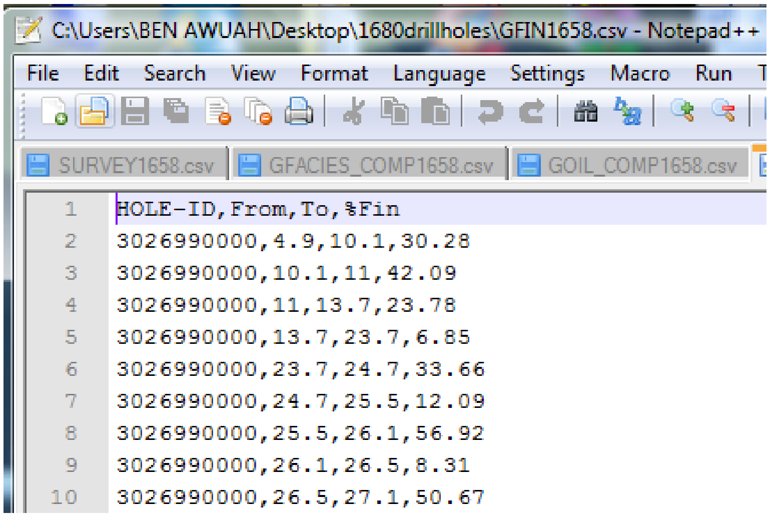

H:\1680drillholes\GBASIC1658.csv - Notepad++
File Edit Search View Format Language Settings Macro Run TextF
nscore_fines.par
fines_declus.out
histpltfines.par
nscorefines.out

```
HOLE-ID,X,Y,Z,LENGTH,TYPE,YEAR
CH30289000,48500,50932.8,318.5,26.4,SCOR,1992
CH31270600,48467.6,50800,318.8,28.7,SCOR,1992
3026990000,48498.7,50600.6,308,64,CORE,1990
3029960000,48477,51412,317.1,84.5,CORE,1985
3126290000,48399.5,50601.6,318.2,74.5,CORE,1990
3126580000,48299,50554,318.3,74.1,CORE,1985
3126960000,48200.1,50500.9,320.7,76.8,CORE,1990
3126990000,48198.7,50597.3,320.7,78.3,CORE,1990
3127000000,48476,50609,317.9,80.2,CORE,1976
```

Figure 2: An example of a header file.

The auxiliary files are:

- Fines file – this contains information on the percentage fines along the borehole depth. This file has the hole-id, the ‘from’, ‘to’ fields and the percentage fines content for each segment of the drillhole.
- Indurated Sediments file – this contains information on the siltstone thickness of the indurated zone in millimeters and the relative siltstone hardness of the interval as observed in the core, ranging from 1 to 6.


C:\Users\BEN AWUAH\Desktop\1680drillholes\GFIN1658.csv - Notepad++
File Edit Search View Format Language Settings Macro Run T
SURVEY1658.csv
GFACIES_COMP1658.csv
GOIL_COMP1658.csv

```
HOLE-ID,From,To,%Fin
3026990000,4.9,10.1,30.28
3026990000,10.1,11,42.09
3026990000,11,13.7,23.78
3026990000,13.7,23.7,6.85
3026990000,23.7,24.7,33.66
3026990000,24.7,25.5,12.09
3026990000,25.5,26.1,56.92
3026990000,26.1,26.5,8.31
3026990000,26.5,27.1,50.67
```

Figure 3: An example of a fines file.

The drillhole data is imported into GEMS drillhole workspace where tables are prepared to store the various file types. Import profiles which map the various drillhole data files to the GEMS drillhole workspace tables are set up and used in importing the drillhole data. The primary key that ties all the tables together is the ‘HOLE-ID’. The data imported into

GEMS is verified to ensure that all the tables contain the appropriate data. To ensure that the drillholes can be viewed on the screen, a drillhole display profile is set up to display the drillholes and the various attributes along the hole depth like lithology and grade. With this display profile, the drillhole data can be loaded and viewed on the screen. Figure 4 shows a view of the drillholes displaying the lithology which comprises of the PLU, CWF, UKM, MKM, LKM and the DVN along the drillhole depth.

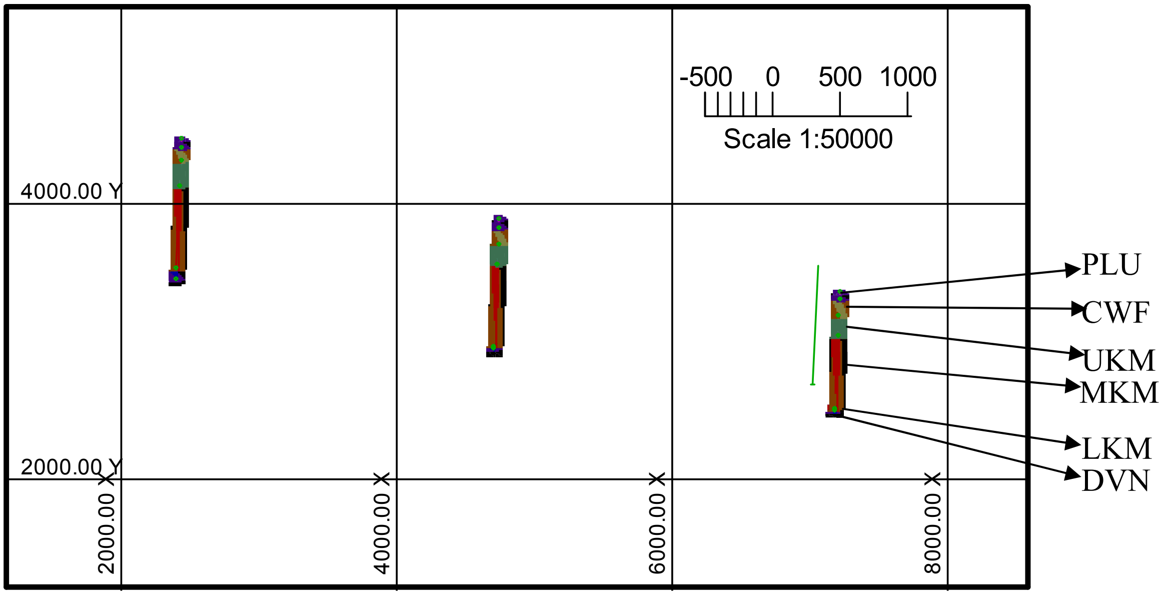


Figure 4: Drillholes displaying the lithology along the hole.

4. Drillhole composites

Drillhole composites enable grades (or any other values) to be combined or averaged over different ranges. The compositing was carried out in three stages. The first stage involves creating composites of similar material (GEMS Lab Instructions, 2008). This is where all data belonging to the same rock type along a borehole are put together resulting in the six main rock types. This is stored in the table labeled 'COMP_LITH' in the GEMS drillhole workspace.

The second stage involves dividing the drillhole into equal lengths within the same rock type. This is to ensure that the composited lengths are not too big to reduce the accuracy of the composited data (grade). In this project, equal lengths of 5m within the same rock type are used. This is stored in the table labeled 'COMP_LITHEQ' in the GEMS drillhole workspace.

The third stage involves using the 'COMP_LITHEQ' table to create a composite of the grades. In other words, using the assay data, grades are composited over the equal length stored in the 'COMP_LITHEQ' table and stored in the table 'COMPGRADE'.

In all these, the composite tables must be prepared to accept the required composite data as well as a composite profile that is used in controlling the compositing processes.

5. Statistics for composites

In other to obtain various statistics on the grade for a grade-depth trend analysis and creating a grade display profile, it is useful to know the range of values in the data set. This would be univariate statistics on the drillhole grades. In GEMS, this is done by extracting the X, Y, Z location, the grade values, the rock codes and the hole-id from the 'COMP_GRADE' table into a point area workspace, 'COMPBITUMEN', created for this purpose. It is always a good idea to look at the extracted data to see what the information looks like and to verify that you have in fact extracted the data you wanted (GEMS Lab Instructions, 2008).

Univariate statistical analysis can be done on the extracted data and the range of the composited bitumen grades in the various rock types can be used in setting a display profile for the composited data. Further statistics such as histograms for bitumen in the various rock types as well the grade-depth trend analysis is carried out. Figure 5 is a graph showing the bitumen grade variation with depth. Figure 6 is a histogram for bitumen grade content of entire deposit. Figure 7, 8 and 9 are the histograms for bitumen grade content of Upper, Lower and Middle McMurray formations respectively.

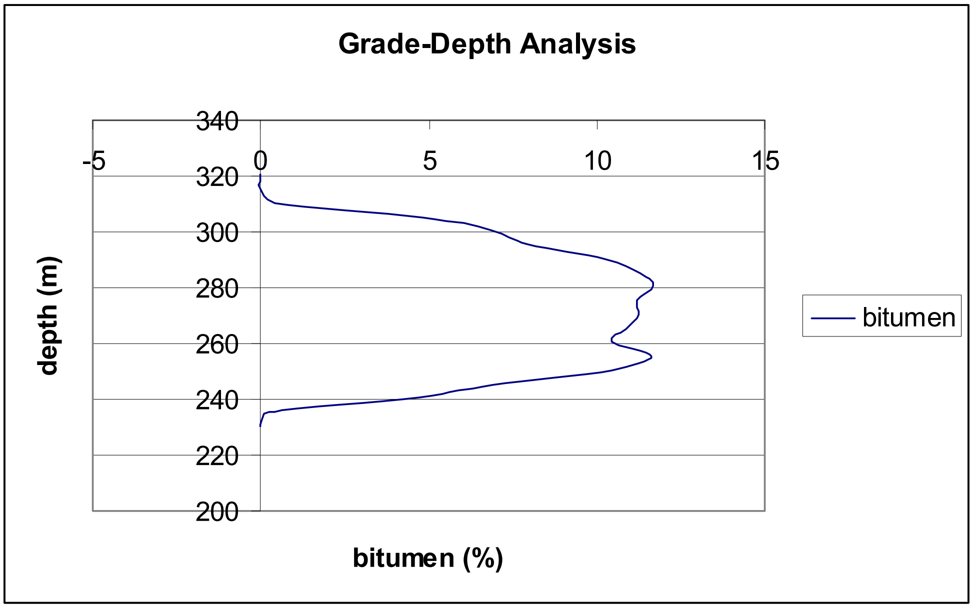


Figure 5: Bitumen grade variation with depth (datum on ground surface = 325.4m)

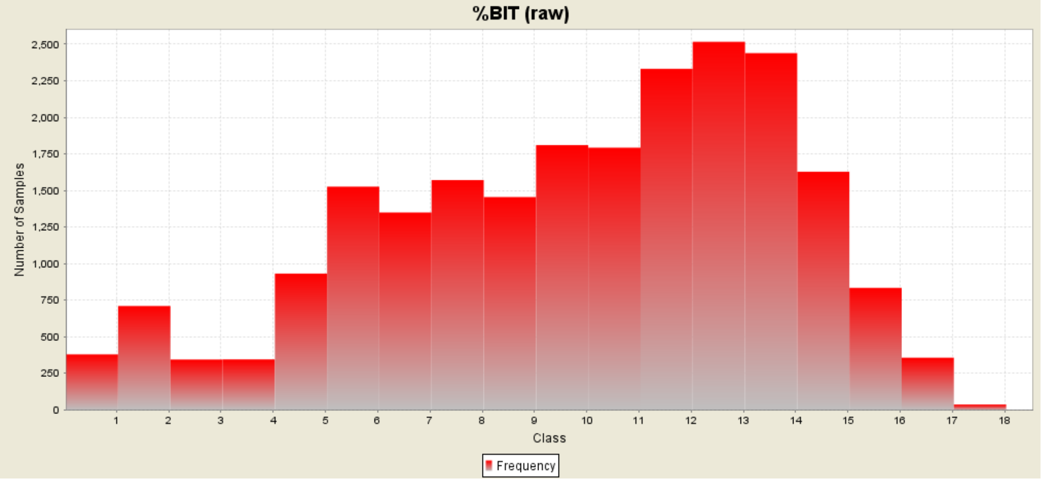


Figure 6: Histogram for bitumen grade content of deposit

Table 1: Descriptive statistics for bitumen grade content of deposit

Number of Samples	Mean (%)	Variance $(\%)^2$	Standard deviation (%)	Median (%)
22355	9.82	14.81	3.84	10.46

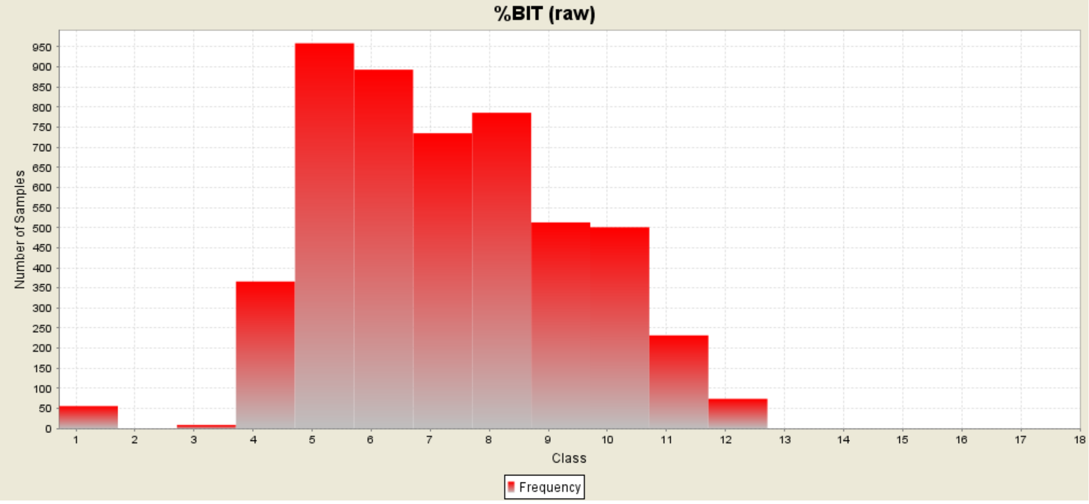


Figure 7: Histogram for bitumen grade content of Upper McMurray formation

Table 2: Descriptive statistics bitumen grade content of Upper McMurray formation

Number of Samples	Mean (%)	Variance $(\%)^2$	Standard deviation (%)	Median (%)
5124	7.28	4.41	2.10	7.07

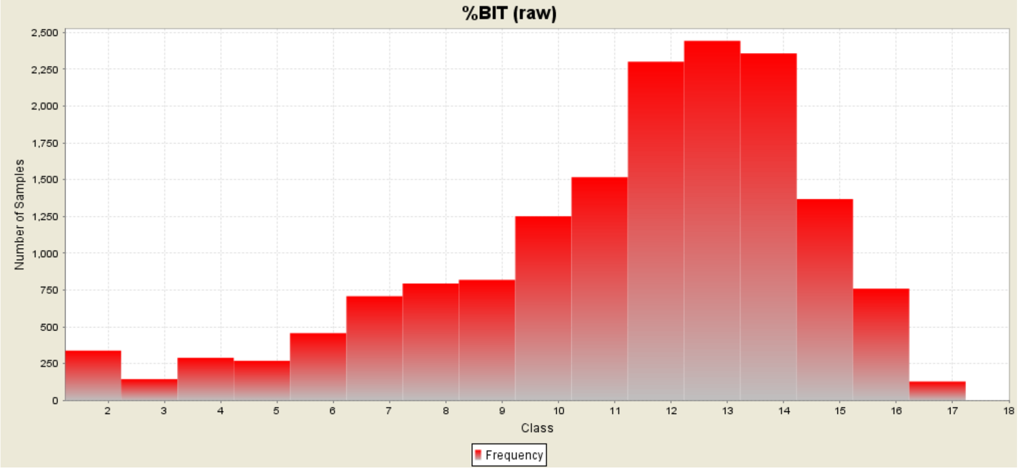


Figure 8: Histogram for bitumen grade content of Middle McMurray formation

Table 3: Descriptive statistics for bitumen grade content of Middle McMurray formation

Number of Samples	Mean (%)	Variance $(\%)^2$	Standard deviation (%)	Median (%)
15941	11.13	10.69	3.27	11.91

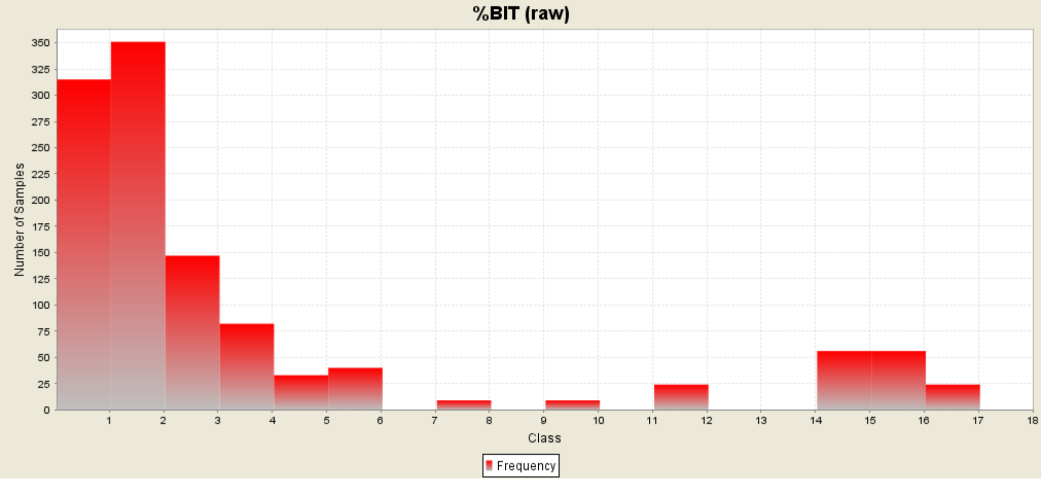


Figure 9: Histogram for bitumen grade content of Lower McMurray formation

Table 4: Descriptive statistics for bitumen grade content of Lower McMurray formation

Number of Samples	Mean (%)	Variance $(\%)^2$	Standard deviation (%)	Median (%)
1146	3.66	22.08	4.70	1.81

6. Creating solids for rock types

After compositing and analyzing the drillhole data, solids are now created to represent the various rock types. Since oil sand deposits are layered in nature, seam modeling approach is used. In seam modeling in GEMS, surfaces representing the top and bottom boundaries of the rock type are created. These boundaries are used to create a solid between the two surfaces and this becomes the solid seam model representing a particular rock type. The solid models are created for PLU, CWF, UKM, MKM and LKM, which are the various rock types to be excavated. Properties of the rock type are attached to the solid model in the form of attributes such as waste, ore and density. Figure 10 is the solid model created for UKM.

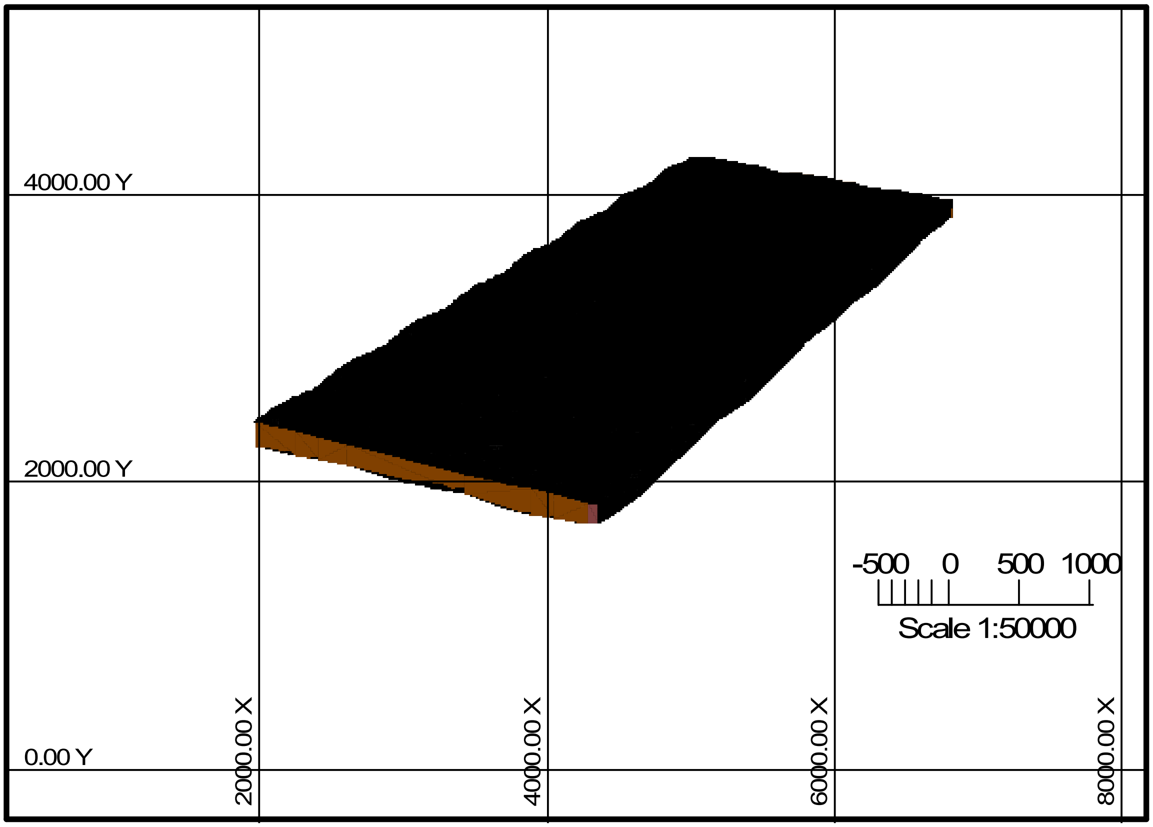


Figure 10: Solid model for Upper McMurray formation

PLU and CWF are modeled as solids which are waste with density 1.8tons/m^3. UKM, MKM and LKM are modeled as solids which are ore with density 2.1tons/m^3.

7. Block model

Block models are three-dimensional arrays of rectangular blocks used to model ore bodies. Each attribute in a block model represents one characteristic of the volume of material. These include but are not limited to rock types, grades, density and dollar value. These attributes are stored in individual models, which have the same number of rows, columns and levels. The models can be used in reserve estimation and mine planning (GEMS Lab Instructions, 2008).

The block model used in modeling the oil sand deposit has a size of 195 x 85 x 7 which yields 116,025 cells or blocks with dimensions of 50 x 50 x 15. The cell dimensions are chosen based on parameters such as ore grade variability and mining bench height. This is called Block Model 2 in this project. A block model cell display profile is setup to be able to view the different attributes of the model with the appropriate colors.

7.1. Rock type model

The first attribute to be transferred to the block model is the rock type model. To do this, we first initialize the entire block model to waste, in our case, the Devonian Carbonates. Then we load the topographical surface of the model area. All blocks above the topographical surface is set to air. The solid rock type model is then loaded one at a time and the block model is updated from the solids. This results in a block model with all the rock type attributes for waste (PLU, CWF, DVN) and ore (UKM, MKM, LKM).

7.2. Density model

With the rock profiles and rock codes already in the block model, we update the density attribute from the rock type model. This will assign 0 ton/m^3 to air blocks, 1.8 tons/m^3 to waste blocks and 2.1 tons/m^3 to ore blocks.

7.3. Grade model

The grade model may be calculated either using simple sample-distance methods (Inverse Distance) or Geostatistical techniques. In this project, a grade estimate is made using the inverse distance squared method. To implement this, a point area workspace is created and the composited grades are extracted and cross referenced with the composited lithology and called ‘BitArea’. An interpolation profile is set up for the Inverse Distance Square (IDS) method selecting the ore bearing rock types for bitumen and using the ‘BitArea’ as the source data for interpolation within the rock types. A search radius of 2000m in X, Y, Z directions was used for interpolation. A similar profile is set up to interpolate the fines grade, the siltstone hardness and thickness with the appropriate rock type model.

A cross-section of the block model showing the color profile of the lithology is shown in figure 11.

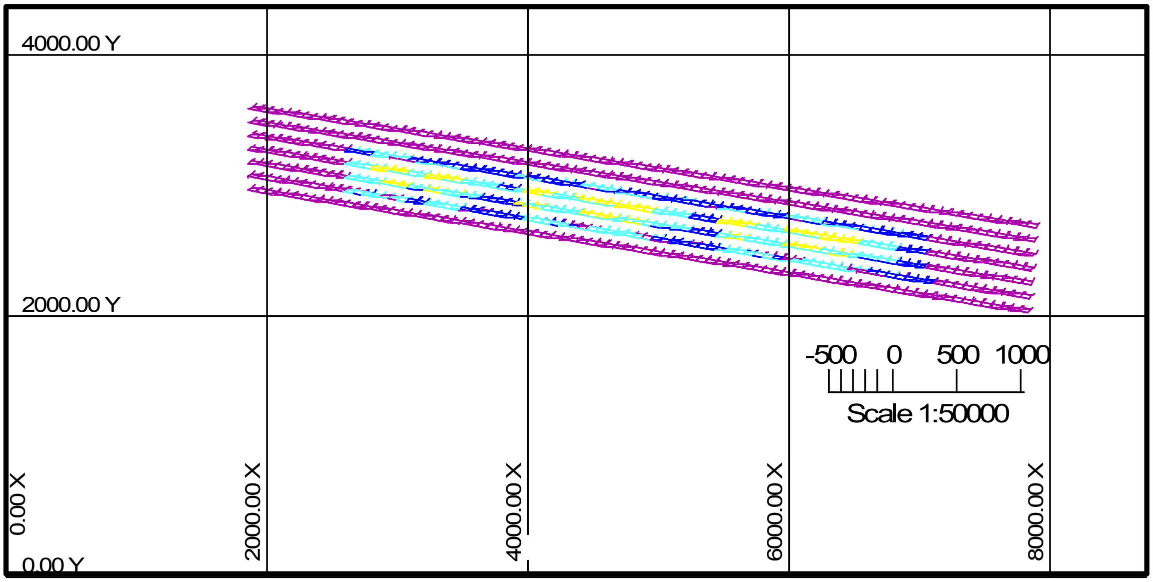


Figure 11: A cross-section of the block model showing lithology

8. Pit optimization

After the block model is completed, we proceed to pit optimization to determine the ultimate pit limit that maximizes the net present value (NPV). The ultimate pit limit can be defined as the determination of the final mining limits of a mineral deposit within certain set constraints such that maximum profitability will be derived from the mining process (Hartman, 1992). This is basically done by maximizing the difference between the revenue generated from mining the desirable mineral and the cost of mining the waste material associated with the desirable mineral.

The pit optimization is done with Whittle Four X. The block model is exported from GEMS to Whittle Four X and the economic parameters for pit optimization are set up as follows:

Table 5: Economic parameters for pit optimization

Economic Parameter	**Figure**
Mining Recovery Cost	$4.60
Mining Recovery	$0.88
Processing Cost	$5.03
Processing Recovery	0.95
Selling Price ($/Bit %mass)	$2.81
Annual Production Target	120,000,000 tons

The technical parameter for the pit optimization is an overall pit slope 14^{o}. After optimization, Whittle pit number 26 has the maximum NPV of $5,127,186,148 which becomes our ultimate pit limit to be used for the pit design. It has a mine life of 38.05 years.

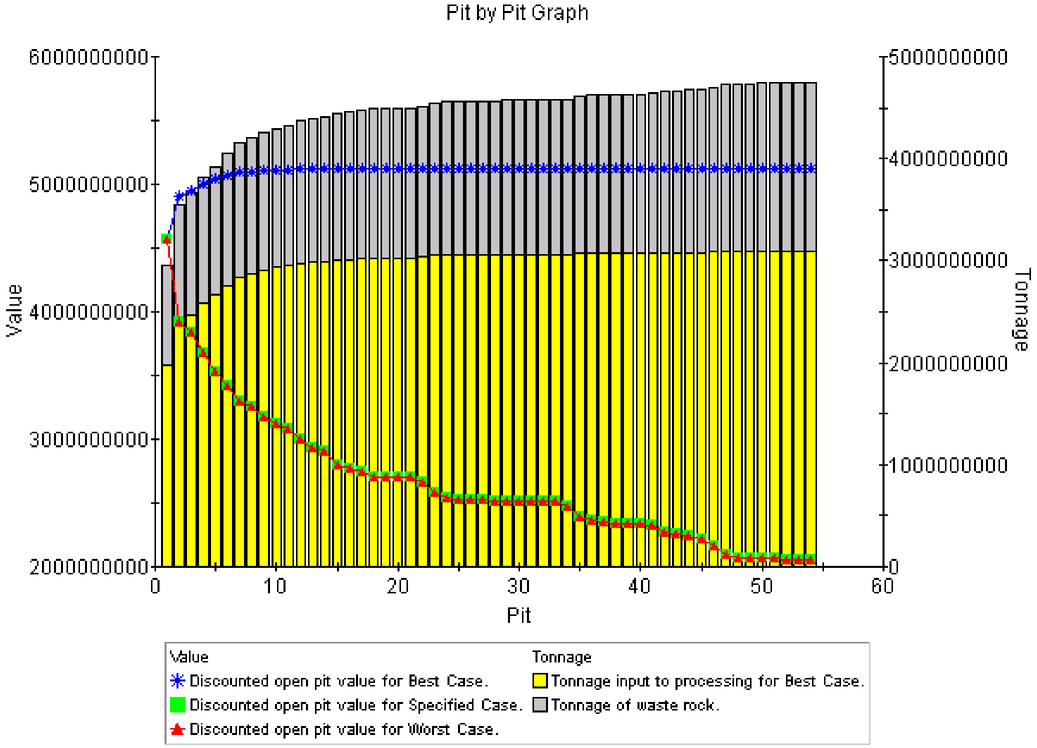


Figure 12: Pit by Pit NPV

The tonnage of ore to be mined is 3,057 million tonnes and the tonnage of waste is 1,508 million tonnes making a total of 4,566 million tonnes of material. The average grade of bitumen to be mined for this pit is 8.93% with a pit utilization of 83.6%. The waste to ore ratio is 0.49. Figure 13 shows the annual ore tonnage produced for processing and waste tonnage produced for the waste dump.

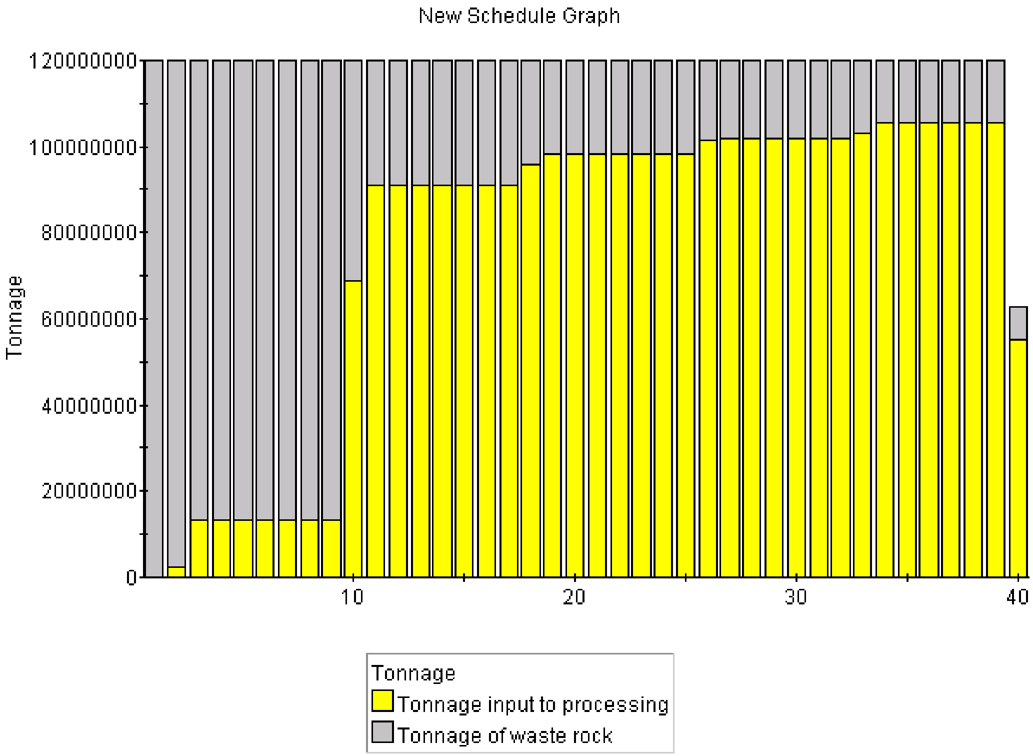


Figure 13: Annual ore and waste tonnage produced

The ultimate pit limit is now imported from Whittle to GEMS. This is to enable us visually analyze it to see the extent to which it contains the orebody and use it as a basis for the pit design. Figure 14 shows a cross section of the ultimate pit limit and how it intersects the block model. Figure 15 shows the ultimate pit limit after optimization.

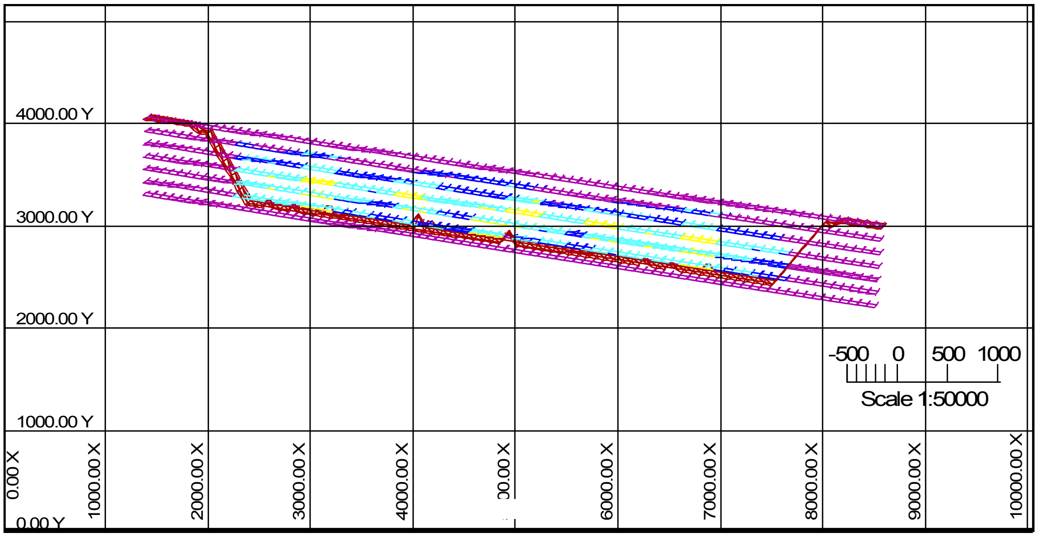


Figure 14: Cross section of pit and block model

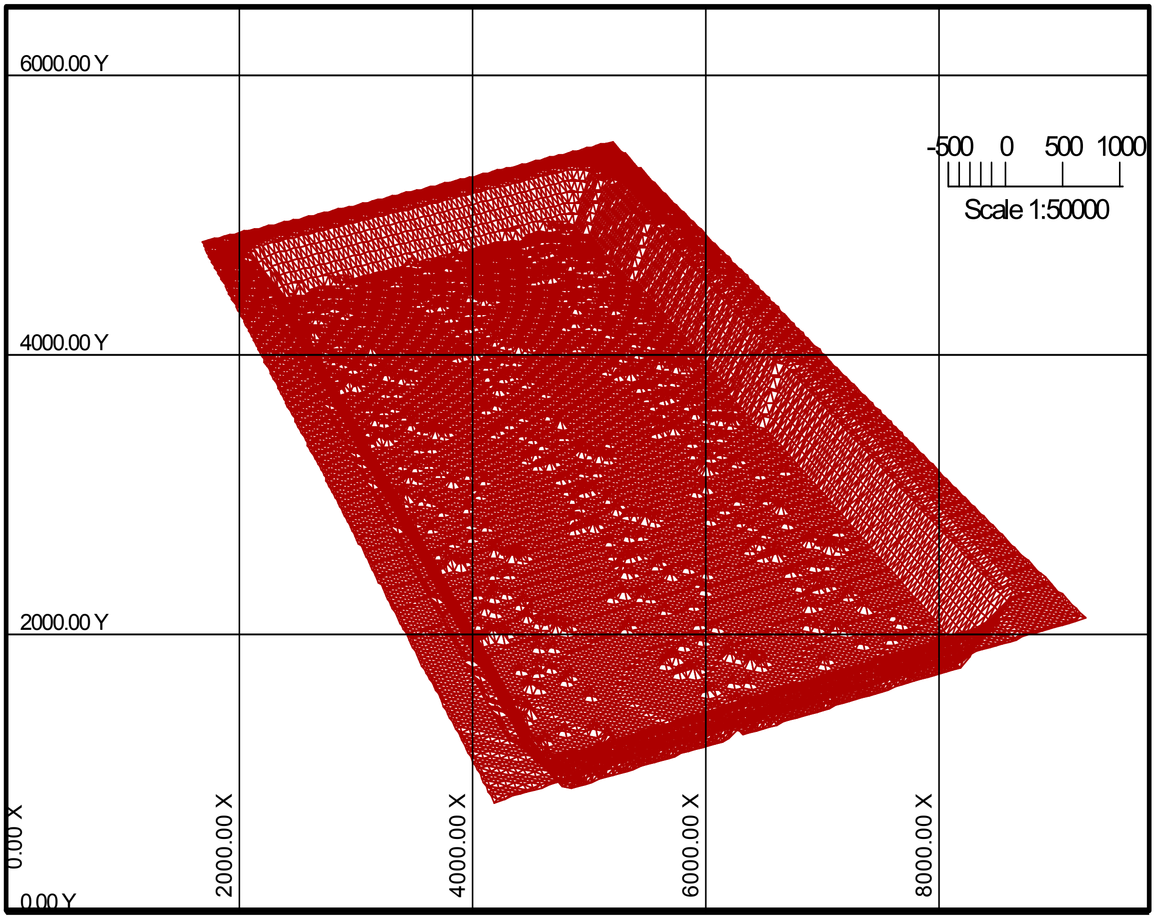


Figure 15: Ultimate pit limit

9. Pit volumetric report

In GEMS, through a combination of block models and surfaces, the economic value of the pit can be assessed. In addition to the total quantities of material, the amount of material on a bench-by-bench basis can be obtained (GEMS Lab Instructions, 2008). It is very convenient that the block model dimension in the Z direction corresponds to the proposed bench height of 15m. In this model, we will have 7 benches starting from a toe elevation of 315.699m on bench one to a toe elevation of 225.699m on bench seven. Plan views are created for each of the benches to enable us generate volumetric reports of the material characteristics of each bench.

After the volumetric report profile is set up, the bench-by-bench volumes, tonnage, economic, grade and grade-tonnage data can be reported. Table 6 is the bench-by-bench statistics reported from the block model and the ultimate pit limit which compares well with the tonnages obtained from Whittle Four X.

Table 6: Bench-by-bench statistics reported from the block model and ultimate pit limit

PLANE	ROCK	Volume	Density	Tonnage	%BIT	%BIT_P
	GROUP	M**3 x 1000	T per M**3	T x 1000	Grade	Product
Lev_1	AIR	208,917.589	0.000	0.000	0.00	0.0
	ORE	0.000	0.000	0.000	0.00	0.0
	WASTE	659.230	1.800	1,186.614	0.00	0.0
	Total	**209,576.818**	**0.006**	**1,186.614**	**0.00**	**0.0**
Lev_2	AIR	0.000	0.000	0.000	0.00	0.0
	ORE	53,699.992	2.100	112,769.977	6.54	737,060,418.1
	WASTE	416,420.656	1.800	749,557.161	0.00	0.0
	Total	**470,120.648**	**1.834**	**862,327.138**	**0.85**	**737,060,418.1**
Lev_3	AIR	0.000	0.000	0.000	0.00	0.0
	ORE	395,174.959	2.100	829,867.377	7.75	6,427,741,098.8
	WASTE	56,287.489	1.800	101,317.478	0.00	0.0
	Total	**451,462.448**	**2.063**	**931,184.854**	**6.90**	**6,427,741,098.8**
Lev_4	AIR	0.000	0.000	0.000	0.00	0.0
	ORE	415,895.968	2.100	873,381.493	10.42	9,104,470,190.6
	WASTE	17,255.904	1.800	31,060.627	0.00	0.0
	Total	**433,151.872**	**2.088**	**904,442.120**	**10.07**	**9,104,470,190.6**
Lev_5	AIR	0.000	0.000	0.000	0.00	0.0
	ORE	413,630.027	2.100	868,623.016	9.29	8,070,784,650.8
	WASTE	1,843.038	1.800	3,317.469	0.00	0.0
	Total	**415,473.065**	**2.099**	**871,940.485**	**9.26**	**8,070,784,650.8**
Lev_6	AIR	0.000	0.000	0.000	0.00	0.0
	ORE	366,920.288	2.100	770,532.570	8.49	6,544,188,351.7
	WASTE	7,217.041	1.800	12,990.674	0.00	0.0
	Total	**374,137.329**	**2.094**	**783,523.244**	**8.35**	**6,544,188,351.7**
Lev_7	AIR	0.000	0.000	0.000	0.00	0.0
	ORE	0.000	0.000	0.000	0.00	0.0
	WASTE	0.000	0.000	0.000	0.00	0.0
	Total	**0.000**	**0.000**	**0.000**	**0.00**	**0.0**
Total		**2,353,931.703**	**1.850**	**4,354,613.978**	**7.09**	**30,884,244,709.8**

10. Pit design

Pit design basically involves a systematic approach in deciding how an orebody will be mined from the surface until the ultimate pit limit (UPL) is reached. The Whittle optimized pit shell is used as the UPL and a bottom to top pit is generated around the Whittle optimized pit shell. When creating a bottom to top pit, the lowest plan view that you want to start the design on is selected and this level should allow a large enough area for mining equipment maneuverability. The design parameters used for the pit are in table 7.

Table 7: Pit design parameters

Pit Design Parameter	**Value**
Pit slope	14^o
Bench width	60m
Berm width	12m
Batter angle	17.3^o
Bench height	15m
Main haul ramp width	50m
Main haul ramp grade	8%

The final pit design has six benches with the ramp on the eastern wall. Figure 12 shows the outlines of the final design showing the toes, crest and ramp.

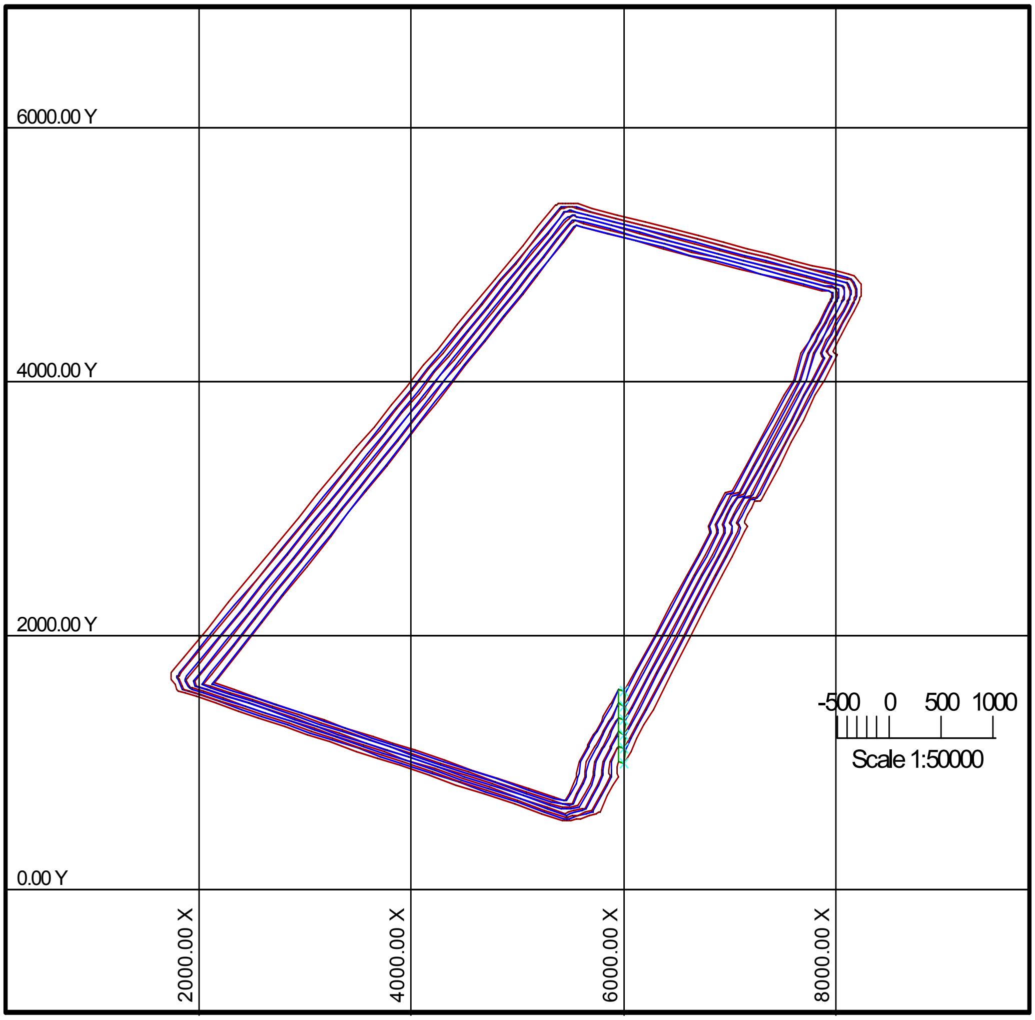


Figure 12: Final pit design

11. Conclusions

Orebody modeling, optimization and pit design are the most important processes involved in mine planning. They serve as the pivot about which both long and short term mine planning revolves. These models are also used in assessing the viability of the mining project and the extent to which the project is profitable.

These baseline data and models for the mining operation requires critical and analytical thinking and should therefore be carried out by highly skilled and experienced personnel to ensure the successful implementation of the project since mining operations are always capital intensive.

12. References

[1] Hartman, H. L., (1992), "SME Mining Engineering Handbook", © Society for Mining, Metallurgy, and Exploration, Retrieved from: books.google.ca/books?isbn=0873351002

[2] Hein, F. J., Cotterill, D. K. and Berhane, H., (2000), "An Atlas of Lithofacies of the McMurray Formation, Athabasca Oil Sands Deposit, Northeastern Alberta: Surface and Subsurface", © Alberta Geological Survey, Edmonton, Alberta, 07, June, 217.

[3] GEMCOM Software International, (2008), “GEMS Lab Instructions”, Vancouver, British Columbia, pp 54.

Simulation of a crushing process

Eugene Ben-Awuah [1] and Hooman Askari-Nasab

Abstract

Rock Fragmentation in the mining industry is one of the major processes used in the extraction of the mineral content from the gangue material associated with the mineral. This process is known as Comminution and it basically involves blasting, crushing and grinding to liberate the associated mineral from the waste. Monte Carlo Simulation of crushing process is an effective way to determine the various design parameters that would affect the crushing process. In this project, a simulation model of a crushing process is built from a mathematical model using Simulink and Matlab. The range of input rock sizes to the crusher is captured by a normal distribution. The material fed to the crusher is fragmented to smaller size after crushing process. Rock fragmentation is assumed to be function of crushing energy used. We used the results of the simulation study to establish the relationship between the crushing energy and the product particle size distribution for a given feed particle size distribution. In conclusion, the crushing energy in a crushing process can be varied appropriately to obtain a desired fragmented product particle size.

1. Introduction

Mineral deposits normally exist in massive forms referred to as rocks with different types of properties. These properties range from rock strength, density, permeability and many others. In mining operations, the massive rock deposit has to be reduced in size consecutively to enable hauling from the pit to the crushing plant where it is further reduced by the crushing process to desirable sizes. The size from the crushing operation is required by the downstream process to be able to liberate the desired mineral from its associated gangue material. This project will seek to simulate a crushing process and analyze the effect of different input energy used in crushing a given feed particle size of rocks to a corresponding product particle size.

1.1. Comminution

Most minerals are intimately associated with the waste material and must be initially liberated before separation can take place. This is achieved by comminution. Comminution is the process whereby particle size of a mineral deposit is continuously reduced until smaller fragments, which enables the mineral to be separated from the gangue by such methods as are available (Wills and Napier-Munn, 2006). Comminution in general involves the process of blasting, crushing and grinding of the ore into desirable fragments for onward processing.

[1] PhD student, School of Mining, University of Alberta

1.2. Blasting

Rock blasting is the process of fragmenting rocks into smaller desirable sizes with the controlled use of explosives. This is the technique mostly used in mine operations because most rocks must be blasted before excavation. Blasting is one of the most important aspects of mine operations and if it is not done properly, further development of the mine becomes a problem. Some of the main factors that affect blasting results are the properties of the explosives being used, the blasting round initiation sequencing, the blast pattern, the rock structure and other properties (Kennedy, 1990).

There are two main types of blasting operations carried out in most mines and these are: 1) Primary blasting and 2) Secondary blasting. Primary blasting is intended to fragment the insitu rock to sizes required by the crushing plant while secondary blasting reduces the boulders resulting from the primary blasting into smaller fragments of desired sizes (Ben-Awuah and Akayuli, 2008). The size of rocks that results from primary blasting is normally the specification used in building the crushing plant. Therefore it is required that if due to variability in rock formation the designed blast does not yield the required fragments, then secondary blasting must be carried out to reduce oversize fragments before hauling the material to the crushing plant for crushing.

1.3. Crushing

Crushing is the process whereby particulate fragments are reduced by grinding to the product sizes required for downstream processing. This is to ensure that valuable constituents are physically liberated from gangue material before separations are attempted (Fuerstenau and Han, 2003). Crushing is achieved by compression of the ore against rigid surfaces or by impact against surfaces in a rigidly constrained motion path. Usually, crushing is a dry process undertaken at different stages with specified reduction ratios which ranges from three to six at each stage. Reduction ratio in a crushing process can be defined as the ratio of maximum particle size entering to maximum particle size leaving the crusher (Wills and Napier-Munn, 2006).

1.3.1 Principles of crushing

Most ores are made up of crystalline materials which have their atoms regularly arranged in three dimensional arrays. The size and type of physical and chemical bonds holding the atoms together determines their configuration. In the crystalline lattice of minerals, the bonds between the atoms are over small distances and when extended by a tensile stress results in the breaking of these bonds. These tensile stresses can be achieved by compressive or tensile loading (Wills and Napier-Munn, 2006).

During the crushing process rocks are loaded uniformly, however the internal stresses are not evenly distributed resulting from the fact that the rock is made up of different minerals dispersed as grains of various sizes. The stress distribution depends on the mechanical properties of the individual minerals as well as the presence of cracks in the matrix. Cracks act as areas for stress concentration. From Wills and Napier-Munn (2006), it can be shown that the increase in stress at such an area is proportional to the square root of the crack length perpendicular to the stress direction. Therefore for any particular level of stress, there is a critical value for the crack length at which any increase in stress level will cause the atomic bond at the crack tip to break causing further crack. This increase in crack length causes further increase in stress concentration at the crack tip causing a rapid propagation of the crack through the matrix. Fracture eventually occurs (Wills and Napier-Munn, 2006). It has also been stated by Fuerstenau and Han (2003), that the amount of breakage in a matrix of particles depend on how energy is dissipated in the matrix and its distribution into the various types. Figure 1 shows a rock fracture by crushing.

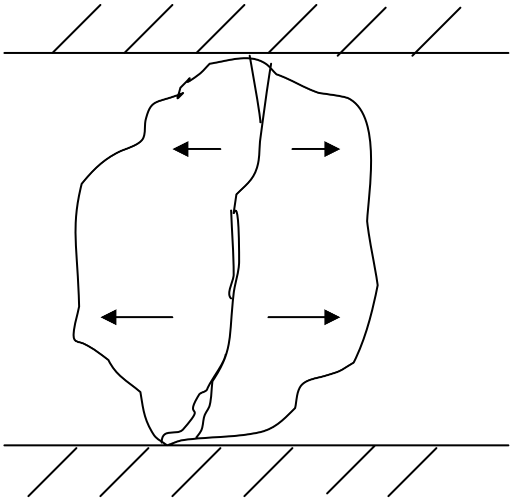


Figure 1: Rock fracture by crushing
(Wills and Napier-Munn, 2006)

2. Problem definition

A rocky material from an iron ore deposit is to be crushed into smaller fragments after the material has been hauled from the pit to the crusher. The blasted material from the pit has a particle size distribution represented by a normal distribution with a mean of 1000 mm and a variance of 300 mm. The expected particle size distribution of the product from the crushing process is a normal distribution with a mean of 203 mm and a variance of 5 mm. A simulation is to be carried out to obtain the corresponding input energy required in the crushing process to obtain our desirable product particle size.

3. Theoretical framework and models

3.1. Crushing theory

This theory is about the relationship between energy input and the product particle size that results from a given feed size. These theories have their limitations resulting from the fact that majority of the energy input to a crushing plant is absorbed by the plant itself and very little energy is left for breaking the material. Another limitation is that, the theories of crushing assume the material to be brittle which means no energy is adsorbed during the crushing process. However, in reality some materials have plastic properties which enable them to consume energy in changing shape which may not create a significant new surface equivalent to the amount of energy consumed (Wills and Napier-Munn, 2006).

The oldest of these theories is that of Von Rittinger (1867). *This states that the energy consumed during size reduction is proportional to the area of the new surface produced. The surface area of a known weight of particles of uniform diameter is inversely proportional to the diameter of the particle* (Wills and Napier-Munn, 2006). This theory applies fairly well in the fine grinding range of 10-1000µm. Rittinger's equation is shown by equation (1):

$$E = K\left(\frac{1}{D_2} - \frac{1}{D_1}\right) \tag{1}$$

where E = input energy (kWh/t)

D_1 = initial particle size (µm)

D_2 = final particle size (µm)

K = a constant

Another theory is that of Kick (1885), which states that the work required in breaking a particle is proportional to the reduction in volume of the particle involved (Wills and Napier-Munn, 2006). Let *f* be the diameter of the feed particles and *p* be the diameter of the product particles, then the reduction ratio *R* will be f/p. This theory is reasonably accurate in the crushing range above 1 cm in diameter. From Kick's law, equation (2) was developed:

$$E\alpha\frac{\log R}{\log 2} \tag{2}$$

where E = input energy (kWh/t)

In 1952, Bond also developed an equation based on the principle that the work input is proportional to the new crack length produced at the crack tip during particle breakage. This also equals the work represented by the product minus that represented by the feed (Wills and Napier-Munn, 2006). This theory is reasonably accurate in the range of conventional ball-mill grinding.

Bond's operating work index equation is shown by equation (3) (Kawatra, 2006):

$$W_i = \frac{W}{10\left(\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{F}}\right)} \tag{3}$$

where W = specific energy (kWh/t)

W_i = operating work index (kWh/t)

P = 80% passing 100 μm size for the product (μm)

F = 80% passing 100μm size for the feed (μm)

The work index is the crushing parameter, which expresses the difficulty in crushing and grinding a given material. Numerically, it is the kilowatt hours per short ton required to reduce the material from an infinite feed size to 80% passing 100μm (Wills and Napier-Munn, 2006). Table 1 shows Bond's work indices for some selected materials.

Table 1: Bond's work indices for some materials (Wills and Napier-Munn, 2006)

Material	Work Index (kWh/t)	Material	Work Index (kWh/t)
Barite	4.73	Fluorspar	8.91
Bauxite	8.78	Granite	15.13
Coal	13.00	Graphite	43.56
Dolomite	11.27	Limestone	12.74
Emery	56.70	Quartzite	9.58
Ferro-silicon	10.01	Quartz	13.57

4. Methodology

The objective of this project is to use a mathematical model of a crushing process to build a simulation model to simulate a crushing process. This objective will be achieved by:

- Building a simulation model from the mathematical model of a crushing process using Simulink and Matlab.
- Performing sensitivity analysis on the crushing energy required for the crushing process in the model.

5. Implementation

5.1. Simulation of the crushing process using Simulink

The mathematical model for the crushing process can be simulated using a Monte Carlo Simulation method. This is a method where iterative process is used to evaluate a deterministic model using sets of random numbers as inputs. This method is usually used in simulating complex nonlinear models which may involve more than just a few uncertain parameters (Wittwer, 2004).

The crushing model to be simulated is Bond's operating work index equation shown by equation (3). The simulation is carried out to find the relationship between the product particle size from a crushing process and the energy used in crushing a given feed particle size of material.

5.2. Simulating the product particle size and the energy used in crushing a given feed particle size of material

Since the deposit is an Iron ore, Bond's operating work index, W_i, chosen from Table 1 is 10.01kWh/t. The feed particle size is chosen from a normal distribution with a mean of 1000mm and a variance of 300mm. The energy parameter, W, used in the simulation ranges from 3 kWh/t to 6 kWh/t. This range of energy was chosen because of the required product particle size which is a normal distribution with a mean of 203mm and a variance of 5mm. This is the required particle size distribution for the downstream process after the crushing circuit. The Bond's operating work index equation can be written in terms of the product particle size as shown by equation (4).

$$P = \left(\frac{10W_i\sqrt{F}}{W\sqrt{F} + 10W_i} \right)^2 \tag{4}$$

The simulation model for this equation was built using Simulink and can be seen in figure 2. It was however, executed from Matlab and it was simulated for 100 seconds for 10000 multiple runs and the results are summarized in Appendix 1. The Matlab code used in executing the simulation is also shown in Appendix 2 attached.

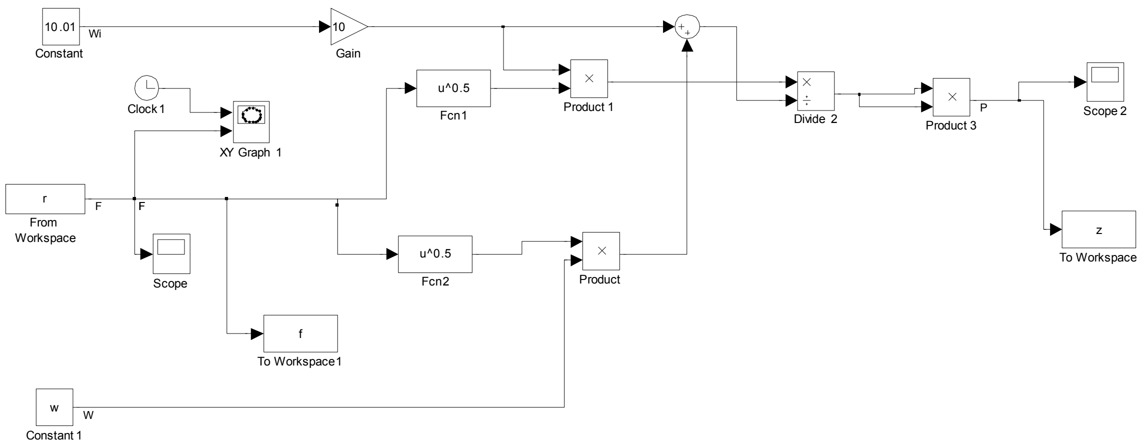


Figure 2: Simulation model for product particle size from Bond's equation

Figure 3 shows the distribution of the feed particle size that went into the simulation. This was obtained after the compilation of the mean of the multiple runs over the simulation time. It shows how erratic the feed particle size was before the crushing process. Figure 4 also shows the resulting product particle size obtained for the given feed particle size with varying crushing energy. The varying energy input was defined in Matlab to run with the Simulink model.

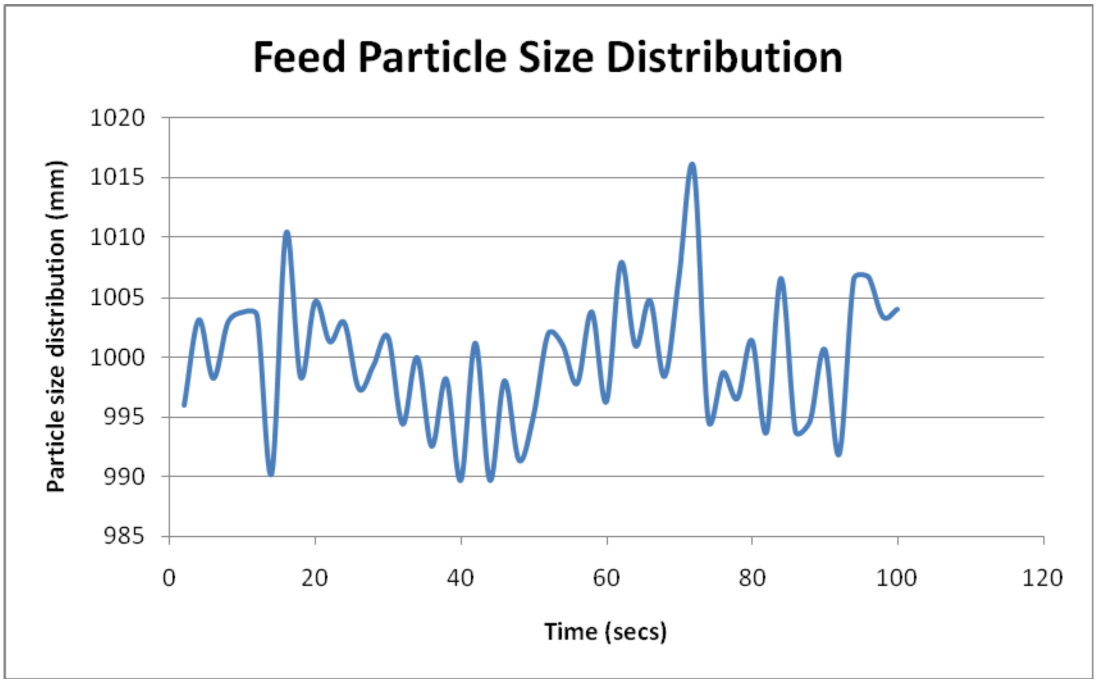


Figure 3: Particle size distribution of feed (mm) over the simulation time

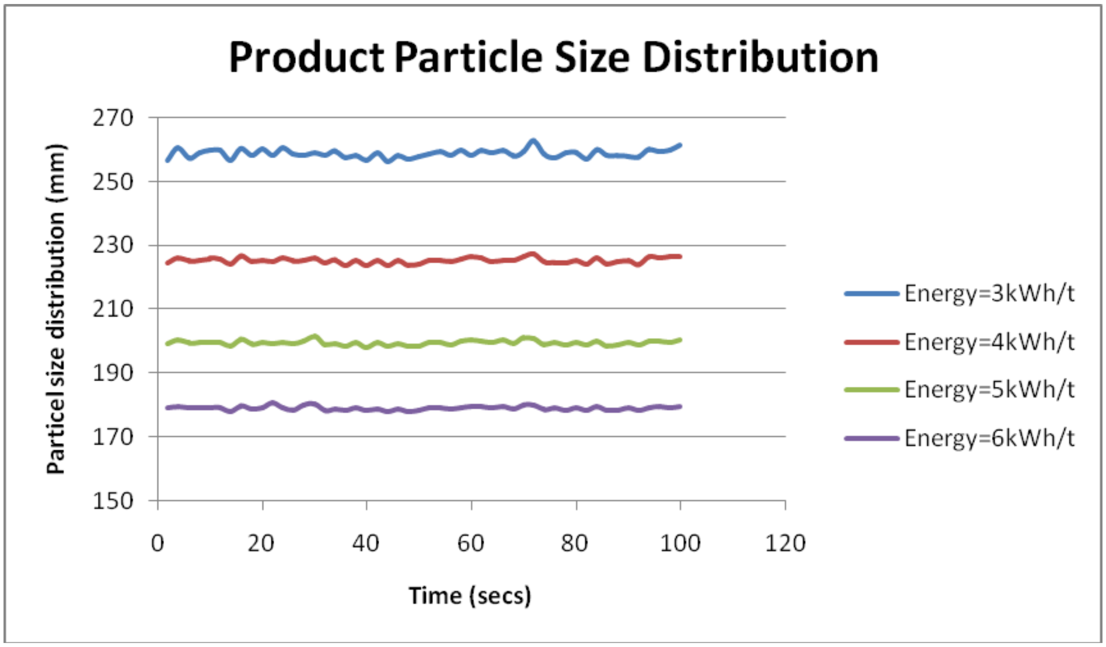


Figure 4: Particle size distribution of product (mm) over the simulation time

6. Results and discussion

After simulating for different input energy ranging from 3 kWh/t to 6 kWh/t for the given feed particle size, the summary of the product particle size obtained is shown in Appendix 1. From figure 4, it can be seen that as the crushing energy increases the particle size distribution curve becomes smoother which represents a better fragmentation during crushing. The variance of the product particle size also reduces with increase in crushing energy which means oversize and undersize particles becomes less likely to occur when there is enough energy for fragmentation.

Table 2 illustrates a compilation of the mean and variance of the product particle size obtained from the results of the simulation. It was compiled by finding the mean product particle size for each of the crushing energy used in the simulation. It shows that as the crushing energy increases the mean of the product particle size reduces as well as the variance. This is as a result of improvements in fragmentation with increase in crushing energy.

Table 2: Summary of mean product particle size and input energy

Energy (kWh/t)	Mean of Product Particle Size (mm)	Variance of Product Particle Size (mm)
3	259	1.74
4	225	0.76
5	199	0.54
6	179	0.37

Figure 5 shows the graphical relationship between the mean and variance of the product particle size and the input energy used in the crushing process. From the graph, for any required product particle size distribution covered by the simulation, the corresponding crushing energy required to achieve that fragmentation from the given feed particle size distribution can be mapped. For our required product particle size distribution with mean 203mm and variance 5mm, the required energy is about 4.9kWh/t. This is shown on figure 5.

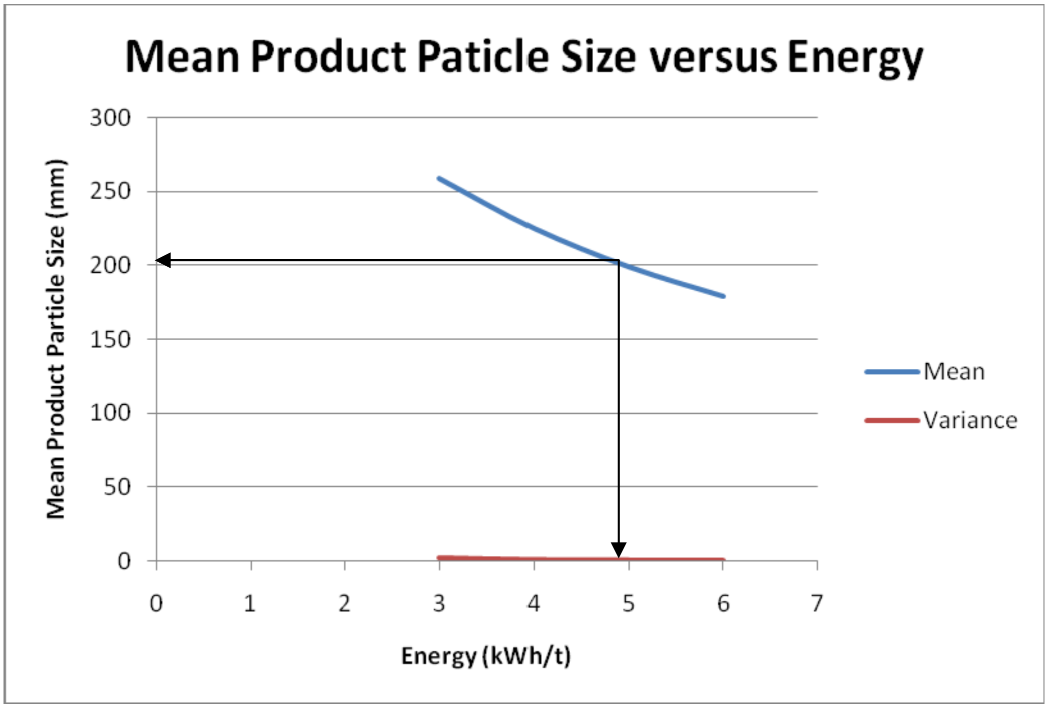


Figure 5: Graph of mean and variance of product particle size (mm) versus energy (kWh/t)

6.1. Reduction ratio

Reduction Ratio is the ratio between the product particle size and the feed particle size. It is normally used as first hand information to know the extent to which particles are being reduced from their initial size in the crushing process. The reduction ratio for the simulation is calculated using the mean of the product particle size distribution and the mean of the feed particle size distribution and this is shown in table 3. Table 3 shows that as the crushing energy increases, the reduction ratio also increases which means greater fragmentation is being achieved.

Table 3: Reduction Ratio of crushing process

Input Energy (kWh/t)	Mean of Product Particle Size (mm)	Mean of Feed Particle Size (mm)	Reduction Ratio (%)
3	259	1000	74.1
4	225	1000	77.5
5	199	1000	80.1
6	179	1000	82.1

7. Conclusions

From figure 5, the relationship between energy and product particle size can be mapped. For the given feed particle size, the product particle size can be varied by varying the energy input to the crushing process. The graph shows that as the input energy increases the product particle size decreases. Similarly, the variance, which is a measure of the oversize and undersize particles with respect to the mean also decreases with increase in energy. Therefore, to reduce process rejects or oversize, the energy has to be increased appropriately.

The reduction ratio of the crushing process increases with increase in crushing energy. This means greater fragmentation is achieved as the crushing energy is increased. With the given feed particle size distribution with mean 1000mm and variance 300mm, the required crushing energy to obtain the required product particle size distribution with mean 203mm and variance 5mm is 4.9kWh/t.

8. References

[1] Ben-Awuah, E. and Akayuli, C. F. A., (2008), "Analysis of Embankment Stability of The Water Storage Facility at Awaso Mine", MSc Thesis, © University of Mines and Technology, Tarkwa, Pages 86.

[2] Fuerstenau, M. C. and Han, K. N., (2003), "Principles of Mineral Processing", © Society for Mining, Metallurgy and Exploration, Inc, Littleton, Colorado, First ed, Pages 573.

[3] Kawatra, S. K., (2006), "Advances in Communution", © Society for Mining, Metallurgy and Exploration, Inc, Littleton, Colorado, First ed, Pages 557.

[4] Kennedy, B. A., (1990), "Surface Mining", © Society for Mining, Metallurgy and Exploration, Inc, Littleton, Colorado, Second ed, Pages 1194.

[5] Wills, B. A. and Napier-Munn, T. J., (2006), "Wills' Mineral Processing Technology", © Elsevier Ltd, Seventh ed, Pages 444.

[6] Wittwer, J. W., (2004), "Monte Carlo Simulation Basics", Retrieved April 9, 2009 from: http://vertex42.com/ExcelArticles/mc/MonteCarloSimulation.html.

9. Appendix 2: Matlab Code used in executing Simulink model

Matlab code used in executing the Simulink model

10. Appendix 3: Instructions on running the MATLAB code

Simulation of a Crushing Process

Table of Contents

----

1. Introduction

This package is built for the simulation of a crushing process using the mathematical model of Bond's operating work index equation. A Monte Carlo Simulation method is used. Using Simlulink, the mathematical is converted to a simulation model. This simulation model is executed from a code written in Matlab which runs the simulation model and compiles the results for further analysis.

The default multiple runs of the simulation is 10000, however this can be altered to suite personal requirements. The feed particle size distribution and the input crushing energy can also be altered depending on your crushing requirements.

a) Matlab Code filename: crushing1.mat

b) Simulink model filename: crushing_rev1.mdl

2. Minimum requirements

Matlab 7.5.0 (R2007b)

Simulink 7.5.0 (R2007b)

3. Getting Started

c) Open the Matlab file containing the code.

d) Open the Simulink file containing the simulation model.

e) Ensure that the Simulink model filename is the same as that used in the code on line 20.

f) Change line 12 to the number of simulation runs for each energy variable.

g) Change line 18 to the feed particle size distribution.

h) Change line 8 to the initial crushing energy value and subsequent increase in this value for subsequent runs can be made on line 44.

i) Once all these parameters are set, execute the simulation from the run icon in Matlab.

4. Results

The results of the Simulation will be output to the Matlab workspace. The variable "H" displays the feed particle size distribution and "C" displays the product particle size distribution. The struct files for these variables can be exported to excel for further analysis.

Simulation of mine production using MATLAB and AweSim

Samira Kalantari[1] and Hooman Askari-Nasab

1. Introduction

Abstracting a real system into a system which is not real is called simulation; in other words, simulation is the mathematical representation of the interaction of real-world objects. A system can be specified by a set of variables where each variable can describe the state of the system and changes in the system from state to state. A simulation model involves considering the behavior of the model by changing it from state to state. In case of formulating a problem, simulation should be amongst the first approaches considered, and it's a way of developing the level of understanding of the interactions between different parts of the system (Alan et al., 1999).

1.1. Discrete event simulation

Generally to simulate a system, a mathematical model of the system should be created. Changes in the state of the system can happen over continuous time or at discrete points of time, therefore there are three types of simulation models: continuous, discrete-event simulation, and a combination of two (Raczynski, 2009).

In discrete event simulation the models are restricted to discrete-event models. In a discrete-event model there is a finite number of transitions overtime. In a discrete-event simulation, the goal is defining the states of a system and constructing the activities that move from a state to another state. In other words, discrete-event simulation models the system as a set of individual entities which move along the system in discrete time (Tako and Robinson, 2009).

2. Problem definition

2.1. Simulation of mine scheduling

In this project, a discrete-event simulation model will be used to simulate the mine production schedule. The mine schedule is simulated by the order of extraction of blocks in each period of mining. Four disposal locations are possible for the extracted material, waste dumps, crusher, low-grade stockpiles, and high-grade stockpiles. In this model, the trucks will have different traveling speeds, and payloads. The governing relationships between these objects will be simulated. There are some unexpected events which can make a delay in the schedule such as equipment breakdowns and weather conditions that can be included in the model.

[1] MSc candidate in Mining Engineering

We will use AweSim simulation software (Alan et al., 1999) to model the mine production process. AweSim includes the Visual SLAM language, which we will use to create networks, sub networks, discrete event and continuous models of the mine production. Visual SLAM network structures consist of nodes and branches, which are used to build a network that represents the mining process.

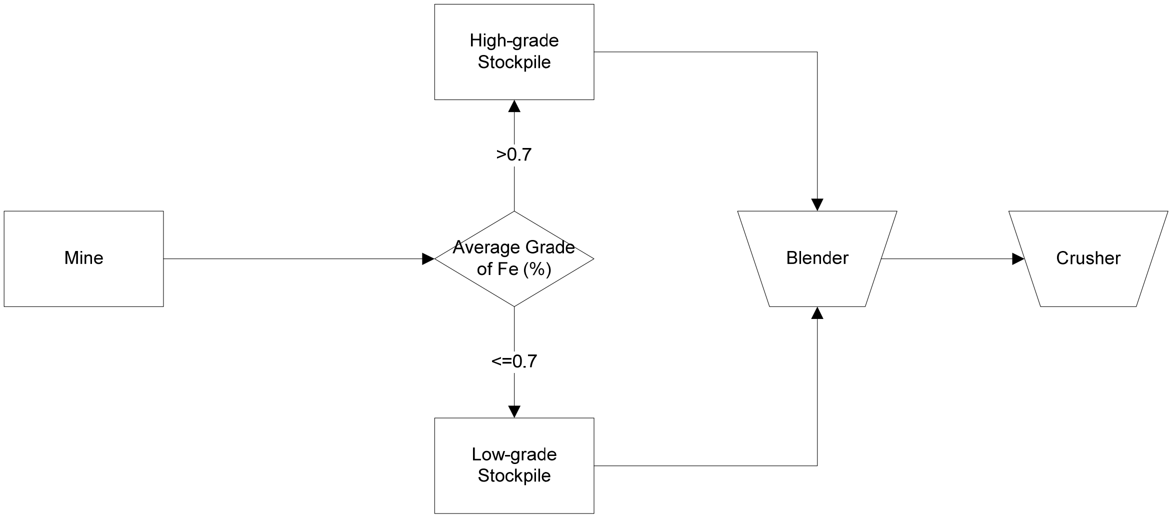


Figure 1: Schematic diagram of the simulation process

Figure 1 shows the schematic diagram of the simulation process, the blocks coming from the mine go to either stockpile 1 ore stockpile 2 according to their average grade of Fe. Different portions of material will go to the blender according to the desired cut-off grade of Fe. Afterwards, the blocks will go to the crusher. We want to simulate this process, and see how the breakdown of blender and crusher will affect the whole system.

3. Methodology

3.1. Building the network

In this study we have used an iron ore block model. Every block has its own properties such as: average grade of ore, ore tonnage, waste tonnage, economic block value, etc. The block model contains 8456 blocks. Three types of blocks are defined within the block model, ore, waste, and air blocks. The following attributes are modeled for each block: coordinates, ore tones, block tonnage, grade of Fe, grade of P, grade of S, etc.

The waste blocks are those that economic block values are negative. The economic block value for the air blocks is equal to zero.

First, we imported the block model data into MATLAB. The imported data is saved within a MATLAB *struct* data structure. The struct consists of one row and 8456 (number of blocks) columns. For statistical analysis we needed to separate different block types. We developed a code in MATLAB code to separate different block type's data. The code for removing waste and air blocks is represented in appendix A.

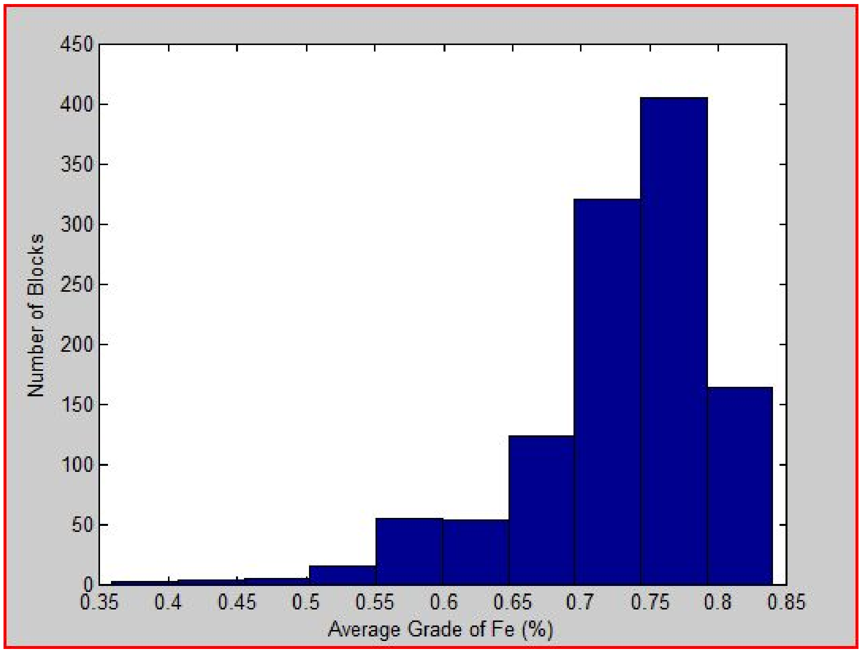


Figure 2: Histogram of the average grade of Fe (%) for blocks

Figure 1 illustrates the histogram of grade distribution of Fe% in the block model. We have used the histogram in Figure 2 to fit a probability density function. In simulation with AweSim, a distribution to represent the data is used where the distribution provides more information than the direct use of data. The question is whether or not the distribution fits the corresponding data. Using the difittool in Matlab, the best distribution on data was found, which is shown in figure which is the Weibull distribution 2.

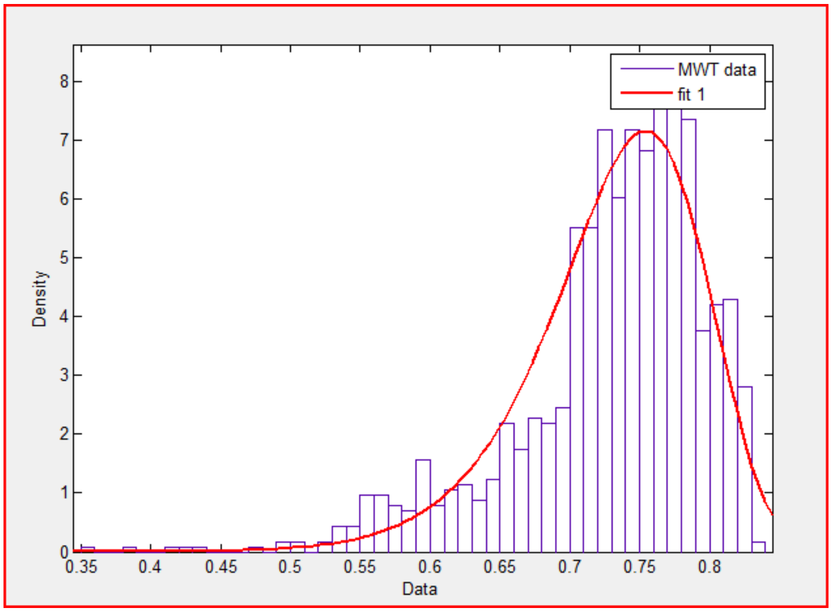


Figure 3: Weibull distrubution

Table 1: Data from the distribution of figure 2.

DISTRIBUTION	MEAN	VARIANCE	STD.	β	α
Weibull	0.7313	0.0037	0.0608	0.7570	14.7234

Similarly, using MATLAB we should find the tonnage of each block. In order to import this data in Awesim, we fit a distribution on the tonnage of the blocks. The best distribution fitting on this data (block tonnage) is the normal distribution, the data for this distribution is provided in table 2.

Table 2: Data for the distribution for block tonnage.

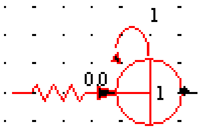
DISTRIBUTION	MEAN	VARIANCE	STD.
Normal	705180	6.53e+10	255572.49

As this project deals with the section moving from the stockpiles to the crusher, we can have two different stockpiles according to the grade of Fe, low-grade stockpile and high-grade stockpile. A condition can be set in order to separate the blocks according to their average grade of Fe.

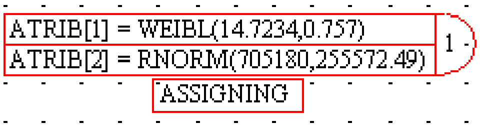
If average grade of Fe in the block >0.7 → goes to the high-grade stockpile

If average grade of Fe in the block <=0.7→ goes to the high-grade stockpile

In Awesim, after creating the entities with the create node, the time between entities can be considered, in this project, the entities are assumed to be block models, and the time between created entities can be set to 1 unit of time. Create node generates entities and routes them into the system over activities emanating from this node.

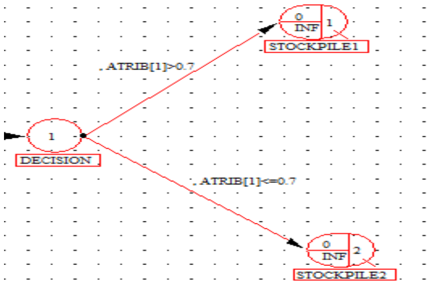


After entering of the blocks to the system, the grade of Fe and the tonnage of each block should be assigned to it. Using the distributions fitted to the data, the ATRIB[1] is the average grade of Fe of each block, and ATRIB[2] is the tonnage of each block. Assign node is used to assign values to the activities which are passing through this node. Therefore, the assign node should be as follows.



```
ATRIB[1] = WEIBL(14.7234,0.757)
ATRIB[2] = RNORM(705180,255572.49)
```

After assigning the average grade and the ore tonnage, the blocks can be going to either the low-grade and high-grade stockpiles according to ATRIB[1]. Queue node: entities wait at this node for service.



The above figure shows two queue nodes which are high-grade and low-grade stockpiles, stockpile 1 contains the blocks with the grade of Fe greater than 0.7, and stockpile 2 contains the blocks with the grade of Fe less than or equal to 0.7.

After sending the entities to two different queue nodes (stockpiles) these entities should go to the blender, to have different portions of each stockpiles in the blender, the duration for sending blocks from these two stockpiles should be different. In other words, in order to achieve the desired grade variation after being processed in the blender, the duration for sending blocks from stockpile 1 and stockpile 2 should be different.

Therefore, the duration for two different paths should vary in such a way that the final grade variation is in the desired range. In order to count the number of blocks going to blender from stockpile 1 and stockpile 2, an assign node was created.

$$XX[2]=XX[2]+1 \qquad (1)$$

In equation (1), the initial value for XX[2] is zero, and when an entity departs from stockpile1, XX[2] is increased by 1. Similarly, an assign node should be created to count the number of blocks going from stockpile2 to the blender, which is shown in equation (2).

$$XX[3]=XX[3]+1$$

Where the initial value for XX[3] in equation (2) is zero. In addition, the ore tonnage of each block can be calculated by assigning an attribute which is the product of the average grade of Fe of the block and the block tonnage of equation (3).

$$\text{Ore Tonnage of the block= average grade of Fe * block tonnage} \qquad (3)$$

Therefore, the value of XX[2] and XX[3] will be dependent on the duration of the activity. To find the average grade of the blocks which are going to the blender from the high-grade stockpile the ore tonnage of the block and the average grade should be found. In order to find the average grade of the blocks after blending, in the assign node these values should be assigned which is showed in equation (4).

$$\sum_{i=1}^{n} X_i * O_i \tag{4}$$

Where:
n, number of blocks coming from stockpile1
$X_{i,}$ is the average grade of Fe for the ith block
$O_{i,}$ is the ore tonnage of the ith block.

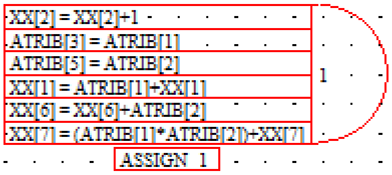
XX[7]=(ATRIB[1]*ATRIB[2])+XX[7] (5)

In equation (5), XX[7] will calculate the desired summation of the blocks coming from stockpile 1. In addition, for finding the average grade of Fe after being processed in the blender, the total tonnage coming from stockpile 1 should be calculated, which are shown in equation (6) and equation (7).

XX[8]=ATRIB[2]+XX[8] (6)

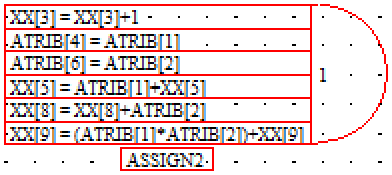
XX[9]=(ATRIB[1]*ATRIB[2])+XX[9] (7)

The assign node for stockpile 1 is as follows:



```
XX[2] = XX[2]+1
ATRIB[3] = ATRIB[1]
ATRIB[5] = ATRIB[2]
XX[1] = ATRIB[1]+XX[1]
XX[6] = XX[6]+ATRIB[2]
XX[7] = (ATRIB[1]*ATRIB[2])+XX[7]
```

And similarly, the assign node for stockpile 2 is as follows:



```
XX[3] = XX[3]+1
ATRIB[4] = ATRIB[1]
ATRIB[6] = ATRIB[2]
XX[5] = ATRIB[1]+XX[5]
XX[8] = XX[8]+ATRIB[2]
XX[9] = (ATRIB[1]*ATRIB[2])+XX[9]
```

The above expression is assigned with the initial value of XX[7] equal to zero. Therefore,

After the await node for the blender a collect node should be created in order to calculate the average grade of the blender. This average grade depends on the number of the blocks coming from stockpile number one, and also on the number of the blocks coming from stockpile number 2, respectively XX[2] and XX[3].

The average grade of the blender should be calculated is shown in equation (8).

$$g = \frac{\sum_{i=1}^{2} grade * tonnage}{tonnage1 + tonnage2} \tag{8}$$

The nominator of equation (8) is equal to:

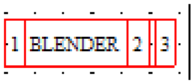
$$\sum_{i=1}^{n} X_i * O_i + \sum_{i=1}^{m} X_i * O_i \tag{9}$$

Where:
n is the number of blocks coming from stockpile1.
And m is the number of blocks coming from stockpile2.
The nominator of equation (8) is equal to:

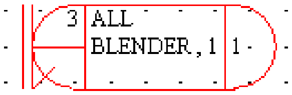
$$\sum_{i=1}^{n} O_i + \sum_{i=1}^{m} O_i \tag{10}$$

3.2. Blender

The blocks will wait in the await node for the blender, the general process of the blender is mixing the blocks to get the desired average grade of the mixture. The blocks which are waiting in the await node for the blender, will be sent to the blender, and the capacity for the blender is 2 which means that the crusher (resource) can process 2 blocks at time. And the processing time for the crusher is 7 unites of time.



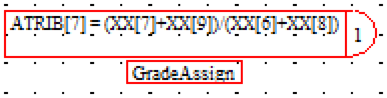
The figure above shoes the resource node which is the blender for this system; resource block identifies the resource name or label and the initial capacity for the resource. The initial capacity of the resource is 10, and the await node which will work with the resource is as follows:

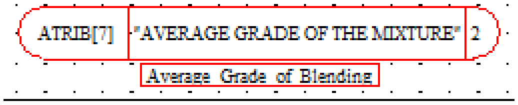


Await node stores entities waiting for units of resources. The entities are blocked in this node, which means that the entities (blocks) should stay in this node until the resource be free.

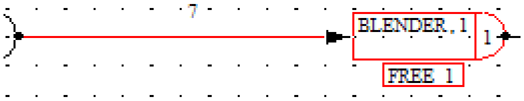
The whole process of the blender is blending the blocks coming from the two stockpiles in such a way that the grade variation of the blocks going to the crusher and afterwards to the mill be in-spec. In order to achieve the desired grade variation the duration of moving blocks from the two stockpiled should vary. The duration difference shows the difference in the number of blocks going from two stockpiles to the blender.

Colct node is a location where statistics can be collected on any expression. The colct node should calculate the average grade of blending, therefore an assign node should be created, and an attribute should be assigned to that value:



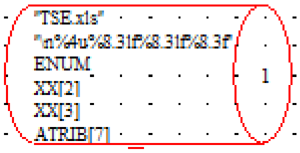


As the processing time for the blender is 7 units of time, this amount of time will take to process each block, therefore the duration for sending the blocks from the recourse to the free node is 7 units of time. Free node releases units of resource type when an entity arrives to the node.



After being processed in the blender, the blocks will be sent to the await node for the crusher. The duration for this process is assumed to be 2.5 units of time which means that 2.5 units of time will take to send the blocks from blender to the crusher.

The blocks will gather up in the await node for the crusher, and in order to send all of them to the crusher the blocking will be done. Blocking will force the blocks to wait in that await node until the crusher be free. The initial capacity for the crusher assumed to be 5, which means it can process 5 blocks at the same time, and the processing time is assumed to be 10 which means that the time for each block to be processed in the crusher is 10 units of time. In order to import the data for average grade of time, number of blocks coming from the stockpile 1, and number of blocks coming from stockpile 2, a write node is defined. Write node writes one or more values to an external file.

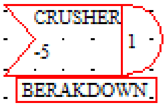


This will import this data in an Excel file.

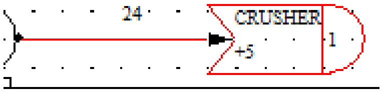
3.3. Breakdown of the crusher

It can be assumed that after 200 units of simulation time, the crusher will break down, therefore the waiting time for the blocks in the crusher await node will increase. The process of fixing and repairing the crusher will take 24 units of time, and after that the crusher will process the waiting blocks:

The process of breaking down and fixing the crusher can be simulated using the Alter node. The Alter node will change the capacity of the resource; therefore in this case, the Alter will change the capacity of the blender to -2 which means that the crusher is out of service.

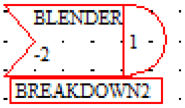


The figure above shows the Alter node which works with the crusher resource, and change the capacity of the crusher by -2, and the initial capacity for the crusher is 2, therefore this Alter node will change the resource capacity to zero which shows that the crusher is out of service. After 24 hours of time, the crusher is fixed; another Alter node should increase the crusher capacity to the initial value:



3.4. Breakdown of the blender

After 200 units of time of the simulation, the blender will be out of service and the fixing process will be 24 units of time.



The above Alter node will decrease the blender capacity to zero, indicating that the blender is out of service.

4. Implementation

4.1. Control statement

In the control statement of this simulation, the limits for the maximum index for attributes and max index for XX (global variable) should be defined. The maximum number of attributes is 7, where the maximum number of XX variable is 9.

INTLC is used to assign initial values to the variables.
INTLC,{{XX[2],0},{XX[3],0},{XX[1],0},{XX[5],0},{XX[7],0},{XX[8],0},{XX[9],0},{ XX[6],0}};

5. Results and discussions

5.1. Sensitivity analysis

5.1.1 Running without breaking down of the blender and crusher

In order to find out the influence of the crusher blender on the system, first the network without blender break down should be simulated. The network for this simulation is provided in appendix B, and the summary report of AweSim is provided in appendix C.
The summary report of Awesim is indicated in table 3 and 4.

Table 3: AweSim summary report 1.

Label	Mean Value	Std.	Number of Observations	Minimum value	Maximum Value
Average grade	0.719	0.008	86	0.695	0.767
Ore Tonnage stockpile 1	35429938	20486838	100	807127	70773022
Ore Tonnage Stockpile2	20541517	114145323	89	452553	39271039
No. Blocks from STP1	49.849	29.088	86	1	99
No. Blocks from STP2	32.930	19.769	86	0	68

Table 4: AweSim summary report 2.

File Number	Label or Input Location	Average Wait Time
Queue	High-grade stockpile	86.310
Queue	Low-grade stockpile	0.435
Resource	Blender	70.337
Resource	Crusher	17.00

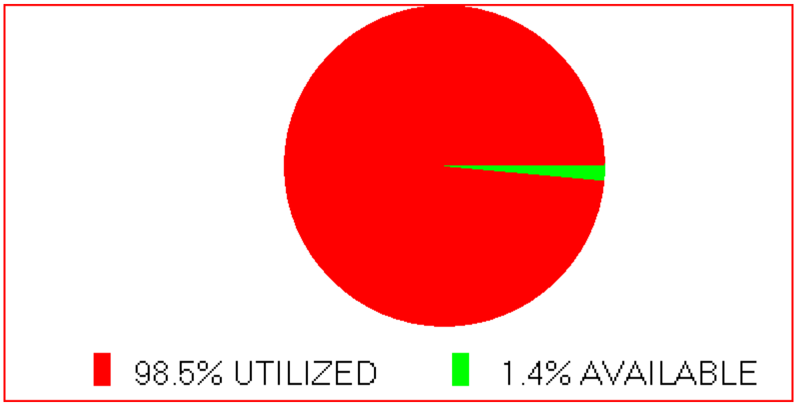


Figure 5: Utilization of the blender

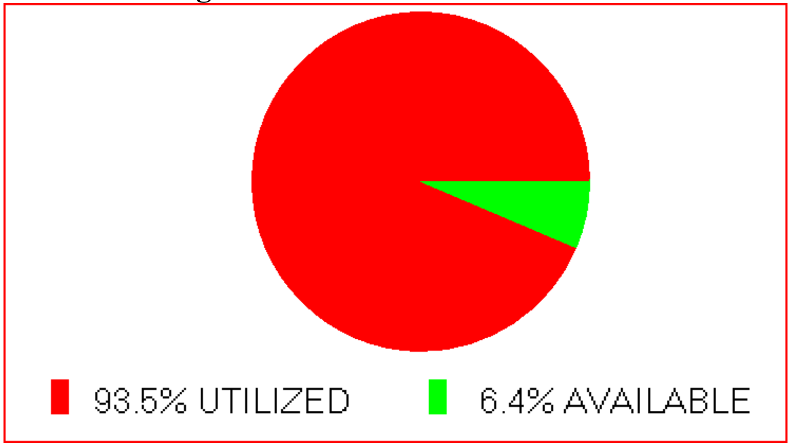


Figure 6: Utilization of the crusher

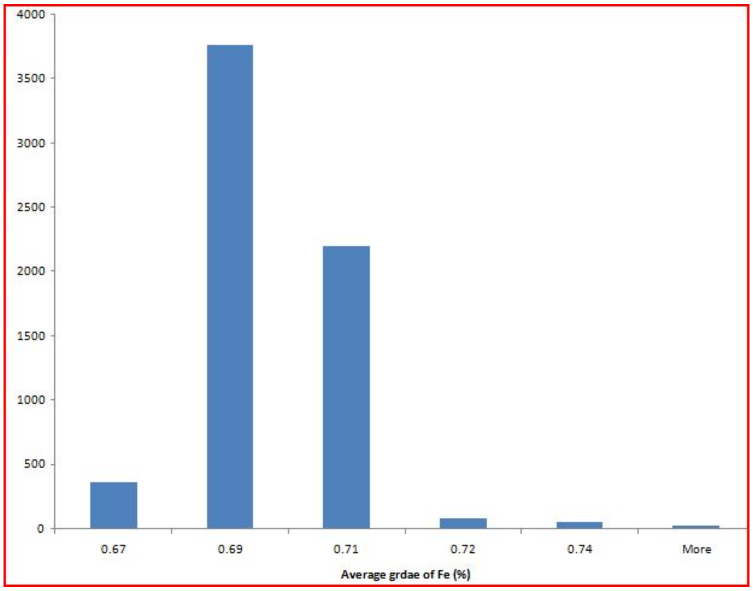


Figure 7: Histogram for average grade of blending after 100 runs

5.1.2 **The impact of breakdown of the crusher on the system**

a) Average waiting time for the crusher:

To consider the impact of the breakdown of the crusher on the system, the Alter node is added to the network. The network is in appendix D and the summary report of Awesim is in appendix E.

Table 5: AweSim report (crusher breakdown has been simulated).

File Number	Label or Input Location	Average Wait Time
Queue	High-grade stockpile	86.310
Queue	Low-grade stockpile	0.435
Resource	Blender	70.337
Resource	Crusher	20.584

Table 5 shows that when the crusher breakdown is simulated in the system, the average waiting time for the crusher at the await node for this resource will increase, which makes sense since the breaking down of the crusher and the time required for fixing it causes the pile up at the await node. According to table 4, the average waiting time for the crusher without simulating the breakdown is 17.00 units, while when the crusher breakdown occurs, the average waiting time for this resource increases to 20.584 units.

b) Utilization of the crusher

From the figure 7 and comparing to figure 5 it can be seen that when the crusher breakdown is simulated in the system, the utilization is increased to 97.1%.

5.2. The impact of breakdown of the blender on the system

a) Average waiting time for the blender

Table 6: AweSim summary report (breakdown of the blender).

FILE NUMBER	LABEL OR INPUT LOCATION	AVERAGE WAIT TIME
Queue	High-grade stockpile	86.310
Queue	Low-grade stockpile	0.435
Resource	Blender	71.036
Resource	Crusher	16.247

Table 6 and comparing it to table 4 shows that by simulating the blender breakdown in the system, the average waiting time for the blender resource will increase to 71.036, while the waiting time for the crusher decreases. The network for this simulation is in appendix F, and the AweSim summary report is in appendix G. The decrease in the average waiting time at the crusher await node is due to the increase in the blender await node, in other words, as the process of blending is before the crusher process, when the waiting time for the first one increases, the waiting time for the other resource will decrease.

b) Utilization of the blender

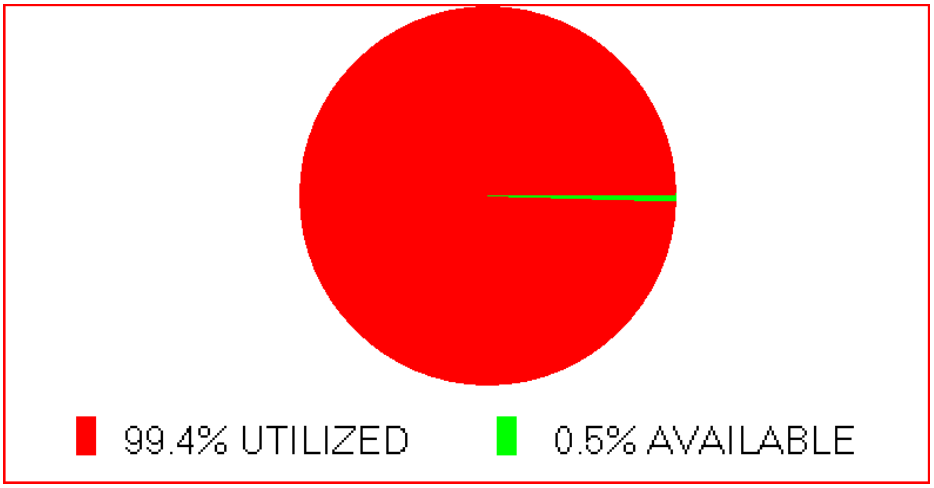


Figure 8: Utilization of the blender when blender breakdown occurs in the system

Figure 7 shows that when the breakdown of the blender occurs in the system, the utilization percent of the blender is increased to 99.4% which means that the idle time for the server is decreased.

5.3. The impact of duration on the average grade

In order to meet the desired range for the average grade the duration for sending blocks from low-grade and high-grade stockpiles should be considered. In other words, to satisfy the conditions for the desired average grade, the portion of blocks which are coming from two different stockpiles should be considered. In the simulation network for this project the duration for sending blocks from high-grade stockpile was set to 3 units of time, while the duration for sending blocks to the blender form low-grade stockpile was 2, and according to table 1 the average grade of the blending was 0.719%.

At this stage it can be assumed that the desired range of the grade is 0.6900 ± 0.01, therefore the portion of blocks from stockpile 2 should be more than the portion of blocks from stockpile 1. To see the impact the durations are changed to 8 and 1.5, the result is as fin table 7.

Table 7: AweSim summary report.

Label	Mean Value	Std.	Number of Observations	Minimum value	Maximum Value
Average grade	0.695	0.005	82	0.675	0.723
Ore Tonnage stockpile 1	15497770	8789186	37	681815	28818244
Ore Tonnage Stockpile2	25822516	13651720	69	633054	4903539
No. Blocks from STP1	18.159	10.70	82	0	37
No. Blocks from STP2	32.646	19.430	82	1	68

Table 8: AweSim summary report.

File Number	Label or Input Location	Average Wait Time
Queue	High-grade stockpile	126.716
Queue	Low-grade stockpile	0.174
Resource	Blender	27.953
Resource	Crusher	20.206

As it can be seen form table 7 and table 8 and comparing them to table 4, increasing the duration of sending blocks from high-grade stockpile, and decreasing the other duration decreases the average grade of blending and increases the average waiting time for blocks in the high-grade stockpile to 126.76 while the average waiting time for the blocks in the low-grade stockpile to be sent to the blender has been decreased to 0.174.

6. References

[1] Alan, A., Pritsker, B., and O'reilly, J., (1999), "Simulation with visual SLAM and AweSim", © System publishing and John Wiley, New York, Second ed, Pages 828.

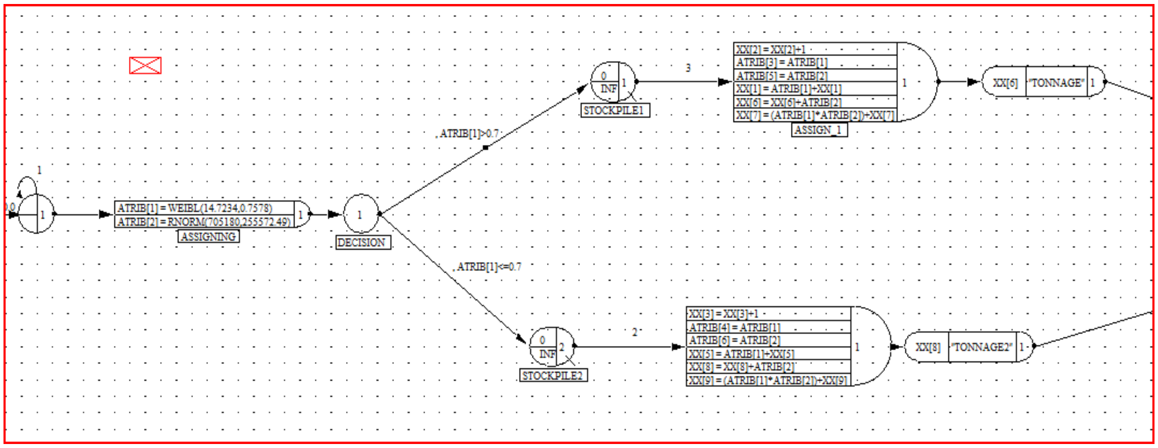
[2] Raczynski, S., (2009), "On validity of discrete event models", © Retrieved april 2009, 2009 from: http://www.raczynski.com/art/validity.doc

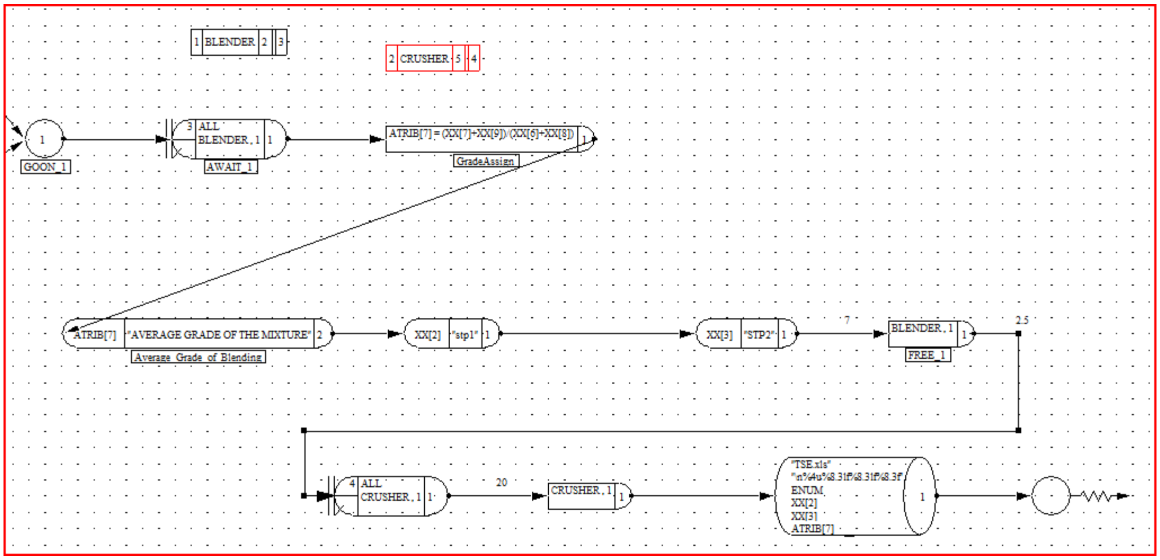
[3] Tako, A. and Robinson, S., (2009), "Comparing discrete event simulation and system dynamics", *Journal of the operational research society*, Vol. 60, pp. 296-312.

Appendix A

```
f=1;
for i=1:8456
   m(i,1)=Blocks(1,i).gradeMWT;
   if m(i,1)>0
      MWT(f,1)=m(i,1);
      f=f+1;
   end
end
hist(MWT)
grid on
```

Appendix B



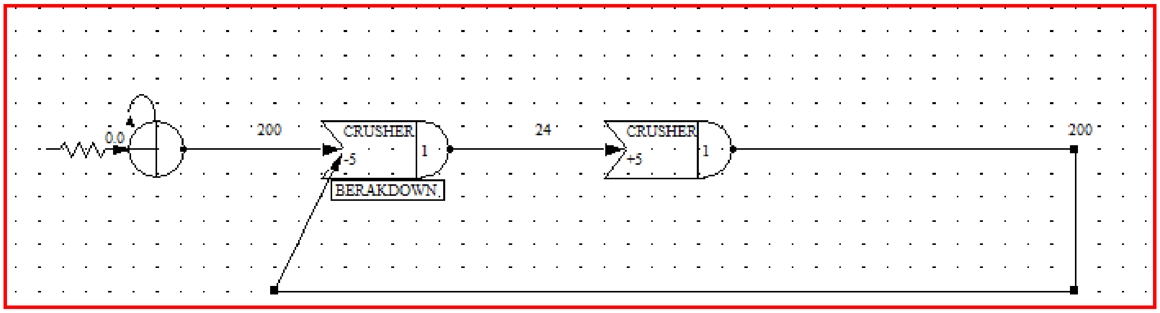


Appendix C

```
                          ** AweSim SUMMARY REPORT **
                           Thu Apr 09 11:45:01 2009


  Simulation Project : modeling
  Modeler : samira
  Date :
  Scenario : BASECASE

  Run number 100 of 100
  Current simulation time    : 300.000000
  Statistics cleared at time : 0.000000


           ** OBSERVED STATISTICS REPORT for scenario BASECASE **


     Label         Mean      Standard    Number of     Minimum
Maximum
                   Value     Deviation  Observations    Value
Value

TONNAGE          38027424.546 20988046.747        100  681815.021
72801400.864
AVERAGE GRADE OF       0.719       0.008         86       0.709
0.767
stp1                  49.849      29.088         86       1.000
99.000
STP2                  32.930      19.769         86       0.000
68.000
TONNAGE2         25822516.855 13651720.785         69  633054.661
49035395.754

             ** FILE STATISTICS REPORT for scenario BASECASE **


  File       Label or       Average    Standard   Maximum  Current
Average
 Number   Input Location    Length     Deviation   Length   Length   Wait
Time

       1 QUEUE    STOCKPI      66.747      38.116      131      131
86.310
       2 QUEUE    STOCKPI       0.100       0.300        1        0
0.435
       3 RES. BLENDER          39.623      23.723       83       83
70.337
       4 RES. CRUSHER           4.703       3.095       10       10
17.000
       0 Event Calendar         9.800       1.555       11       10
3.680
```

```
      ** SERVICE ACTIVITY STATISTICS REPORT for scenario BASECASE **


Activity     Label or      Server    Entity      Average    Standard
 Number    Input Location  Capacity   Count    Utilization  Deviation


       0 Line 11                1           0        1.000        0.000
       0 Line 38                1           0        0.460        0.498

Activity    Current      Average      Maximum      Maximum
 Number    Utilization  Blockage     Idle Time    Busy Time
                                    or Servers   or Servers

       0           1       0.000        0.000      300.000
       0           0       0.000       16.000        8.000
          ** RESOURCE STATISTICS REPORT for scenario BASECASE **


Resource      Resource          Average    Standard    Current     Maximum
 Number        Label             Util.     Deviation    Util.       Util.

       1 BLENDER                 1.970       0.222          2           2
       2 CRUSHER                 4.678       1.128          5           5

Resource    Current     Average    Current    Minimum    Maximum
 Number     Capacity   Available  Available  Available  Available

       1          2       0.030          0          0          2
       2          5       0.322          0          0          5
```

Appendix D

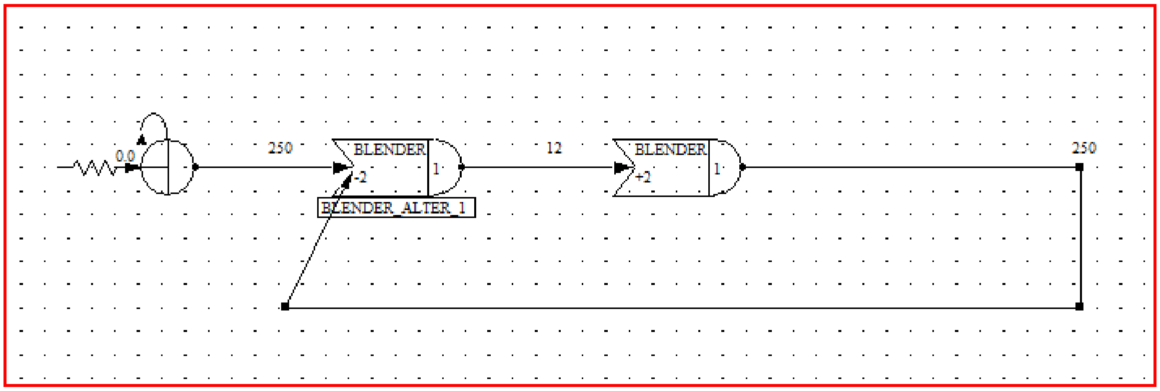


Appendix E

```
                         ** AweSim SUMMARY REPORT **
                          Thu Apr 09 11:52:56 2009


  Simulation Project : modeling
  Modeler : samira
  Date :
  Scenario : BASECASE

  Run number 100 of 100
  Current simulation time    : 300.000000
  Statistics cleared at time : 0.000000


         ** OBSERVED STATISTICS REPORT for scenario BASECASE **


     Label          Mean      Standard    Number of     Minimum
Maximum
                    Value     Deviation  Observations    Value
Value

TONNAGE          38027424.546 20988046.747        100  681815.021
72801400.864
AVERAGE GRADE OF        0.719       0.008         83       0.709
0.767
stp1                   48.542      28.745         83       1.000
98.000
STP2                   32.036      19.516         83       0.000
67.000
TONNAGE2         25822516.855 13651720.785         69  633054.661
49035395.754

           ** FILE STATISTICS REPORT for scenario BASECASE **


  File       Label or       Average    Standard   Maximum  Current
Average
 Number   Input Location    Length     Deviation   Length   Length   Wait
Time

       1 QUEUE    STOCKPI      66.747      38.116      131      131
86.310
       2 QUEUE    STOCKPI       0.100       0.300        1        0
0.435
       3 RES. BLENDER          40.017      24.307       86       86
7.337
       4 RES. CRUSHER           5.378       3.992       14       13
20.584
       0 Event Calendar        11.512       1.742       14       11
4.355
```

```
      ** SERVICE ACTIVITY STATISTICS REPORT for scenario BASECASE **


Activity      Label or      Server    Entity      Average     Standard
 Number    Input Location  Capacity    Count    Utilization  Deviation


       0 Line 11                 1           0        1.000        0.000
       0 Line 38                 1           0        0.460        0.498

Activity     Current      Average      Maximum      Maximum
 Number    Utilization  Blockage     Idle Time    Busy Time
                                   or Servers   or Servers

       0           1        0.000        0.000      300.000
       0           0        0.000       16.000        8.000

          ** RESOURCE STATISTICS REPORT for scenario BASECASE **


Resource      Resource         Average    Standard    Current     Maximum
 Number        Label           Util.     Deviation    Util.       Util.

       1  BLENDER                 1.910        0.385          2           2
       2  CRUSHER                 4.467        1.365          5           5

Resource     Current     Average     Current     Minimum     Maximum
 Number      Capacity    Available   Available   Available   Available

       1           2        0.010          0          -2           2
       2           5        0.133          0          -5           5
```

Appendix F



Appendix G

```
                        ** AweSim SUMMARY REPORT **
                         Thu Apr 09 11:58:24 2009


  Simulation Project : modeling
  Modeler : samira
  Date :
  Scenario : BASECASE

  Run number 1 of 100
  Current simulation time    : 300.000000
  Statistics cleared at time : 0.000000


        ** OBSERVED STATISTICS REPORT for scenario BASECASE **


     Label          Mean      Standard    Number of     Minimum
Maximum
                    Value     Deviation  Observations    Value
Value

TONNAGE          38027424.546 20988046.747        100  681815.021
72801400.864
AVERAGE GRADE OF       0.719       0.008         83       0.709
0.767
stp1                  48.542      28.745         83       1.000
98.000
STP2                  32.036      19.516         83       0.000
67.000
TONNAGE2         25822516.855 13651720.785         69  633054.661
49035395.754

          ** FILE STATISTICS REPORT for scenario BASECASE **


  File       Label or       Average    Standard   Maximum  Current
Average
 Number   Input Location    Length     Deviation   Length   Length   Wait
Time

       1 QUEUE    STOCKPI      66.747      38.116      131      131
86.310
       2 QUEUE    STOCKPI       0.100       0.300        1        0
0.435
       3 RES. BLENDER          40.017      24.307       86       86
71.036
       4 RES. CRUSHER           4.387       2.705        9        8
16.247
       0 Event Calendar        10.723       1.617       13       10
4.052
```

```
      ** SERVICE ACTIVITY STATISTICS REPORT for scenario BASECASE **

Activity     Label or      Server    Entity      Average    Standard
 Number   Input Location  Capacity    Count    Utilization  Deviation


       0 Line 11                 1           0       1.000       0.000
       0 Line 38                 1           0       0.460       0.498

Activity    Current     Average     Maximum     Maximum
 Number   Utilization  Blockage    Idle Time   Busy Time
                                  or Servers  or Servers

       0           1       0.000       0.000     300.000
       0           0       0.000      16.000       8.000

          ** RESOURCE STATISTICS REPORT for scenario BASECASE **

Resource     Resource       Average    Standard    Current    Maximum
 Number       Label          Util.     Deviation    Util.      Util.

       1 BLENDER                1.910       0.385          2          2
       2 CRUSHER                4.678       1.128          5          5

Resource   Current    Average     Current    Minimum    Maximum
 Number    Capacity   Available  Available  Available  Available

       1         2       0.010          0         -2          2
       2         5       0.322          0          0          5
```

Cable shovel dynamic simulation application in pre-blasting decision-making

Kwame Awuah-Offei[1] and Hooman Askari-Nasab

Abstract

Shovel loading costs constitute up to 35% of total surface mining operating costs. Additionally, cable shovels are capital intensive units purchased only because of the anticipated efficient utilization over its long life. The efficient utilization of shovels is important in the quest to reduce mining costs. The decision to pre-blast has been traditionally made based on total energy efficiency of blasting and comminution processes without regard to the impact on shovel utilization. However, the efficiency of blasting has been measured with excavator performance. This suggests that there is a strong correlation between excavator performance and the efficient use of blasting energy. This work applies dynamic simulation techniques to develop both economic and technical blasting decision criteria to aid production engineers in making optimal pre-blasting decisions.

It is proposed that blasting criteria can be developed using digging energy and power consumption analysis. Cable shovel kinematics and dynamic modeling combined with formation resistance modeling has been applied to build a 2-D shovel digging simulator. This simulator predicts, among other things, the digging energy and power consumption. The economic blasting decision criterion is based on the cost differential while the technical criterion is based on the shovel hoist and crowd motor power predictions. The models are validated using the P&H 2100BL (small) and 4100TS (large) shovels operating in oil sands material. The results indicate that pre-blasting results in about 15¢/t extra cost, and thus, not economically viable in oil sands. This result is independent of the shovel size. Sensitivity analysis indicates that this decision does not change with all feasible oil sands bulk densities and ±20% change in electrical energy and blasting costs. This work is significant as it makes it possible for planning engineers to decide a priori the excavation strategy without expensive trials.

1. Introduction

Shovel excavation constitutes a significant component of production costs in any surface mining operation. The percentage of loading costs in surface mining operations is between

[1] Assistant Professor, Department of Mining & Nuclear Engineering, University of Missouri-Rolla, USA

3% and 35%, of the total operating costs with a mean of 15% (Hustrulid. and Kuchta, 1995) (Hustrulid and Kuchta, 2006). For instance, at Syncrude's Aurora mine loading and hauling costs were estimated to be 40% of total operating costs (Syncrude, 1996). Consequently, efficient utilization of cable shovels is a key component of cost reduction in surface mining. The connection between shovel loading efficiency and blasting performance is undisputed. In fact, several researchers have used shovel performance as a measure of blast performance (Williamson et al., 1983; Mol and Leung, 1987; and Frimpong et al., 1996). Yet, the discussion of blasting optimization is limited to the effect of blasting on comminution processes. The supposed system optimization of blasting, considers the effective use of blasting and comminution energy in reducing material sizes. However, this ignores the role of shovel loading energy in the system.

This paper proposes that analysis of the energy and power consumption during shovel excavation can be applied to develop blasting criteria. The objective is to develop such criteria based on dynamic simulation modeling of the excavation process. These criteria can be used to determine whether a particular formation with a given shovel needs to be blasted before shovel excavation. Shovel kinematics and dynamics models together with dynamic formation resistance models have already been developed by the authors (Frimpong and Awuah-Offei, 2005; and Awuah-Offei et al., 2005). These models and appropriate numerical models and algorithms have been combined to build a cable shovel simulator (CSS) in the MATLAB/SIMULINK (Mathworks, 2004) environment. These dynamic models are applied to predict the digging energy during a digging cycle. Subsequently, blasting criteria are developed based on the power and energy consumption predictions.

This study is a pioneering effort in applying dynamic simulation results to assist mine planning engineers to decide whether to blast or not before excavation. Its successful implementation will save the mining industry significant costs on designing and implementing trials to determine whether the selected shovel is capable of efficient excavation with or without blasting. The following section describes the pre-blasting decision criteria modeling. This is followed by a discussion on cable shovel energy modeling. Next, a numerical example in oil sands is presented and the results discussed. The last section presents the conclusions and recommendations for future work.

2. Pre-blasting decision criteria modeling

Figure 1 shows a conceptual model of the critical formation resistance during shovel excavation. The formation resistance to soil cutting during excavation (f4) depends of the formation properties. The formation resistance varies with formation properties (or different formations). Conceptually, the total excavation costs per unit of material with blasting should decrease as the formation becomes more difficult to dig. Conversely, the cost per unit without blasting should rise with increasing formation resistance. Therefore, the unit cost differential should also decrease with increasing formation digging resistance. At the point where the cost differential becomes negative then continued in-situ excavation will lead to losses incurred to the operation.

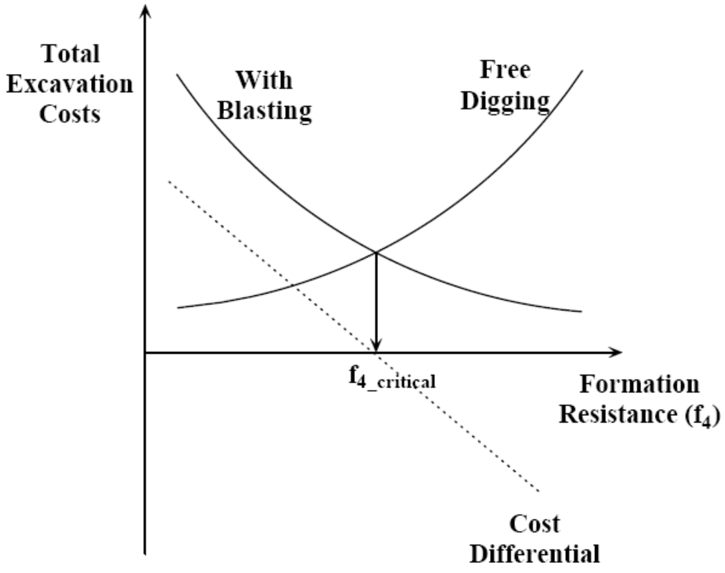


Figure 1 Conceptual model of critical formation resistance

Equation (1) shows the conceptual economic model to determine the critical formation resistance. BC is the blasting cost/tonne and ΔEC, and E, the digging energy per tonne. Equation (1a) shows the loss incurred by management when the pre-blasting is adopted with any formation resistance below the critical formation resistance. This formulation of the critical formation resistance serves as one of the pre-blasting decision criteria presented in this work.

$$\text{Unit cost differential (\$/tonne)} = \text{BC} - \Delta\text{EC} \tag{1a}$$

$$\Delta\text{EC} = \text{Energy Cost/KW-h} \times \frac{\Delta\text{E}}{\text{Payload}} \tag{1b}$$

Alternatively, the critical formation resistance can be viewed in the light of machine health. Both experimental and analytical work shows that the formation diggability (ease of digging) is directly proportional to the shovel motor power consumption (Patnayak et al., 2005; and Awuah-Offei, 2005). However, the power consumption is also proportional to the digging rate (Awuah-Offei, 2005). Therefore, for the same digging strategy, the peak power consumption corresponds to the peak formation resistance. Thus, the peak formation resistance can be correlated to a peak allowable power consumption. This allowable peak need not be the peak power rating of the shovel motors. Dipper and teeth wear, productivity, and shovel downtimes are some of the factors to consider in determining this peak allowable power consumption.

Equation (2) shows the model for technical formation resistance determination. H and C are the hoist and crowd forces, respectively; η^h and η^c [2] are the hoist and crowd motor

[2] The efficiencies here refer to the combined motor energy efficiency and the mechanical efficiency of the mechanism by which the force is delivered.

efficiencies, respectively; $\dot{r}_3$ and $\dot{r}_4$ are the hoist and crowd digging speeds, respectively; and P^h_{peak} and P^c_{peak} are the peak allowable hoist and crowd motor power consumptions, respectively. Equation (2) shows that the formation should be excavated without pre-blasting so long as the hoist and crowd power consumption do not exceed the peak allowable power consumption. If this is the case, then blasting will be a preferred option.

$$\frac{H\dot{r}_3}{\eta^h} \le P^h_{peak} \tag{2a}$$

$$\frac{C\dot{r}_4}{\eta^c} \le P^c_{peak} \tag{2b}$$

These criteria are useful only if a means is found to estimate the power and energy consumption during excavation. This is made possible by the modeling discussed in the next section and the dynamic models of the CSS.

3. Cable shovel energy modeling

Equation (3) shows the model of the total digging energy used in this work. The two terms represent the hoist and crowd energies respectively. Over the digging cycle (lasting t seconds), the digging energy can be computed with Equation (3). The digging power consumption is given by Equation (4). comment

$$E = \int_0^t H\dot{r}_3 dt + \int_0^t C\dot{r}_4 dt \tag{3}$$

$$\text{Hoist Power} = \frac{H\dot{r}_3}{\eta^h} \tag{4a}$$

$$\text{Crowd Power} = \frac{C\dot{r}_4}{\eta^c} \tag{4b}$$

To obtain the hoist and crowd forces, shovel kinematics and dynamics models have been developed using vector loop analysis and the simultaneous constraint method, respectively (Frimpong and Awuah-Offei, 2005; Awuah-Offei et al., 2005; and Awuah-Offei, 2005). The vector loop used in the shovel kinematics model is shown in Figure 2 whereas the free body diagram in Figure 3 is the basis of the shovel dynamics model. The dynamics model is based on resolving the forces in the x- and y- coordinate directions and resolving the moments about the center of mass of the crowd arm. Three extra equations are obtained by direction constraints of the hoist, crowd and normal reaction forces. The acceleration characteristics of the crowd arm are then computed to provide two more equations. In all, a system of ten differential-algebraic equations constitutes the model (Frimpong and Awuah-Offei, 2005; Awuah-Offei et al., 2005; and Awuah-Offei, 2005). The equations are solved with a combined Runge-Kutta (4, 5) embedded algorithm and a linear solver. The linear solver applies Gaussian elimination with partial pivoting.

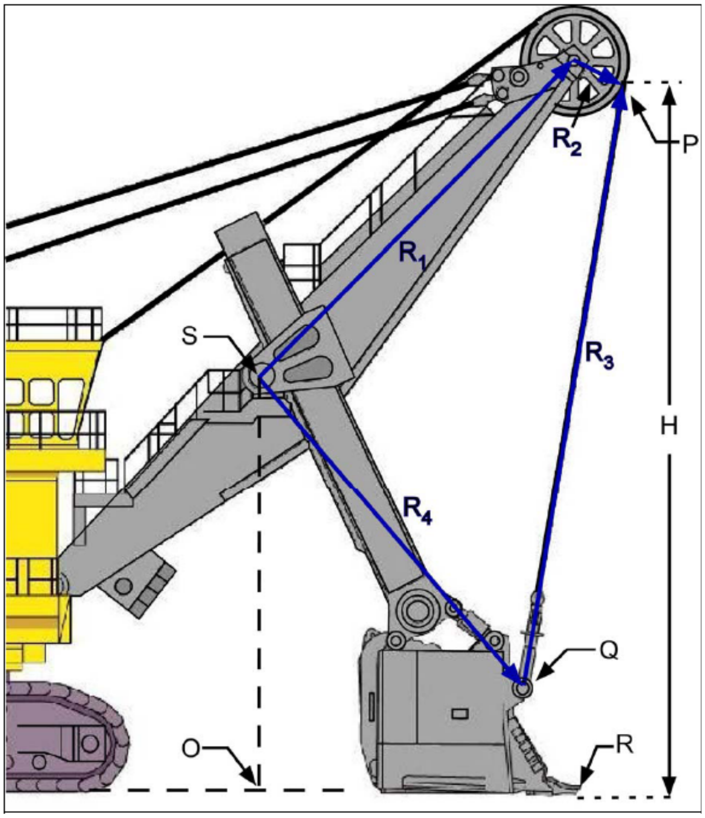


Figure 2 Vector loop for cable shovel kinematics.

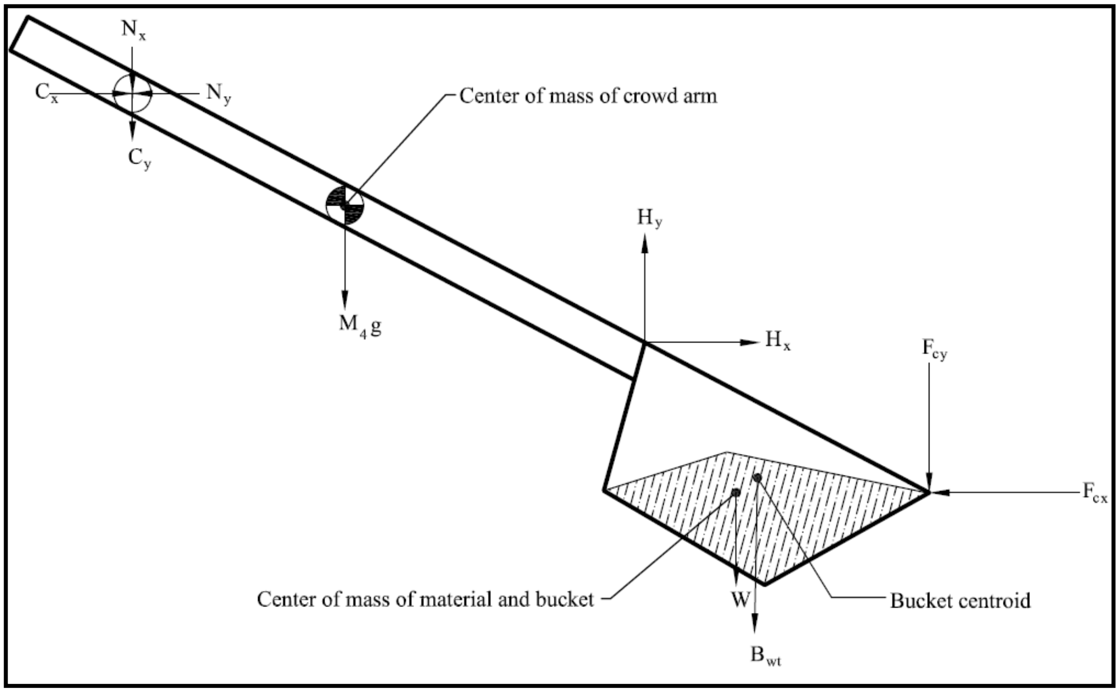


Figure 3 Free body diagram of cable shovel crowd arm.

4. Oil sands application

Oil sands material properties are used in these simulation experiments with a data from two P&H shovel models to validate the economic criterion. These are the 2100BL (small) and 4100TS (large) models with data shown in Tables 1 and 2, respectively[3]. Using these two shovels provides a means to assess the impact of shovel size on the decision criterion. All the simulations are done with constant crowd and hoist speeds of 0.25 ms-1 and 0.7 ms-1, respectively. The oil sands data used in this study comes from results published by Dusseault and Morgensten (1978). Table 3 shows the physical properties of the tested oil sands samples and Table 4 shows the geotechnical properties. The data in Table 4 is then provided as input data for in-situ oil sands for the CSS.

Table 1 P&H 2100BL shovel input data.

Data Type	Parameter	Value
Dipper dimensions	Bucket capacity [m^3]	12.997
	Bucket mass [kg]	21,205
	Lip to latch keeper (front) [m]	2.78
	Bucket inside width [m]	2.74
	Bucket internal angle, υ	90°
	Bucket fill factor	0.95
Initial position parameters	Boom length [m]	15.24
	Boom sheave radius [m]	1.04
	Boom point sheave height (H) [m]	15.57
	Boom angle	45°
	Effective crowd arm length [m]	8.84
	Crowd arm mass [kg]	16,329
	x-coordinate of starting point [m]	4
	y-coordinate of starting point [m]	0

Table 2 P&H 4100TS shovel input data.

Data Type	Parameter	Value
Dipper dimensions	Bucket capacity [m^3]	44.3
	Dipper mass [kg]	680,810.4
	Lip to latch keeper (front) [m]	3.175
	Bucket inside width [m]	4.7244
	Bucket internal angle, υ	90°
	Bucket fill factor	0.95
Initial position parameters	Boom length [m]	21.34
	Boom sheave radius [m]	1.2192
	Boom point sheave height (H) [m]	20.11
	Boom angle	45°
	Effective crowd arm length [m]	10.668
	Crowd arm mass [kg]	38,101.76
	x-coordinate of starting point [m]	4
	y-coordinate of starting point [m]	0

[3] The data on these shovels are modified for proprietary reasons.

Table 3 Physical properties of simulated oil sands.

PROPERTY	VALUE
Permeability (Laboratory)	10-5,000 millidarcys air permeability
Mineralogy	90-98% Quartz 1-5% feldspar 0-3% muscovite 0-4% clay minerals <1% accessory minerals
Approximate bitumen content ranges	0-1%: clay shales and clayey silts 0-10%: sandy silts and silty sands 8-16%: fine- to medium-grained sands 12-16%: coarse-grained sands
Bitumen viscosity	700 Pa s, 90 km north of Fort McMurray 60 kPa s, 2 km west of Fort McMurray
Grain shape	Coarse- and medium-grained sands; well-rounded to sub-angular Fine-grained sands and silts; sub-angular to angular

The next step is to determine which of the material properties is significant in determining the digging energy consumption. Since oil sands cohesion is negligible, cohesion is excluded from the screening experiment. Also, if oil sands is assumed to have negligible cohesion before blasting, then the cohesion will be unaffected by blasting. Factorial experimentation with the remaining three properties (internal and external friction angles and bulk density) was conducted to determine which of these factors are significant for the digging energy. Full factorial experimental design for three (3) factors at two levels each was applied with one center point. Such an experimental program requires 23 experiments plus the center point resulting in 9 experiments in all. The high and low levels of the factors were considered as ±20% of the default values given in Table 4. These were the mean values (except for external friction angle) of the measured parameters by Dusseault and Morgenstern (1978). The external friction angle is estimated by taking into account the values for other soils (e.g. Suministrado et al., 1990) and the fact that oil sands is known to be particularly abrasive. For all oil sand experiments, the muckpile slope is assumed to be equal to oil sands repose angle of 30° (McKenna and Shelbourn, 1995). The 23 full factorial standard order matrix for the factors is shown in Table 5. Only these 9 experiments were run since the experimental output (digging energy) does not vary for the same inputs. In this instance, homogeneity of variance and randomization of the experimental runs are of no consequence (NIST/SEMATECH, 2007).

Table 4 Geotechnical properties of simulated oil sands.

SOIL PROPERTY	VALUE
External friction angle, δ	45°
Internal friction angle, φ	50°
Bulk density, ρ [tonnes/m^3]	2.21
Cohesion [KPa]	0

Table 5 23 Full factorial order matrix

RUN	X1 (δ)	X2(φ)	X3 (ρ)
1	-1	-1	-1
2	+1	-1	-1
3	-1	+1	-1
4	+1	+1	-1
5	-1	-1	+1
6	+1	-1	+1
7	-1	+1	+1
8	+1	+1	+1
9	0	0	0

The results of the 9 experiments are shown in Figure 4. The figure shows the digging energy for the different experimental runs. The grouping of the results into two distinct groups with the center point lying in the middle shows that only a single factor affects the digging energy significantly. The two groups represent the low and high setting of this particular factor. The matrix in Table 5 shows that this factor is the bulk density. This is because bulk density is low in the first four experiments and high in the next four. Compared with Figure 4, this easily points to bulk density as the only significant factor affecting digging energy. Figure 4 shows digging energy increases with bulk density. This shows that bulk density is by far the most significant material property affecting shovel digging energy. Consequently, the change in oil sands digging resistance is modeled as a function of bulk density. Oil sands bulk density after good blast results is estimated as 1.44 tonnes/m3 for average oil sands conditions (JPI, 2005). Consequently, it is estimated that there is a 34.84% reduction in oil sands bulk density with good blasting. Table 6 shows the blasting cost and energy data used in the simulation.

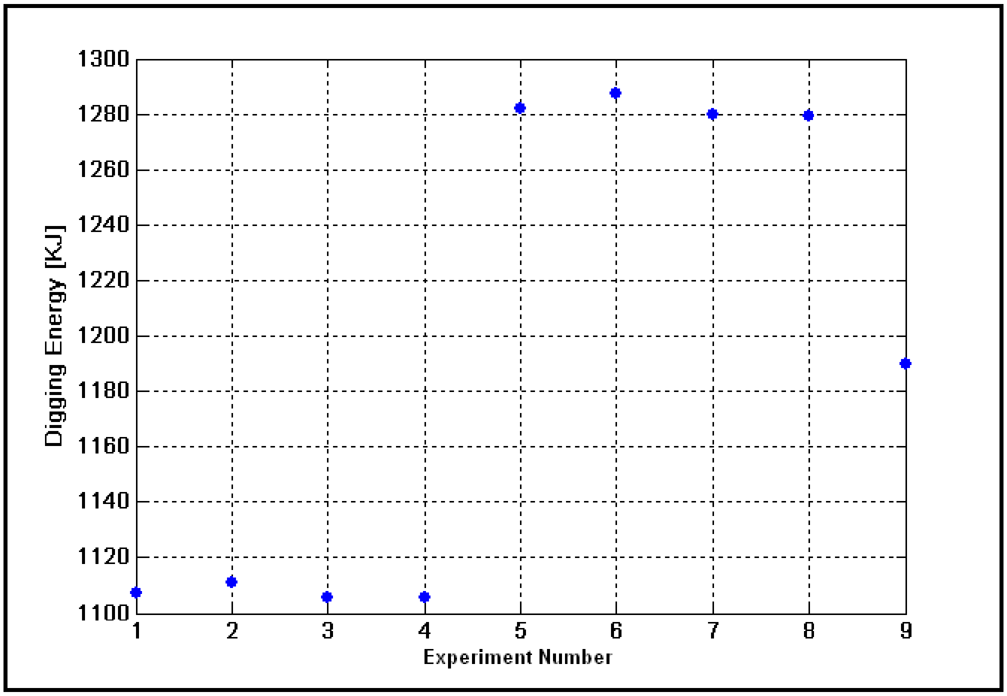


Figure 4 Factorial experiment results.

Table 6 Blasting and energy cost data.

INPUT PARAMETER	VALUE	SOURCE
Drilling & blasting cost (CAD $/ton)	0.20	Hustrulid and Kuchta (2006), p. 98.
Shovel energy efficiency	60%	Engineering estimate
Electric energy cost (US $/KW-h)	0.085	Western Mine Eng. Inc. (2005)

5. Results and discussions

The objective of the oil sands application experiments is to determine whether or not blasting is necessary in varying oil sands formation. This was to illustrate the power of the developed criteria in assessing blasting in any formation once data is available. From the model in Equation (1), when the unit cost differential is less than zero, blasting will be more economical. The models are simulated to determine whether there exists this critical bulk density for the oil sands formation for a small (P&H 2100BL) and a large (P&H 4100TS) shovel. Figures 5 to 8 show the results of the P&H 2100BL while Figures 9 to 12 show the P&H 4100TS results. The experiments were conducted to find the unit cost differential for feasible bulk densities from the literature (1.79922 to 2.4395 tonnes/m3) [**Error! Reference source not found.**]. Figure 5 shows the energy predictions from the CSS for both in-situ and blasted material. The figure shows that both digging energy and energy difference decreases with increasing bulk density. The decreasing energy difference is in line with the conceptual model in Figure 1 though the decreasing digging energy appears to contradict it. Total digging energy increases with increasing bulk density (increasing formation resistance) (see Figure 4) but the increasing payload outweighs that increase thus decreasing the energy per tonne. Figure 6 shows the P&H 2100BL simulation results. Even though the unit cost differential decreases from 15.07¢/tonne to 15.05¢/tonne it does not fall below zero within the feasible oil sands bulk density range. Consequently, one can conclude that blasting is not required in oil sands excavation. This was to be expected since blasting in the Athabasca oil sands is rarely used by production engineers. This validates the decision criterion in this study and is thus, a significant contribution.

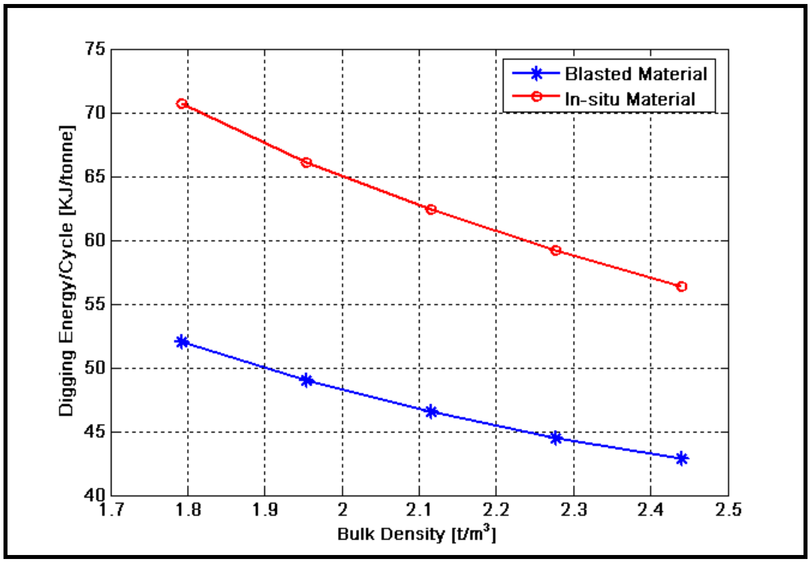


Figure 5 Unit digging energy per cycle for 2100BL.

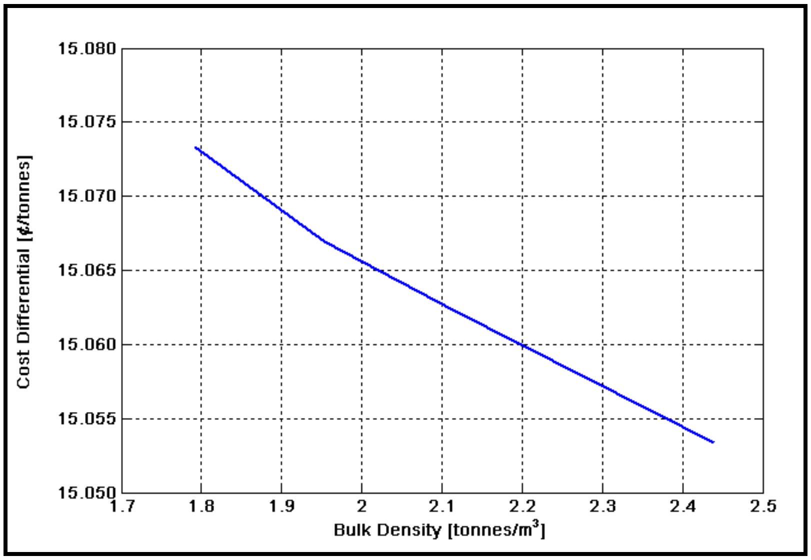


Figure 6 Cost differential against bulk density for 2100BL.

Also, sensitivity analysis was carried out to see how sensitive this decision is to blasting and energy costs. Figure 7 shows the results of varying the energy costs from -20% to +20% of the estimated electrical energy cost for different bulk density values. The figure clearly shows that the unit cost differential decreases with both energy cost and bulk density. This is to be expected as the pre-blasting option becomes more attractive with increasing electrical energy cost and bulk density. Increasing bulk density means increasing energy consumption during digging, and thus, when both parameters increase, pre-blasting becomes more attractive. Yet, even a 20% increase in electrical energy cost will not make the pre-blasting option attractive as the cost differential is still positive over the entire domain. Figure 8 shows the unit cost differential sensitivity to blasting costs for different bulk densities within the domain. The figure shows a much more significant decrease in unit cost differential as the blasting cost decreases compared to increases in energy costs and bulk density. It is obvious from Figures 7 and 8 that the most important parameter is the blasting cost. Nonetheless, over the domain of bulk densities, in-situ digging is still preferred over pre-blasting and digging.

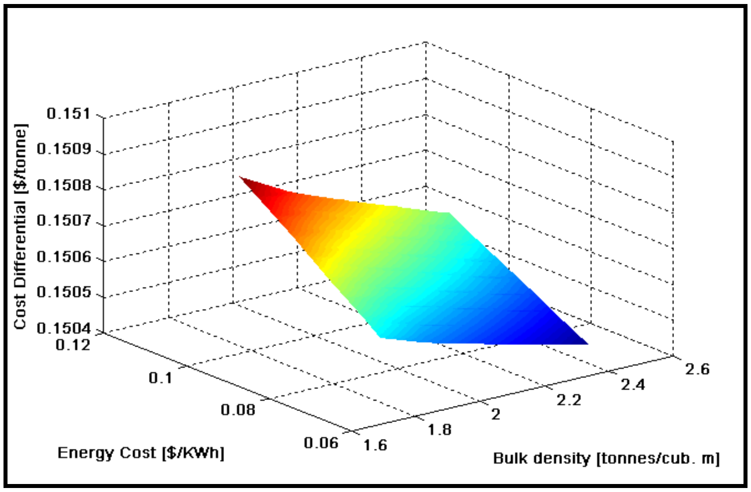


Figure 7 Cost differential sensitivity to energy costs for P&H 2100BL.

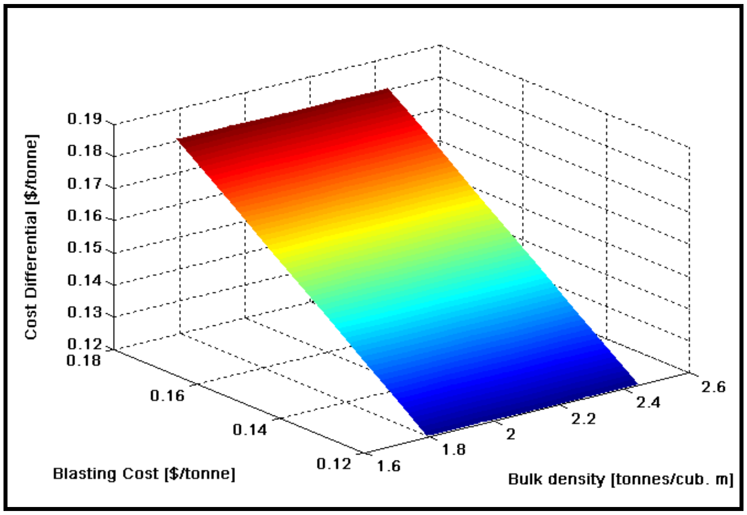


Figure 8 Cost differential sensitivity to blasting costs for P&H 2100BL.

The low sensitivity of the unit differential cost to blasting and electrical energy costs is because of the relatively small difference between energy consumption with or without blasting. For example, for the base case bulk density (2.21 tonnnes/m3) the energy per tonne is 60.76 and 45.31 KJ/tonne for in-situ and blasted material, respectively. This constitutes an energy differential of 15.44 KJ/tonne (0.0043 KWh/tonne). Since most blasting energy goes to reduce formation cohesion, blasting in a cohesionless formations (like oil sands) does not have the same impacts as blasting in other formations. This low energy differential means that unless blasting cost decreases to 0.0275¢/tonne (effectively zero) or electrical energy cost increases to $ 46.37/KWh blasting will not be attractive. These figures represent a 99.82% decrease and a 54,448% increase in blasting cost and energy cost, respectively.

Figures 9 to 12 show the same results with similar trends for the P&H 4100TS. Figure 9 shows the energy predictions from the CSS. Again, both digging energy and energy difference decreases with increasing bulk density. Figure 10 shows the unit cost differential plotted against bulk density. The cost differential decreases from 15.10¢ to 15.07¢. This shows again that, whereas in-situ digging becomes less attractive with increasing bulk density, it is still more attractive within the domain. Figure 11 shows the unit cost differential sensitivity to electrical energy costs over the bulk densities domain. The same trend is repeated with differential cost decreasing with increasing energy cost and bulk density. However, it is still above zero thus indicating that in-situ digging is preferred for the larger shovel also. Figure 12 shows a surface plot of the unit differential cost as a function of bulk density and blasting cost. The differential cost reduces more significantly with decreasing blasting costs compared to both bulk density and energy costs. Yet, over the domain all the values are still positive indicating that for the larger shovel also, in-situ digging is the preferred excavation strategy.

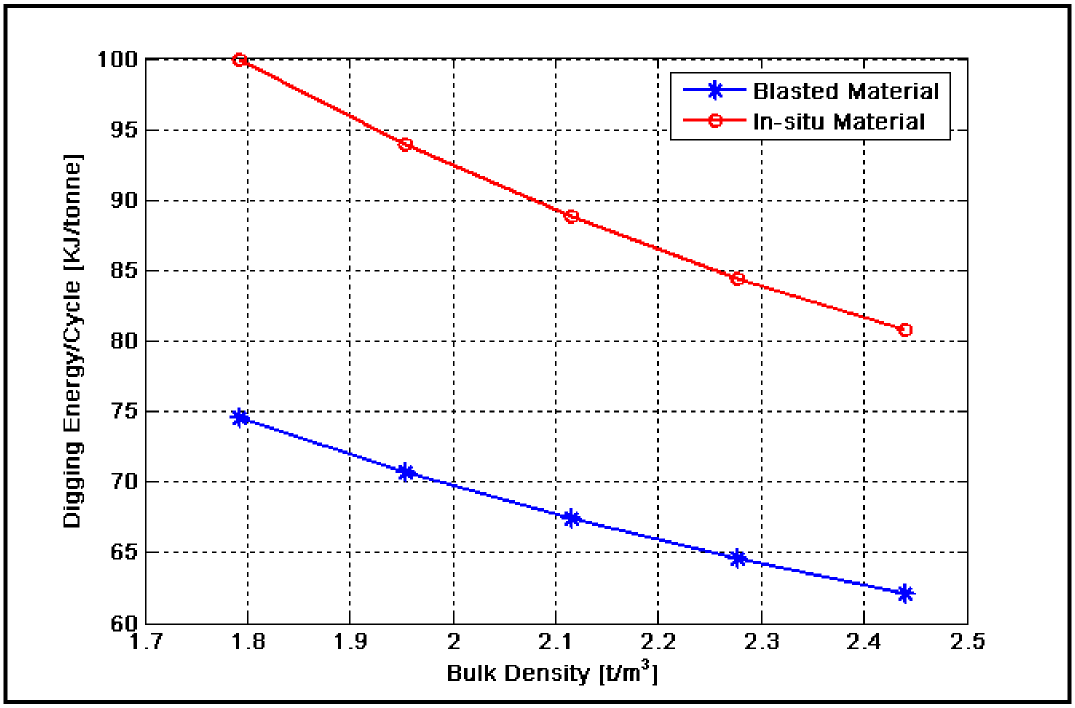


Figure 9 Unit digging energy per cycle for 4100TS

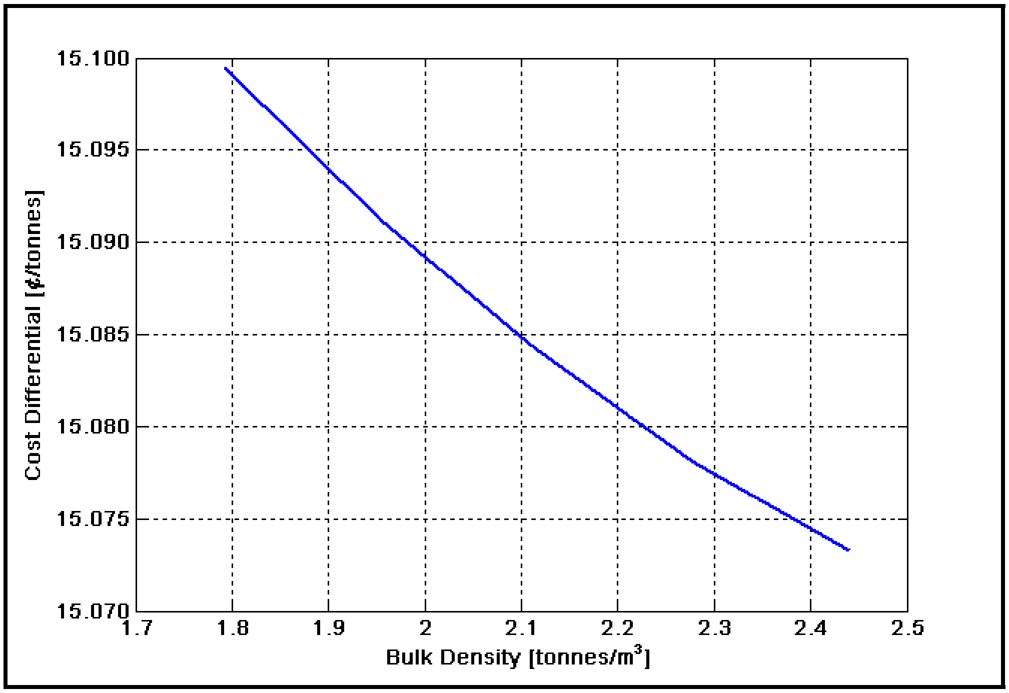


Figure 10 Cost differential against bulk density for P&H 4100TS

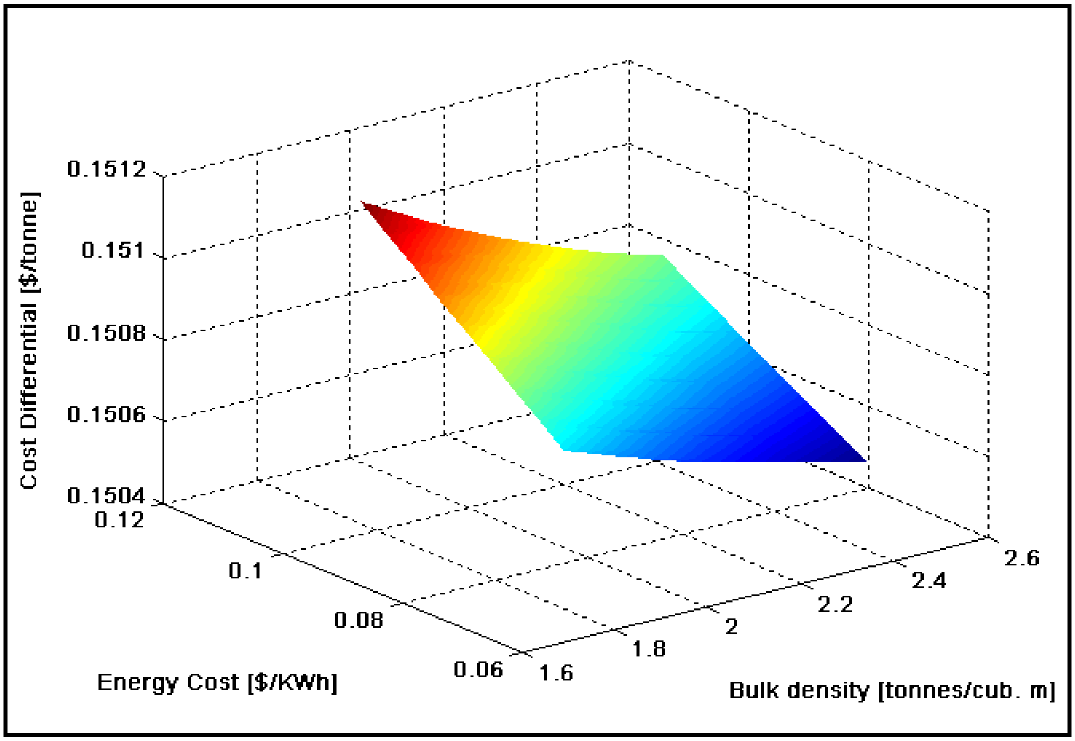


Figure 11 Cost differential sensitivity to energy costs for P&H 4100TS

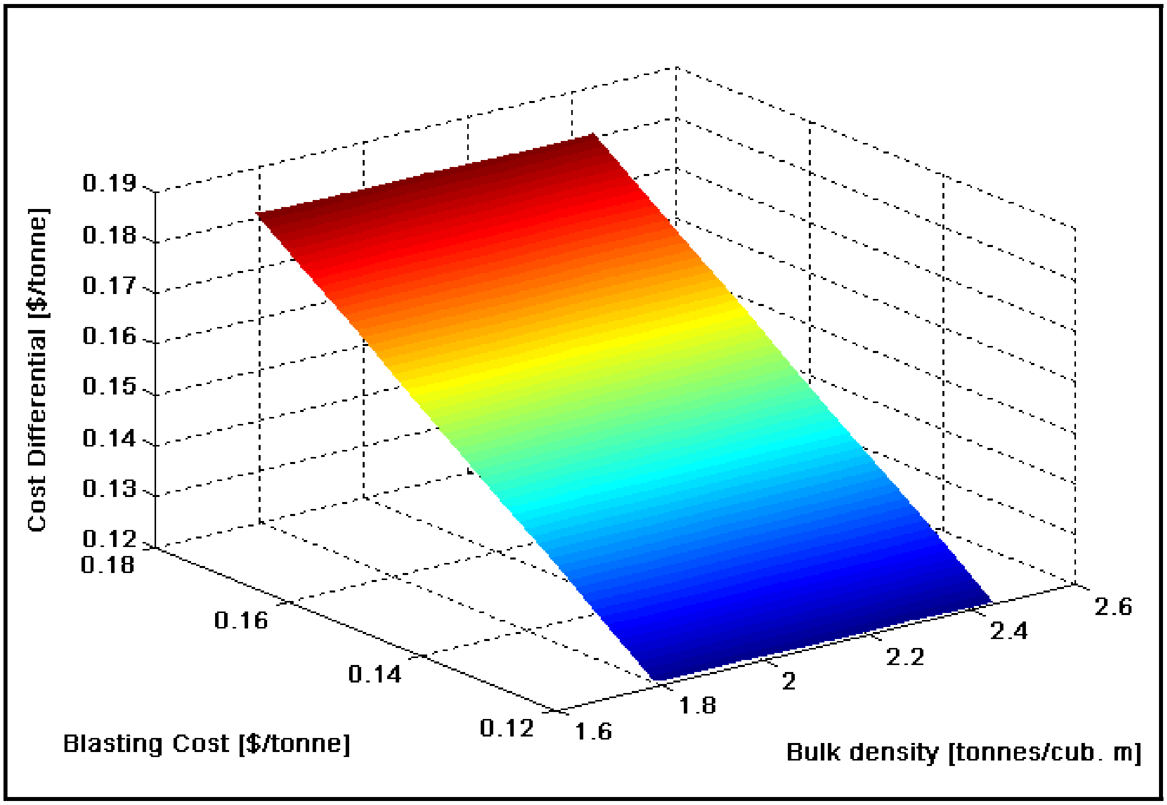


Figure 12 Cost differential sensitivity to blasting costs for P&H 4100TS

A similar low sensitivity of unit differential cost to blasting and electrical energy costs is observed because of the relatively small difference between energy consumption with or without blasting. For the base case bulk density (2.21 tonnnes/m3) the unit energy is 86.19 and 65.71 KJ/tonne for in-situ and fragmented material, respectively. This constitutes an energy differential of 20.49 KJ/m3 (0.0057 KWh/tonne). This low energy differential implies that unless blasting cost decreases to 0.0365¢/t (effectively zero) or electrical energy cost increases to $ 34.95/KWh blasting will not be attractive. These figures represent a 99.76% decrease and a 41,020% increase in blasting cost and energy cost, respectively.

The results show that the blasting decision does not depend on the shovel size. Oil sands excavation should be done without pre-blasting regardless of the shovel size. It must be noted however, that this analysis does not take into account the additional cost due to increased teeth and bucket wear and any increased revenue (if grade correlates with increased bulk densities). Consequently, the conclusions should be applied with regard to these limitations.

6. Conclusions and recommendations

The objective of this work was to develop objective blasting decision criteria based digging energy and power analysis. Shovel kinematics and dynamics modeling as well as formation resistance modeling by the authors have been applied to develop a cable shovel simulator (CSS). This simulator can predict the digging energy and power consumption for a given shovel and formation. Pre-blasting decision criteria have been developed based on the power consumption and the cost differential which works with the CSS. The economic decision criterion has been validated with a small (P&H 2100BL) and a large (P&H 4100TS) operating in oil sands.

The results show that blasting in the Athabasca oil sands is unnecessary. For oil sands bulk density of 2.21 tonnes/m3, there is an additional 15¢/tonne cost for pre-blasting. This result is not significantly sensitive to oil sand bulk density, electrical energy costs or blasting cost. In fact, over the known feasible range of bulk densities the decision is the same. It will take a 99.81% decrease in unit blasting cost for the P&H 2100BL shovel, and a 99.77% decrease for the P&H 4100TS shovel, to make blasting economically superior. Also, unless there is a 54,448% increase in unit electrical energy cost for the P&H 2100BL, and a 41,020% increase for the P&H 4100TS, the decision would not change. This is because the difference in digging energy consumption between in-situ digging and digging with pre-blasting is not high enough.

It is recommended that future work explore the effect of weather (and thus temperature) changes on oil sands bulk density since it is the most important material property. Also, the effect of oil sands freezing during winter on cohesion should be explored. This will ensure that the conclusions from this work will be applicable all year round. Also the performance of the criteria in formations with significant cohesion will be explored in future work.

7. References

Awuah-Offei, Kwame (2005), *Dynamic Modeling of Cable Shovel Formation Interaction for Efficient Oil Sands Excavation*, Unpublished, PhD Dissertation, University of Missouri-Rolla, Rolla, MO, USA 147pp.

Awuah-Offei, Kwame, Frimpong, Samuel and Askari-Nasab, Hooman (2005), "Dynamic Simulation of Cable Shovel Specific Energy in Oil Sands Excavation", Computer Applications in the Minerals Industry (CAMI 2005), Banff, Alberta, Canada, October 31-November 3, 2005.

Dusseault, M. B. and Morgenstern, N. R. (1978b), "Shear Strength of Athabasca Oil Sands", *Canadian Geotechnical Journal, Vol. 15*, National Research Council, Ottawa, Canada pp. 216-238.

Frimpong, Mensah, Kabongo, K. K. and Davies, Clive-Workman (1996), "Diggability, a Measure of Dragline Effectiveness and Productivity", *Proc. Of the Annual Conference on Explosive and Blasting Technique*, International Society of Explosive Engineers, Cleveland, OH, USA pp 95-105.

Frimpong, Samuel and Awuah-Offei, Kwame (2005), "Efficient Cable Shovel Excavation in Surface Mines", *International Conference on Energy, Environment and Disasters (INCEED 2005)*, UNCC, Charlotte, NC, July 24-30, 2005.

Hustrulid, W. and Kuchta, M. (1995), *Open Pit Mine Planning and Design, Vol. 1,* A. A. Balkema, Rotterdam, The Netherlands, p. 95.

Hustrulid, W. A. and Kuchta, M. (2006), *Open Pit Mine Planning and Design,2nd ed., Vol. 1,* Taylor & Francis, London, U.K., p. 107

JPI: Geo-Industry Engineering Consultants (2005), Hydraulic Shovel Performance Analysis, Edmonton, AB, Canada.

McKenna, G. T. and Shelbourn, M. A. B. (1995), "Instrumentation of an Advancing Mine Slope in the Athabasca Oil Sands", *Mine Planning and Equipment Selection 1995*, A. A. Balkema, Rotterdam, The Netherlands pp. 989-997.

Mol, O., Dannel R. and Leung, L. (1987), "Studies of Rock Fragmentation by Drilling and Blasting in Open Cut Mines", *Rock Fragmentation by Blasting: Second International Symposium*, Society of Experimental Mechanics, Keystone, Colorado pp 381-392.

NIST/SEMATECH, *e-Handbook of Statistical Methods*, http://www.itl.nist.gov/div898/handbook/, (Accessed December 12, 2007).

Patnayak, Sibabrata, Tannant, Dwayne, Parsons, Ian and Valle, Victor Del (2005), "Performance Monitoring of Electric Shovels Digging Oil Sands", *Computer*

Applications in the Minerals Industry (CAMI 2005), Banff, Alberta, Canada, October 31-November 3, 2005, Proceedings on CD.

Suministrado, D. C., Koike, M., Konaka, T., Yuzawa, S. and Kuroishi, I. (1990), "Prediction of Soil Reaction Forces on a Moldboard Plow Surface", *Journal of Terramechanics, Vol. 27(4)*, Elsevier Science Ltd., Great Britain, pp. 307-320.

Syncrude (1996), At Face Slurrying: Provisional Data; Syncrude Canada Ltd., Edmonton, Alberta.

The Mathworks Inc. (2004), MATLAB 7.0, MA, USA.

Western Mine Engineering Inc. (2005), Mine Cost Service: Electric Power, http://www.westernmine.com/westernmine/epcost.htm, (Accessed November 21, 2005).

Williamson, S., McKenzie, C. and O'Loughlin, H. (1983), "Electric Shovel Performance as a Measure of Blasting Efficiency", *Rock Fragmentation by Blasting: First International Symposium,* Lulea, Sweden pp 625-635.

Reliability analysis of truck-shovel systems in mining

Kwame Awuah-Offei[1] and Hooman Askari-Nasab
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

The high cost of modern mining equipment has necessitated sophisticated equipment management in mines globally. In recent years, reliability studies have become a vital tool in mine equipment management. This paper presents a reliability modeling and analysis framework for mining truck-shovel systems that encompass standard industry reliability practices as well as the k-*out-of*-n *nature of the truck-shovel system. The framework has been validated with a gold mining case study covering data on 2 hydraulic shovels and 11 trucks. The results indicate that alternator and electrical, structural, hydraulics, and bucket related faults combined constitute over 74% of the total downtime of the shovels. However, preventive maintenance constitutes less than 2% of the total downtime. For the trucks, the data analysis indicates that preventive maintenance and engine related faults constitute 49% of the cumulative downtime. The reliability analysis indicates that individual trucks are more reliable than the shovels. At 50 hours of operation, the trucks and shovels have average reliabilities of 48.29% and 22.51%, respectively. This trend is confirmed by the average time between failures (34 hours for trucks compared to 80 hours for shovels). This may be attributed to the minimal preventive maintenance on the shovels given the production pressures induced by the lack of excess shovel capacity. The reliability function of the truck and shovel sub-systems is computed based on the reliability functions of the individual units. At 50 hours of operation, the reliability of the shovel sub-system is estimated at 5.06% while that of the truck sub-system is estimated to be at most 2.53%. Using these two estimates, the overall truck-shovel system reliability is estimated at 0.13%.*

1. Introduction

Modern mining ventures require huge investments in equipment. To receive the maximum return from these investments, every available tool has to be applied to manage equipment efficiently. The objective of such equipment management is to ensure high productivity, availability, and low maintenance costs. Campbell (1995) shows that the direct maintenance cost in mining as a percentage of overall production costs is high compared to

[1] Assistant Professor, Department of Mining & Nuclear Engineering, University of Missouri-Rolla, USA

other industries. Since the mid-1980s, reliability analysis techniques have been incorporated into mine management techniques. This increasing use of reliability has been spurred on by the relative ease of data collection using computing technology. Truck and shovel systems are popular in mining because of their versatility, flexibility and haulage range. However, attention has not been paid to standardizing the reliability modeling and analysis of truck-shovel systems as used in mining. In particular, the peculiar k-out-of-n nature of truck-shovel systems has not been adequately described in the literature.

The objective of this paper is to present a reliability modeling and analysis framework for truck-shovel systems in mining that encompass standard industry reliability practices as well as the explicit k-out-of-n nature of the truck-shovel system. A case study of a gold mine will then be used to illustrate the proposed framework. The work, however, is limited to trucks and shovels even though truck-shovel systems in mining include auxiliary equipment such as motor graders and bulldozers. Also, the analysis of trends and independence is restricted to the graphical methods. The work presented here provides a fully functional framework for modeling and analyzing the reliability of any truck-shovel system in mining. Given, the wide use of truck-shovel systems, this will be a valuable tool for mine maintenance engineers and shop managers. It is envisaged that this work will contribute to a broader application of reliability studies in mining. This could potentially improve the profitability of many mining operations with truck-shovel systems.

This paper is organized into six sections. Section 2 presents a review of the literature on reliability analysis of mining systems. Section 3 presents the proposed framework while Section 4 presents the results of using the framework on real-life data from a gold mine. Section 5 is the concluding section whereas Section 6 is a list of the references.

2. Reliability analysis of mining systems

Since the early 1980s, mining companies have introduced computerized maintenance systems (e.g. Jardine and Freudenriech, 1981). This advent of computerized maintenance systems and equipment automation, has allowed equipment reliability techniques to flourish as a mine equipment management tool. Several authors have contributed to the development of reliability analysis techniques as applied to mining equipment. Generally, the reliability analysis process for a repairable system involves data collection, sorting and classification; component failure analysis; testing the data for trends and correlations; computing the reliability depending on the nature of the data; parameter evaluation; and reliability and maintainability analysis (Barabady and Kumar, 2008).

Data for reliability studies come from mine maintenance records, dispatch systems and other sources (e.g. equipment age and waiting times for parts may come from different sources). The data should be collected over a suitable period to provide representative data. Often, mine maintenance personnel have to enter the data into the database and this leads to data *censoring* during data sorting and classification (Ansell and Phillips, 1989; Barabady and Kumar, 2008). Once the data has been sorted chronologically and into appropriate components categories, the time between failures (TBF) and time to repair (TTR) can be computed for the different components. The resulting TBF and TTR can then be used to conduct component failure analysis. The essence of the component failure analysis is to identify the critical components of the system (Paraszczak, 2000; Samanta et al., 2001; Barabady and Kumar, 2008). This can be done even if the machine hours are not available (Vagenas et al., 1997).

The next step in the analysis is to test the data for the assumption of independent and identically distributed (iid) nature of the TBF and TTR data for each component. This can be done using graphs or by hypothesis testing (Kumar and Klefsjö 1992; Vagenas et al., 1997; Nuziale and Vagenas, 2000; Barabady, 2005; Barabady and Kumar, 2008). Generally, the five main stochastic processes which have been used to model repairable systems are: (i) the homogeneous Poisson process; (ii) the non-homogeneous Poisson process; (iii) the branching Poisson process; (iv) the renewal process; and (v) the superimposed renewal process (Ansell and Phillips, 1989). The applicability of any of these processes to modeling the reliability of a particular component depends on the test for data trends and correlation. The majority of the mining reliability studies in the literature have validated the iid nature of the data. There are a few examples, though, that use other stochastic processes besides fitting statistical distributions (Kumar et al., 1989; Kumar and Klefsjö 1992; Barabady and Kumar, 2008). After the reliability of individual components have been computed using the appropriate stochastic process, the parameters are evaluated and reliability and maintainability analysis can be undertaken for the system. It must also be noted that Markov processes can also be used to model failure phenomena (Ansell and Phillips, 1989; Yuan and Grayson, 1995; Samanta et al., 2004).

The reliabilities of the individual components ($R_i(t)$) can be used to estimate the reliability of the system ($R_{sys}(t)$). A series system is one composed of *n* components (or subsystems), the failure of any of which will cause a system failure. A parallel system, on the other hand, is composed of *n* components (or subsystems) that perform identical functions, the success of any of which will lead to system success. In other words all the components in a parallel system must failure for the system to fail (Kales, 1998). Equation (1a) gives the reliability of a series system given the reliabilities of the *n* components while Eq. (1b) gives the reliability of a parallel system. A *k*-out-of-*n* system is a system of *n* components (or subsystems) which functions if, and only if, at least *k* of the components function. Parallel systems are 1-out-of-*n* systems and series systems are *n*-out-of-*n* systems. Equation (1c) represents the reliability of a *k*-out-of-*n* system given the reliabilities of the individual components (Boland and Proschan, 1984). $\vec{\varepsilon} = \left[\varepsilon_1 \quad \varepsilon_2 \quad \cdots \quad \varepsilon_n\right]$ is a vector with components equal to zeroes and ones.

$$R_{sys}(t) = \prod_{i=1}^{n} R_i(t) \tag{1a}$$

$$R_{sys}(t) = 1 - \prod_{i=1}^{n} \left[1 - R_i(t)\right] \tag{1b}$$

$$R_{sys}(t) = \sum_{\vec{\varepsilon},\quad \varepsilon_1+\varepsilon_2+\cdots+\varepsilon_n \ge k} \left\{ \prod_{i=1}^{n} R_i(t)^{\varepsilon_i} \left[1 - R_i(t)\right]^{1-\varepsilon_i} \right\} \tag{1c}$$

The discussion of reliability of mining systems in the literature has been limited to series and parallel systems (Vagenas et al., 1997; Nuziale and Vagenas, 2000). In fact, almost all the mining systems modeled have been modeled as series systems (Table 1). The modeling of load and haul fleets as:

1. Series systems is limited in real-life applications because most mining fleets are designed with excess capacity and hence the failure of a single machine will not constitute system *failure*;

2. Parallel systems is limited because load and haul systems often require more than one unit to meet designed productivity. Hence they cannot be 1-out-of-n (parallel) systems; and

3. Series and/or parallel systems do not allow sensitivity analysis of all the possible systems given a fleet of *n* machines.

Table 1 Mining systems described as series system

Source	System description	Comments
Barabady (2005)	Crushing plants modeled 7 subsystems	Reliability estimated as a function of operating time
Barabady and Kumar (2008)	Crushing plant modeled with 6 subsystems	Reliability estimated as a function of operating time
Kumar and Huang (1993)	Underground mine production system modeled with 7 subsystems	Reliability estimated at 100 days using simulation
Mandal and Banik (1996)	Longwall production system modeled as 4 subsystems	Static estimate of the risk to production (not reliability)
Nuziale and Vagenas (2000)	Scooptram modeled with 9 components	Reliability estimate at 100 hrs
Samanta et al. (2001)	Hydraulic shovels modeled with 6 components	Reliability estimated as a function of operating time

However, the calculation of system reliability for a k-out-of-n system is quite cumbersome even for relatively simple cases. For instance, a 9-out-of-11 system will involve evaluating 67 products of 11 numbers (assuming all 11 machines have different reliabilities). Nonetheless, the reliability function of a k-out-of-n system exhibits properties that can be used to estimate useful bounds of the reliability without fully computing all the terms. For instance, if $\sum_{\forall i} R_i(t) \leq k-1$ then assuming all the component reliabilities are equal to the mean (i.e. $R_i = \overline{R}$) results in an upper bound of the system reliability (Boland and Proschan, 1984). If the component reliabilities are assumed to be equal to the mean, then the system reliability follows the Binomial probability law and system reliability is given by Eq. (2).

$$R_{sys} = \sum_{x=k}^{n} \binom{n}{x} \left(\overline{R}\right)^{x} \left(1-\overline{R}\right)^{n-x} \tag{2}$$

3. Reliability modeling of truck-shovel systems

The process of determining the reliability of individual trucks and shovels in the truck-shovel system has been adequately discussed in the literature (Vagenas et al., 1997; Barabady and Kumar, 2008). This process is summarized in the previous section. The model proposed for a truck-shovel system is shown in Fig. 1. Both the truck fleet and the shovel system are modeled as *k*-out-of-*n* systems. Hence, after the individual reliabilities of the trucks and shovels have been determined, the truck fleet and shovel system reliability can be determined as using Eq. (1c). Due to the relatively high capital expenditure associated with shovels, several mines have no excess shovel capacity. In that case, the shovel system becomes a series (*n*-out-of-*n*) system where all the shovels need to be working for the system to be functional. The reliability of the shovel system can then be calculated using Eq. (1a). The two systems should then be considered series systems (as illustrated by Fig. 1) as both need to be working for the truck-shovel system to be functional. The reliability of the truck-shovel system is then calculated using Eq. (1a).

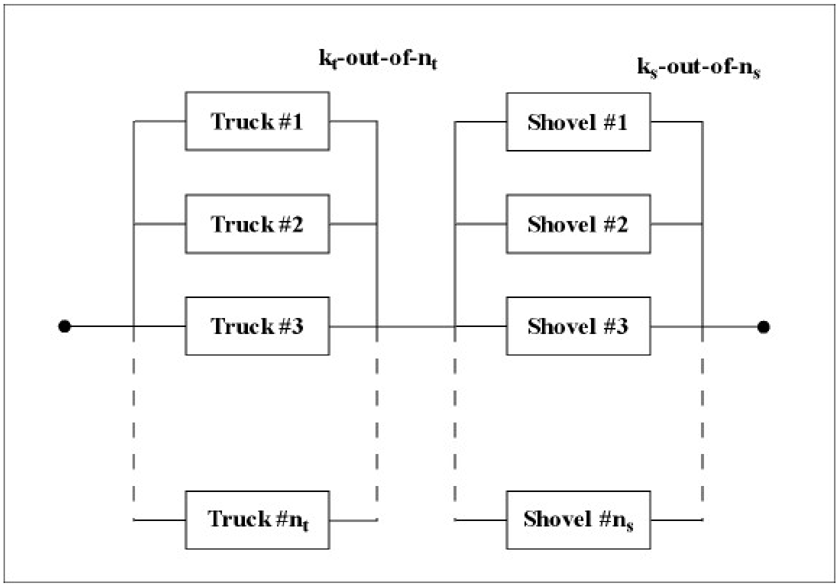


Fig. 1 Truck-shovel reliability model

The challenge posed by the model proposed in Fig. 1 is that modern mines have large fleets and therefore Eq. (1c) becomes cumbersome to compute as illustrated earlier. The temptation will be to find the average reliability of a particular fleet and use Eq. (2) to compute the reliability of the system. This approach however, results in an estimate of the reliability which may be lower or higher than the actual reliability depending on the value of $\sum_{\forall i} R_i(t)$ and *k* (Boland and Proschan, 1984). Consequently, the engineer needs to understand these relationships in order to properly articulate the limitations of the analysis and its impacts on the overall reliability estimate. This approach is illustrated with the case study discussed next.

4. Case Study

4.1. Data Analysis

The data used in this paper to illustrate the proposed reliability modeling framework is from the Damang Gold Mine in the Western Region of Ghana. The mine produces gold from the famous Ashanti Belt. The data includes data on 11 trucks (out of 15 trucks on site) and the two main production hydraulic shovels (there is a third smaller shovel used to augment production from the two) from January 2001 to October 2002 (a period of 22 months). This is consistent with the standard of at least 1 year of data for reliability analysis (Vagenas et al., 1997; Barabady and Kumar, 2008). The data is from the mine maintenance database which is a repository for data entered by shop maintenance personnel. Besides the time down and time up data, the data also included a description of the repair, the sub-assembly, the component code, failure code, and repair type.

The first task was to sort the data into data pertaining to specific equipment since the data was already in chronological order. As would be expected from reliability data some data censoring was done to remove *anomalies* which would have made analysis impossible. The main *anomalies* related to how the maintenance personnel entered data on a unit that had multiple repairs done during the same period in the shop. In some cases, the repairs were entered sequentially creating zero TBFs. In other cases, the data was entered as one repair being nested in the other. In both cases, the repair time was aggregated into one repair time and the repair ascribed to the major (the one that took the most time to repair) fault. There were several entries of zero TTRs. These records were not altered in order to preserve the TBFs that were associated with them.

After sorting and *cleaning up* the data, the TBFs and TTRs were calculated for all equipment units. Then, graphical methods were used to test for trends and correlation in the data. Fig. 2 shows sample results for one of the shovels. As can be seen from Fig. 2 there is no obvious trend (the slope is almost constant) or correlation in the data. All the data exhibited properties similar to the one shown in Fig. 2. Hence, the iid assumption was assumed to be valid for this data set. Next, component failure frequency analysis was done to identify the critical components for both trucks and shovels. The mine breaks down both trucks and shovels into 14 components in the maintenance database. These codes and their descriptions are shown in Table 2. Fig. 3 and 4 show the results of the component failure frequency analysis. Fig. 3 shows that alternator and electrical (AE), structural (FS), hydraulics (HY), and bucket related faults (ME) are the most critical for the shovels. These four components constitute 74% and 79% of the total downtime during the 22 month period for shovel EO 705 and 706, respectively. Any attempt to improve the shovel reliability will have to look at improving the reliability of these components. Service (SV) hours constitute only 2% and 1% of total downtime during the period for shovels EO 705 and 706, respectively. This reflects a general attitude of running the primary production tool as much as possible.

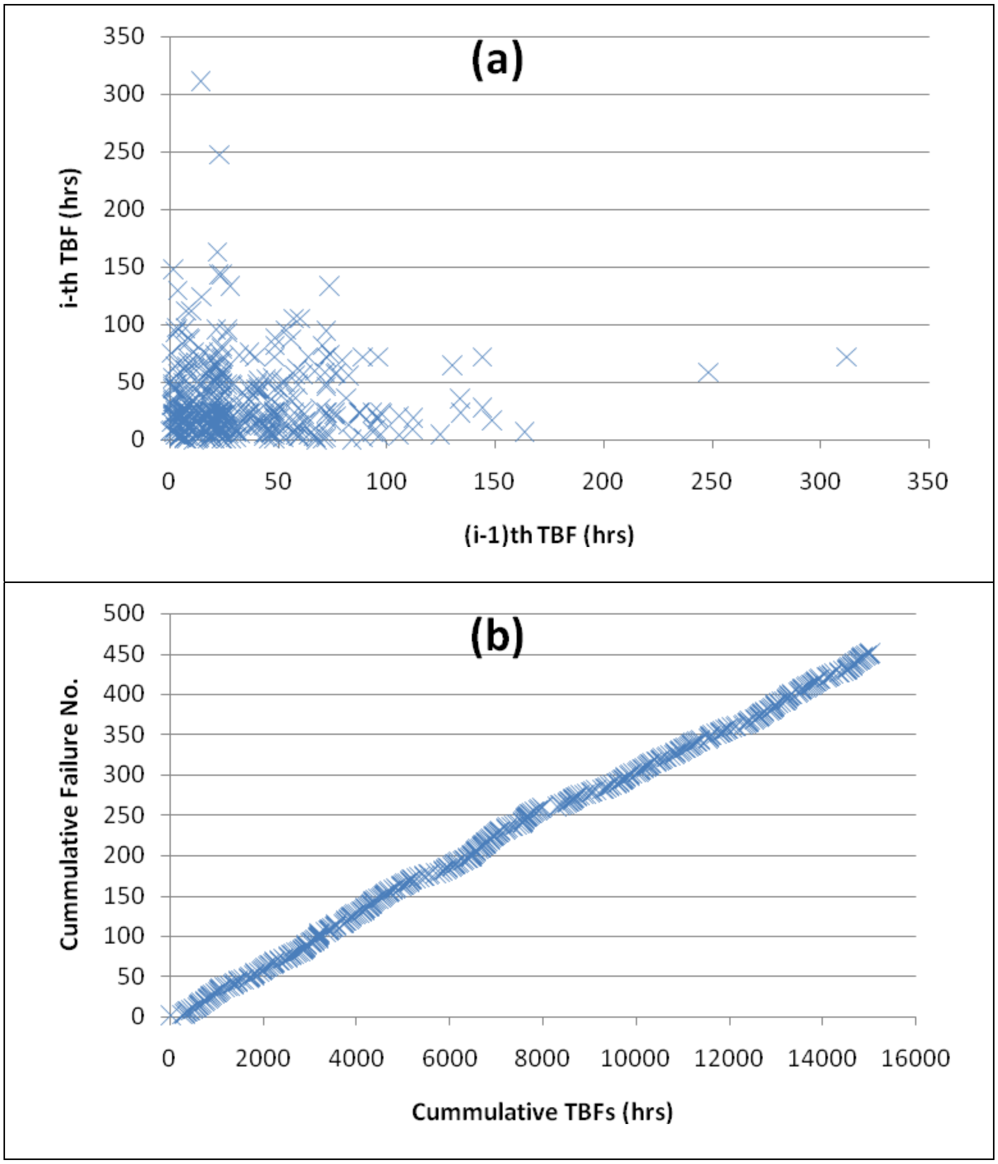


Fig. 2 Sample test for iid for shovel EO705; (a) correlation test; (b) trend test.

Table 2 Component codes

Component Code	Description	Component Code	Description
AC	Air Conditioner	LU	Lubrication
AE	Alternator/Electrical	ME	Bucket (shovel) or Body (truck)
BR	Brakes	MI	Miscellaneous
DR	Drive train (transmission)	PN	Pneumatic s
EN	Engine	SS	Suspension and Steering
FS	Frame/Structural	SV	Service Hours
HY	Hydraulics	TU	Tire (truck) or Undercarriage(shovel)

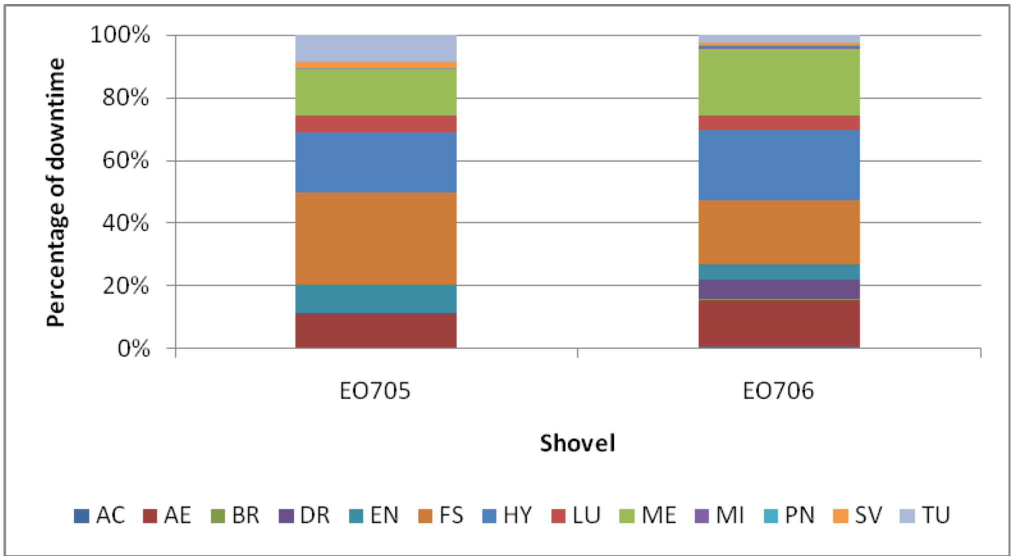


Fig. 3 Shovel breakdowns by component

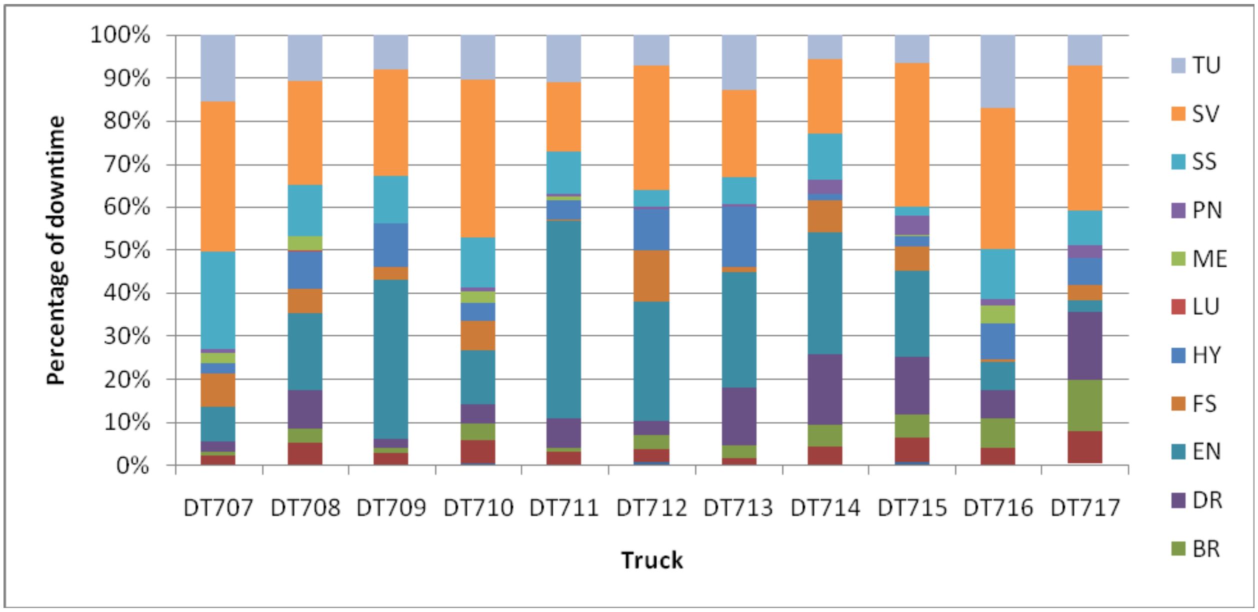


Fig. 4 Truck breakdowns by component

Fig. 4 shows that service (SV) and engine (EN) related faults are the most critical components of the trucks (49% of total downtime). Service hours accounts for 26% of the cumulative downtime of all the trucks while engine faults accounts for 23%. Individually, service hours ranges from 2% to 23% of total downtime for the trucks. This is a significant improvement over the shovels. This reflects the design over-capacity which allows the shop to spend more time on planned maintenance for trucks. For individual trucks, engine related faults range from 3% to 46% of total downtime.

Since the data was assumed to be iid, the Input Analyzer in ARENA 10.0 (Rockwell Software, 2005) was used for fitting probability distribution functions to the data. The Erlang, Exponential, Gamma, Lognormal, Normal, Triangular, Uniform and Weibull two parameter distributions were considered for fitting (Rockwell Software, 2005). Even though the software could fit the Beta distribution to the data, this was eliminated by restricting the selected distributions to non-translated data. Fig. 5 shows sample results of the curve fitting exercise. Table 3 and 4 show the detailed results of the ARENA curve fitting for the trucks while Table 5 shows that for the shovels.

These distributions can then be used to determine the mean time before failure (MTBF) and mean time to repair (MTTR) of the various units. For the reliability analysis, the distributions are used to compute the reliabilities of the different units as a function of operating (mission) time. Fig. 6 and 7 show the reliabilities of the shovels and trucks, respectively. As would be expected, the reliabilities decrease with operating time. Fig. 6 shows that the reliabilities of the two shovels are very similar with EO706 being slightly more reliable. At 50 hours, the reliabilities are 21.73% and 23.30% and only 3.88% and 5.11% at 100 hours for shovels EO705 and EO706, respectively. Using Eqs. (1a) and (1b), the reliabilities of the shovel system at 50 hours was computed assuming a 1-out-of-2 and 2-out-of-2 system. At the Damang mine, these two shovels are the main production shovels. Consequently, a 2-out-of-2 system is better suited than the 1-out-of-2 system. Using this modeling approach, the reliability of the shovel system at 50 hours will be 5.06%.

Fig. 7 shows the truck reliabilities as function of operating time. Once again, the reliabilities are very similar with DT712 being the most reliable overall. Table 6 shows the reliabilities of all the trucks at 50 hours. These reliabilities range from 40% to 53% with an average of 48.29%. Assuming that the mine needs 9 out of the 11 trucks surveyed to be working to meet production targets, we can model the truck system as a 9-out-of-11 system. As indicated earlier, computing the reliability of this system will require computing 67 products of 11 numbers (the truck reliabilities). This will be cumbersome and prone to errors. However, since $\sum_{\forall i} R_i(t)$ (5.31) is less than the *k*-1 (8), we can easily find the upper bound of the system reliability by using Eq. (2). Using $\overline{R} = 0.4829$ and Eq. (2), the truck system reliability at 50 hours will be 2.53% which is a high estimate of the reliability. In contrast, if we had assumed a parallel or series system the system reliability would have been 99.93% or 0.03%, respectively. The assumption of a series system is clearly wrong in this case since the failure of just one unit will not constitute system failure. Assuming a parallel system will also be wrong since just one truck running does not constitute a successful system. This particular assumption provides undue confidence in the system (almost 100% reliability).

Fig. 1 shows that the truck-shovel system is a series system composed of two sub-systems. Having obtained estimates of the reliabilities of the two sub-systems, the overall system reliability at 50 hours can be calculated using Eq. (1b). This results in an overall system reliability of 0.13%. However, this estimate is on the high side given the assumption that all the truck sub-system reliability is a high estimate. Yet, the theoretical work done about the nature of the reliability function of k-out-of-n systems allows us to reduce the complex computations while still obtaining useful results.

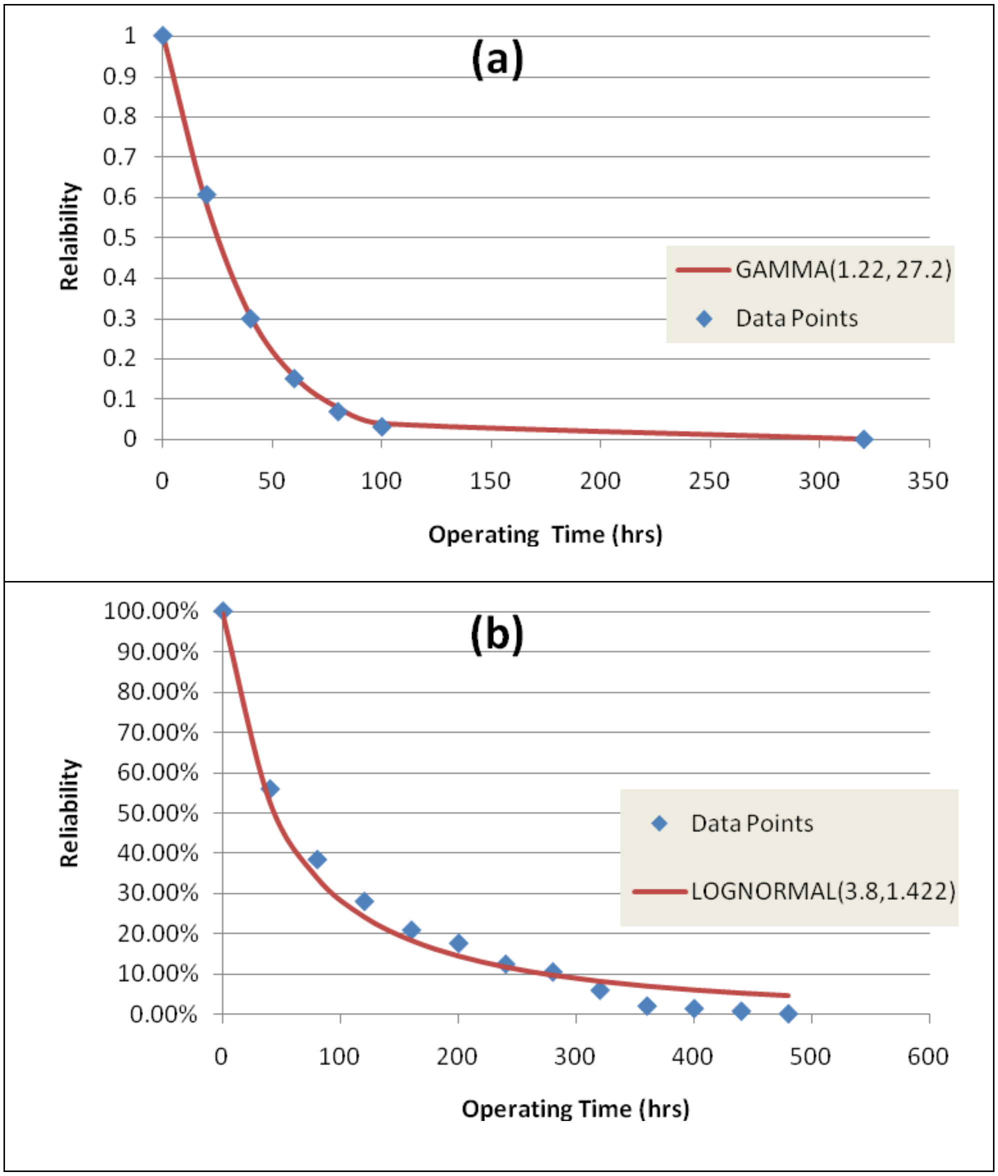


Fig. 5 Sample curve fitting results: (a) Shovel EO705; (b) Truck DT712.

Table 3 TBF summary statistics and curve fitting results for trucks

Statistic	DT707	DT708	DT709	DT710	DT711	DT712	DT713	DT714	DT715	DT716	DT717
Mean	81.84	74.97	76.06	84.29	62.90	95.19	85.01	69.60	83.64	85.47	78.78
Standard Error	6.81	6.90	5.69	7.01	5.82	8.46	7.81	5.64	6.75	7.18	7.05
Median	46.38	34.00	45.50	51.18	32.32	49.09	49.68	33.18	50.50	53.50	39.34
Mode	19.32	24.00	24.00	19.32	24.00	19.32	19.32	24.00	24.00	72.00	19.32
Standard Deviation	91.89	97.28	79.43	93.26	88.48	105.00	102.07	81.09	90.38	94.94	96.64
Kurtosis	2.65	11.07	2.26	10.70	29.71	1.03	6.29	7.84	2.56	5.44	6.20
Skewness	1.72	2.69	1.56	2.53	4.34	1.38	2.34	2.29	1.68	2.07	2.21
Minimum	0.32	0.18	0.50	0.75	0.32	0.68	1.00	1.32	0.32	0.68	0.18
Maximum	439.32	728.50	379.32	695.00	849.00	452.68	619.32	574.43	431.00	576.00	569.50
Sum	14895.37	14918.76	14832.00	14920.17	14529.17	14659.50	14535.90	14406.42	14972.07	14956.75	14811.22
Count	182	199	195	177	231	154	171	207	179	175	188
Distribution	Weibull	Weibull	Weibull	Weibull	Weibull	Lognormal	Lognormal	Weibull	Weibull	Weibull	Weibull
Parameter 1	α = 0.878	α = 0.794	α = 0.923	α = 0.934	α = 0.835	μ = 3.8	μ = 3.9	α = 0.882	α = 0.898	α = 0.911	α = 0.819
Parameter 2	β = 76.6	β = 65.4	β = 73.2	β = 81.6	β = 56.6	σ = 1.422	σ = 1.22	β = 65.2	β = 79.4	β = 81.6	β = 70.4

Table 4 TTR summary statistics and curve fitting results for trucks

Statistic	DT707	DT708	DT709	DT710	DT711	DT712	DT713	DT714	DT715	DT716	DT717
Mean	2.83	3.08	4.31	3.62	3.85	3.73	3.33	3.66	2.93	2.73	3.46
Standard Error	0.32	0.46	0.65	0.46	0.85	0.58	0.51	0.47	0.52	0.32	0.38
Median	1.00	1.00	1.50	1.50	1.00	1.50	1.50	1.00	1.00	1.50	1.50
Mode	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Standard Deviation	4.33	6.55	9.07	6.19	12.88	7.26	6.66	6.82	6.95	4.27	5.22
Kurtosis	15.09	64.14	25.51	26.80	119.28	21.19	22.10	18.53	57.87	13.11	7.70
Skewness	3.26	6.86	4.62	4.29	10.02	4.14	4.34	3.68	6.67	3.31	2.61
Minimum	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Maximum	33.00	72.00	69.00	54.00	168.00	51.50	49.50	55.00	72.00	27.00	31.00
Sum	518.63	615.24	845.00	644.00	893.33	577.50	573.10	760.58	527.75	481.33	654.28
Count	183	200	196	178	232	155	172	208	180	176	189
Distribution	Lognormal	Weibull	Weibull	Lognormal	Weibull	Lognormal	Lognormal	Weibull	Weibull	Weibull	Weibull
Parameter 1	μ = 0.544	α = 0.781	α = 0.710	μ = 0.648	α = 0.682	μ = 0.675	μ = 0.640	α = 0.713	α = 0.746	α = 0.883	α = 0.827
Parameter 2	σ = 1.007	β = 2.43	β = 3.04	σ = 1.097	β = 2.40	σ = 1.070	σ = 0.997	β = 2.59	β = 2.13	β = 2.14	β = 2.97

Table 5 TBF and TTR summary statistics and curve fitting results for shovels.

Statistic	TBF		TTR	
	EO705	EO706	EO705	EO706
Mean	33.24	34.31	1.21	1.07
Standard Error	1.53	1.78	0.21	0.17
Median	24.00	24.00	0.50	0.50
Mode	24.00	24.00	0.00	0.00
Standard Deviation	32.61	37.43	4.53	3.67
Kurtosis	16.41	20.54	155.18	149.64
Skewness	2.97	3.45	11.28	11.42
Minimum	0.18	0.17	0.00	0.00
Maximum	312.00	360.00	72.00	52.68
Sum	15026.15	15131.98	549.85	469.83
Count	452	441	453	441
Distribution	Gamma	Weibull	Weibull	Gamma
Parameter 1	α = 1.22	α = 1.03	α = 0.779	α = 3.26
Parameter 2	β = 27.2	β = 34.7	β = 0.844	β = 0.326

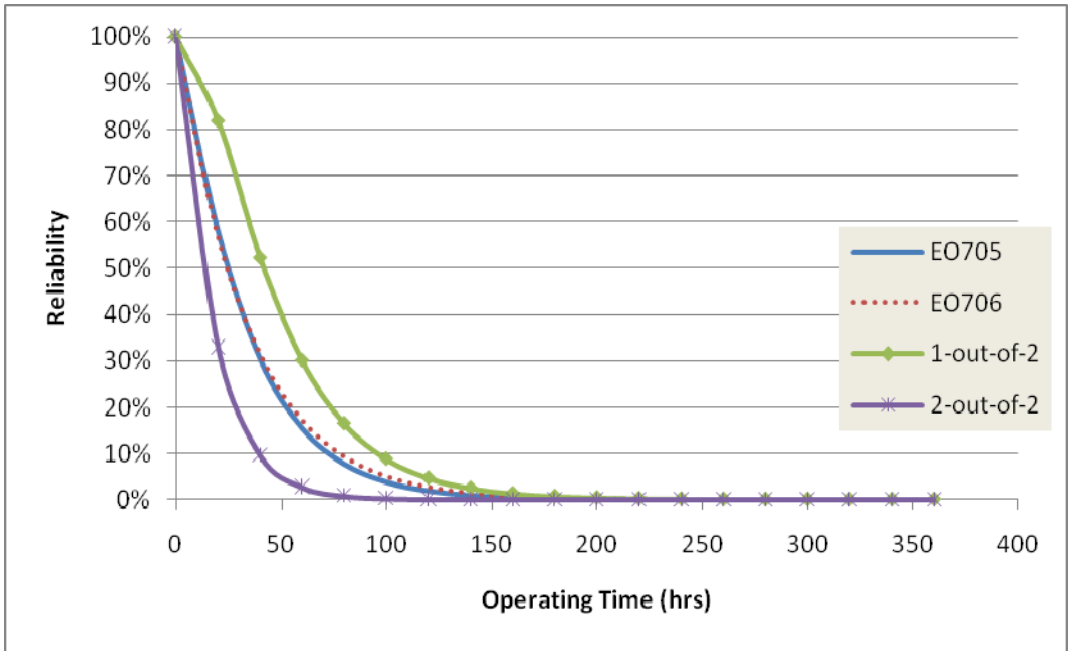


Fig. 6 Shovel reliability.

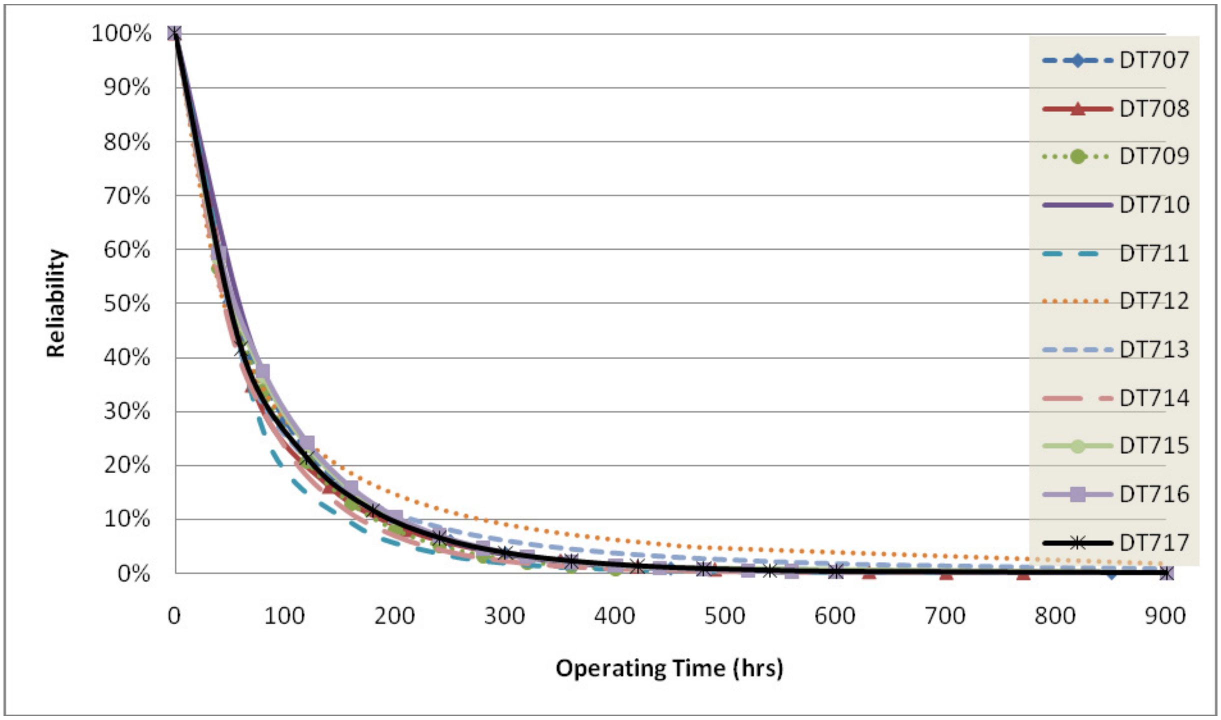


Fig. 7 Truck reliabilities

Table 6 Truck reliabilities at 50 hours operating time

Truck	DT707	DT708	DT709	DT710	DT711	DT712
Reliability at 100 hrs	50.28%	44.57%	49.49%	53.11%	40.59%	46.86%
Truck	**DT713**	**DT714**	**DT715**	**DT716**	**DT717**	**Average**
Reliability at 100 hrs	49.61%	45.33%	51.68%	52.73%	46.97%	48.29%

Fig. 1 shows that the truck-shovel system is a series system composed of two sub-systems. Having obtained estimates of the reliabilities of the two sub-systems, the overall system reliability at 50 hours can be calculated using Eq. (1b). This results in an overall system reliability of 0.13%. However, this estimate is on the high side given the assumption that all the truck sub-system reliability is a high estimate. Yet, the theoretical work done about the nature of the reliability function of k-out-of-n systems allows us to reduce the complex computations while still obtaining useful results.

4.2. Discussions

The individual trucks are more reliable than the shovels. For instance, at 50 hours operating time, the truck reliabilities range from 40% to 53% (Table 6) whereas the shovel reliabilities are 21.73% and 23.30% (Fig. 6). Also, the average TBF for the shovels is 33.78 hours (Table 5) compared to 79.80 hours (Table 3). This may be due to the greater servicing the trucks receive compared to the shovels as indicated earlier. This greater servicing seems to be supported by the difference in average TTR for the shovels and trucks. Whereas the average TTR is 1.14 hours for the shovels, the average TTR for the trucks is 3.41 hours. However, the TTR data alone cannot be enough evidence unless it is combined with labor to provide maintenance man-hours (MMH) data (Paraszczak, 2000). The mine maintenance database did not collect this information and thus this cannot be explored at this time. This may also explain some of the discrepancy between the percentage of total downtime attributable to servicing (SV) for the trucks and shovels (1% to 2% as compared to 26% of cumulative downtime). Suffice to say, there was a definite difference in the tolerance to truck shutdown compared to shovel shutdown.

The 2-out-of-2 shovel system is more reliable than the 9-out-of-11 truck system. This implies that the truck system is the bottleneck in the reliability of the truck-shovel system. This result may seem to contradict the observation that individual trucks are more reliable than individual shovels. However, the resulting systems are so different (2-out-of-2 and 9-out-of-11) that the system reliabilities do not reflect this logic. The lower reliability of the truck system implies that there is a greater probability of having less than 9 trucks operating than having any one of the trucks breakdown. Adding extra trucks will improve the situation (i.e. the system will become a 9-out-of-n_t; where $n_t > 11$). The overall system reliability of 0.13% at 50 hours is low for a mine that operates 20 hours a day. Besides, the increase in the size of the truck fleet other strategies to improve this include improving the preventive maintenance of the shovels in an attempt to improve the shovel reliabilities, and increasing shovel capacity by adding extra shovels of the same capacity. Some sources recommend that preventive maintenance should account for up to 80% of all active maintenance time (Kuruppu and Golosinski, 1998; Paraszczak, 2000). This might be the most feasible near term strategy to improve shovel reliabilities.

5. Conclusions

As mining equipment has increased in size and complexity, the capital expenditure has increased as well. This has caused mine management to desire better management of mine equipment. In recent years, reliability studies have become an important tool in mine equipment management. A framework for reliability modeling and analyses of truck-shovel systems in mining has been presented in this work. The novel aspect of this framework is that it goes beyond the simplistic series and parallel systems that have been used to model mining systems in the past. The limitations of the series and parallel systems in modeling load-haul systems include: (i) Series systems do not account for designed excess capacity that exists in most load-hauls systems; (ii) Parallel systems do not account for the fact that load-haul systems often require more than one unit to meet designed productivity; and (iii) Series and/or parallel systems alone do not allow sensitivity analysis of all the possible configurations of a given fleet.

The framework presented in this paper, makes use of the *k*-out-of-*n* system to describe the truck and shovel sub-systems. The two systems are then described as a series system. The theoretical work done on the nature of the reliability function of *k*-out-of-*n* systems is used to determine approximate estimates of the reliability while understanding the impacts of such approximations. The framework is illustrated with data from the Damang gold mining operation in Ghana. The data on 2 hydraulic shovels and 11 trucks over a 22 month period was analyzed using the proposed framework. The results indicate that alternator and electrical, structural, hydraulics, and bucket related faults constitute 74% and 79% of the total downtime of the two shovels. Preventive maintenance constitutes only 1% and 2% of the overall downtime of the shovels, far less than the recommended 80% of active maintenance activities (Kuruppu and Golosinski, 1998; Paraszczak, 2000). For the trucks, the data analysis indicates that preventive maintenance and engine related faults are the most critical constituting almost half (49%) of the cumulative downtime.

The individual trucks are more reliable than the shovels. At 50 hours of operation, the trucks have an average reliability of 48.29% compared to 22.51% for the shovels. The trend is confirmed by the higher average TBF (33.78 hours for trucks compared to 79.80 hours for shovels). This may be attributed to the minimal preventive maintenance on the shovels given the production pressures induced by the lack of excess loading capacity. Given, the reliability functions of the individual units, the reliability of the shovel system is estimated at 5.06% at 50 hours. The truck system reliability is estimated to be at most 2.53% at 50 hours. Using these two estimates, the overall truck-shovel system reliability is estimated at 0.13% at 50 hours of operation.

6. References

Ansell, J. I., and Phillips, M. J., 1989, "Practical problems in the statistical analysis of reliability data", *Applied Statistics*, Vol. 38, No. 2, pp. 205-247.

Barabady, J., 2005, "Reliability and maintainability analysis of crushing plants in Jajarm bauxite mine of Iran", *Proceedings of the Annual Reliability and Maintainability Symposium*, pp. 109-115.

Barabady, J., and Kumar, U., 2008, "Reliability analysis of mining equipment: a case study Of a crushing plant at Jajarm bauxite mine in Iran", *Reliability Engineering and System Safety*, Vol. 93, pp. 647–653.

Campbell, J. D., 1995, Uptime: Strategies for Excellence in Maintenance Management, Productivity Press, 204 pp.

Elevli, S., and Demirci, A., 1999, "The Effects of Maintenance Activities on Reliability and Profitability", *CIM Bulletin*, Vol. 92, No. 1011, pp. 49-51.

Ghodrati, B., and Kumar, U., 2005, "Reliability and Operating Environment-Based Spare Parts Estimation Approach: A Case Study in Kiruna Mine, Sweden", *Journal of Quality in Maintenance Engineering*, Vol. 11, No. 2, pp. 169-184.

Hall, R. A., Daneshmend, L. K., Lipsett, M. G., and Wong, J., 2000, "Reliability Analysis as a Tool for Surface Mining Equipment Evaluation and Selection", *CIM Bulletin*, Vol. 93, No. 1044, pp. 78-82.

Jardine, C., and Freudenreich, D., 1981, "Computerized maintenance system at the Carol project", *Mining Engineering*, October, Vol. 33, pp.1466-1470.

Kales, P., (1998), Reliability: For Technology, Engineering, and Management, Prentice Hall, 392 pp.

Kumar, U., and Huang, Y., 1993, "Reliability analysis of a mine production system - a case study", *Proceedings of the Annual Reliability and Maintainability Symposium*, pp. 167-172.

Kumar, U., and Klefsjö, B., 1992, "Reliability analysis of hydraulic systems of LHD machines using the power law process model", *Reliability Engineering and System Safety*, Vol. 35, No. 3, pp. 217-224.

Kumar, U., Klefsjö, B., and Granholm, S., 1989, "Reliability investigation for a fleet of load haul dump machines in a Swedish mine", *Reliability Engineering and System Safety*, Vol. 26, pp. 341-361.

Kuruppu, M. and Golosinski, T., 1998, "Maintenance practices of mining machinery", Proceedings of 7th International Symposium on Mine Planning and Equipment Selection, R. K. Singhal, ed., October 6-9, 1998, Calgary, Canada, pp. 607-612.

Mandal, S. K., and Banik, P. K., 1996, "Evaluation of reliability index of longwall equipment systems for production contingency", Mining Technology, Vol. 78, pp. 138-140.

Nuziale, T., and Vagenas, N., 2000, "A software architecture for reliability analysis of mining equipment", *International Journal of Surface Mining, Reclamation and Environment*, Vol. 114, pp. 19-34.

Paraszczak, J., 2000, "Failure and downtime reporting – a key to improve mine equipment performance", *CIM Bulletin*, Vol. 93, No. 1044, pp. 73-77.

Paraszczak, J., 2001, "Standard reliability and maintainability measures as means to improve equipment performance assessment", *SME Transactions*, Vol. 310, pp. 204-208.

Paraszczak, J., and Kallio, P. and Honkanen, J., 1998, "Setting reliability standards for underground mining equipment", *CIM Bulletin*, Vol. 91, No. 1024, pp. 73-77.

Rade, L., 1994, "A Consecutive *k*-Out-of-*n* Reliability System in Random Environment", Microelectronics Reliability, Vol. 34, No. 8, pp. 1311-1318.

Rao, K.R.M., and Prasad, P.V.N., 2001, "Graphical Methods for Reliability of Repairable Equipment and Maintenance Planning", *Proceedings of the Annual Reliability and Maintainability Symposium*, pp. 123-128

Rockwell Software, (2005), *Arena User's Guide*, Rockwell Automation, 150 pp.

Samanta, B. Sarkar, B., and Mukherjee, S. K., 2004 "Reliability modelling and performance analyses of an LHD system in mining", Journal of The South African Institute of Mining and Metallurgy, January, Vol. 104, No. 1, pp. 1-8.

Samanta, B., Sarkar,B. and Mukherjee, S. K., 2001, "Reliability analysis of shovel machines used in an open cast coal mine", *Mineral Resources Engineering*, Vol. 10, No. 2, pp. 219-231.

Vagenas, N., Runciman, N. and Clément, S. R., 1997, "A methodology for maintenance Analysis of mining equipment", *International Journal of Surface Mining, Reclamation and Environment,* pp. Vol. 11, No. 1, 33-40.

Yuan, S., and Grayson, R. L., 1995, "Analysis of the effectiveness of shovel-truck mining systems", SME Transactions, Vol. 296, pp. 1828-1833.

Asphalt mix design optimization for efficient plant management

Kwame Awuah-Offei[1] and Hooman Askari-Nasab
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

The role of aggregate gradation in hot mix asphalt performance is well documented in the literature. Yet, the Bailey method is the only tool available for guidance on aggregate gradation selection for optimal performance (Vavrik et al., 2000). Also, there is a lack of tools for design engineers and plant managers of quarry sites to manage stockpile inventory levels and control cost of aggregate used in asphalt mixes. This work presents a linear programming model of the asphalt mix design problem and a numerical algorithm to solve the model. The algorithm is implemented in MATLAB as an Asphalt Mix Design Optimization (AMIDO) program. The program is successfully verified with an example. The results show that using the Bailey method alone results in sub-optimal results and that cheaper mixes with similar aggregate ratios can be designed with the same aggregate stockpiles. For the specific stockpiles used in the verification, the AMIDO mix design resulted in 53 cents/ton reduction in aggregate cost. This work improves the state-of-the-art in asphalt mix design for dense graded mixes and could potentially be modified for other mixes.

1. Introduction

Hot mix asphalt (HMA) is a composite material consisting of aggregates, asphalt binder and air voids. The role of the aggregate gradation in the strength and performance of HMA has long been recognized by researchers (Richardson, 1912; Goode and Lufsey, 1962; Huber and Shuler, 1992; Roberts et al., 1996; Vavrik, 2000; and Vavrik et al., 2002). For aggregate from a durable source, the mix performance is dependent on the volumetric properties (air voids, voids in mineral aggregate (VMA), and voids filled with asphalt (VFA)). Since the adoption of the SUPERPAVE mix design criteria, an increasing number of engineers have been looking for ways to correlate the mix aggregate gradation to the volumetric properties. The Bailey method is the only approach that provides guidance on the relationship between the aggregate gradation and volumetric properties. The method also provides a means to evaluate the performance of existing mix designs and field changes to designed mixes. However, the method still involves considerable trial and error

[1] Corresponding author: Department of Mining & Nuclear Engineering, Missouri University of Science & Technology, 226 McNutt Hall, 1870 Miner Circle, Rolla, MO, 65401 USA. Phone: +1-573-341-6438. E-mail: kwamea@mst.edu

to achieve the desired material ratios for optimal mix performance. Also, the method does not take into consideration the cost of the aggregate in the mix. Hence, plant engineers and managers have no tools to manage the HMA production cost during HMA mix design. Finally, most engineers are not used to designing by volume as required by the Bailey method.

Hot mix asphalt (HMA) is a composite material consisting of aggregates, asphalt binder and air voids. The role of the aggregate gradation in the strength and performance of HMA has long been recognized by researchers (Richardson, 1912; Goode and Lufsey, 1962; Huber and Shuler, 1992; Roberts et al., 1996; Vavrik, 2000; and Vavrik et al., 2002). For aggregate from a durable source, the mix performance is dependent on the volumetric properties (air voids, voids in mineral aggregate (VMA), and voids filled with asphalt (VFA)). Since the adoption of the SUPERPAVE mix design criteria, an increasing number of engineers have been looking for ways to correlate the mix aggregate gradation to the volumetric properties. The Bailey method is the only approach that provides guidance on the relationship between the aggregate gradation and volumetric properties. The method also provides a means to evaluate the performance of existing mix designs and field changes to designed mixes. However, the method still involves considerable trial and error to achieve the desired material ratios for optimal mix performance. Also, the method does not take into consideration the cost of the aggregate in the mix. Hence, plant engineers and managers have no tools to manage the HMA production cost during HMA mix design. Finally, most engineers are not used to designing by volume as required by the Bailey method.

The objectives of the work presented in this paper are: (i) to develop a model to minimize the cost of the designed HMA mix while respecting all the constraints of the Bailey method principles; and (ii) to develop a numerical solution algorithm to solve such a model. A linear programming (LP) model of the mix design problem was first formulated to minimize the cost subject to the constraints imposed by the Bailey method. A numerical algorithm was then developed to solve the problem for any given data input. The algorithm was implemented in MATLAB R2007a (Mathworks, 2007). MATLAB was used because of the extensive library of numerical routines and the data management flexibility. Finally, numerical examples were solved to validate the models and solution algorithm. The work presented in this paper is limited to dense graded mix design based on the Bailey method. However, the principles could be applied to other mixtures. Also, just like the Bailey method, the approach presented here relies on the expertise of the engineer to evaluate material suitability for HMA production. The developed method significantly reduces the trial and error involved in the mix design process. In fact, if the engineer has a good estimate of the desired Bailey method ratios, trial and error is completely eliminated by the use of this method. Secondly, the method provides a tool for the plant engineer and/or manager to manage the cost of HMA production giving the plant a competitive advantage at bid time. If the quarry or site has both an aggregate plant and an HMA plant, it provides a means to control stockpile volumes to meet sales and marketing strategies. This could potentially reduce aggregate production cost since the mix design can now be tied into the needs and capabilities of the aggregate plant and quarry or pit. The method provides a means to reduce HMA cost by using the least expensive design which meets the design criteria. Finally, this method and developed computer program is likely to increase the number of engineers using the Bailey method because it does not require in-depth understanding of the computations. Especially, the fact that engineers do not have to change from mix design by weight to design by volume is likely to increase the comfort level of a significant number of engineers.

The paper is presented in eight sections. The next section reviews the existing literature on asphalt mix design using the Bailey method. The optimization model developed in this work is presented in Section 3. This is followed by the numerical modeling and solution algorithm. Section 5 presents the numerical examples and validation. Section 6 presents the results of the numerical examples used for verification. Section 7 presents the major conclusions and the final section is a list of the references.

2. Bailey method asphalt mix design

The role of aggregate gradation in the performance of HMA mixtures has long been recognized by researchers (Richardson, 1912; Goode and Lufsey, 1962; Huber and Shuler, 1992; and Roberts et al., 1996). The objective of asphalt mix design is to produce a mixture with *optimal* mixture properties (stability, durability, flexibility, fatigue resistance, skid resistance, permeability and workability). Several mix design methods have been proposed over the years to accomplish this task. Till the 1990s, the Hveem and Marshall methods of mix design were the most predominant methods in the United States with the Marshall method used by 38 states (Kandhal and Koehler, 1985). Both methods provide little to no guidance on aggregate structure selection to achieve *optimal* mixture properties. In the 1990s, the SUPERPAVE mix design method was introduced as a result of the Strategic Highway Research Program (SHRP) (Kennedy et al., 1994; McGennis et al., 1995). The SUPERPAVE method was to produce mixtures based on volumetric properties which could directly be correlated to field performance. The method recognizes the role of coarse aggregate angularity, fine aggregate angularity, flat and elongated particles, clay content, and gradation in the production of quality mixtures. Also, aggregate source properties, such as soundness, toughness, and deleterious materials, were also found to be important. However, the source property criteria were determined to reflect regional aggregate quality, and were usually based on aggregate availability. Hence these source properties were left to States to determine as appropriate (Brown et al., 2001). With respect to aggregate gradation, the SUPERPAVE method seeks to ensure appropriate: (i) maximum aggregate size for the application; (ii) VMA; and (iii) aggregate skeleton. These goals are achieved by controlling the nominal maximum sieve size, and the percent passing the #8 (2.36 mm) and #200 (0.075 mm) sieves (Vavrik, 2000). The two main new SUPERPAVE aggregate control criteria are the VMA and the restricted zone requirements. Since the introduction of the SUPERPAVE standards across the United States, several researchers have studied the limitations of the VMA (Hinrichsen and Heggen, 1996; Kandhal et al., 1998; Coree and Hislop, 1999) and restricted zone (Kandhal and Mallick, 2001; Kandhal et al., 2002) requirements and suggested alternative approaches.

The Bailey method of asphalt mix design pioneered by Mr. Robert Bailey (retired) of the Illinois Department of Transportation (IDOT) is a systematic approach to blending aggregates to achieve the desired mixture properties (Vavrik, 2000; Vavrik et al., 2002a and b). The method has been used by District 5 of IDOT since the early 1980s and in the

1990s IDOT promoted its use throughout the State. The Bailey Method can be used with any mix design method but the method itself is not a mix design method. The Bailey Method as discussed here is suitable for dense-graded mixtures but can be applied to stone matrix asphalt (SMA) and fine graded mixes with some modification (Vavrik et al., 2002a and b). The Bailey method rests on two basic principles - aggregate packing, and coarse and fine aggregate definition. The degree of aggregate packing in a mixture is a function of the type and amount of compactive effort, particle shape, particle surface texture, gradation, and particle strength and durability. The Bailey method proposes an alternate definition of coarse and fine aggregate in HMA mixtures based on the mixture packing and interlock characteristics. Coarse aggregates are defined as those particles that will create voids in a unit volume and fine aggregate are the particles that fill the created voids. A particle ratio of 0.22[2] is used in the Bailey method to break the mixture into different fractions via control sieves. Equation (1) shows the Bailey method definition of primary, secondary and tertiary control sieves. The half sieve is defined in the Bailey Method as shown in Equation (2). Using the standardized set of sieves in Table 1, and Equations (1) and (2), results in the control sieves shown in Table 2. Further to the control sieves, the method defines three aggregate ratios (Equation 3) to characterize the coarse, the coarse portion of the fine, and the fine portion of the fine aggregate in the mixture.

$$\begin{aligned} PCS &= 0.22NMPS \\ SCS &= 0.22PCS \\ TCS &= 0.22SCS \end{aligned} \qquad (1)$$

where NMPS is defined as one sieve larger than the first sieve that retains more than 10%[3]; and PCS, SCS and TCS are the primary, secondary and tertiary control sieves, respectively.

$$\text{Half sieve} = 0.5NMPS \qquad (2)$$

Table 1 US Standard Sieve Sizes for asphalt mix analysis

Sieve #	***Sieve Size***	
	US Standard	*(mm)*
1	1 1/2"	37.5
2	1"	25.0
3	3/4"	19.0
4	1/2"	12.5
5	3/8"	9.5
6	#4	4.75
7	#8	2.36
8	#16	1.18
9	#30	0.600
10	#50	0.300
11	#100	0.150
12	#200	0.075

[2] Vavrik (2000) and Vavrik et al. (2002b) provide exhaustive discussions on the justification of the choice of a particle ratio of 0.22.

[3] As defined by SUPERPAVE terminology.

Table 2 Asphalt mix control sieves

	NMPS (mm)					
	37.5	*25.0*	*19.0*	*12.5*	*9.5*	*4.75*
Half sieve	19.0	12.5	9.5	4.75	4.75	2.36
PCS	9.5	4.75	4.75	2.36	2.36	1.18
SCS	2.36	1.18	1.18	0.60	0.60	0.30
TCS	0.60	0.30	0.30	0.15	0.15	0.075

$$\text{CA ratio} = \frac{\text{\% passing half sieve - \% passing PCS}}{\text{100\% - \% passing half sieve}} \tag{3a}$$

$$\text{FA}_c = \frac{\text{\% passing SCS}}{\text{\% passing PCs}} \tag{3b}$$

$$\text{FA}_f = \frac{\text{\% passing TCS}}{\text{\% passing SCS}} \tag{3c}$$

The Bailey method involves blending the aggregate by volume using a design unit weight and then evaluating the resulting blend with the aggregate ratios (see Vavrik et al., 2002a for a step-by-step procedure). Several researchers have studied the relationships between the aggregate ratios and volumetric properties of the mixture (Mohammad and Shamsi, 2007; Vavrik, 2000; and Vavrik et al., 2002a and b). Table 3 gives recommended ranges for the aggregate ratios for initial mix designs.

Table 3 Recommended aggregate ratio ranges (Varvik et al., 2002a)

	NMPS (mm)					
	37.5	*25.0*	*19.0*	*12.5*	*9.5*	*4.75*
CA Ratio	0.80-0.95	0.70-0.85	0.60-0.75	0.50-0.65	0.40-0.55	0.30-0.45
FA_c Ratio	0.35-0.50	0.35-0.50	0.35-0.50	0.35-0.50	0.35-0.50	0.35-0.50
FA_f Ratio	0.35-0.50	0.35-0.50	0.35-0.50	0.35-0.50	0.35-0.50	0.35-0.50

The work presented in this paper, takes advantage of the evaluation asphalt mixtures using the Bailey method aggregate ratios to develop a mix design optimization algorithm. The procedures outlined here skips the determination of the design unit weight and the volumetric blending in the Bailey method and arrives at blends that meet the aggregate ratio targets by taking advantage of linear programming techniques. The concepts presented here hinge on the fact that, any mixture developed by the Bailey method will be evaluated with the aggregate ratios for acceptance. Hence, the path to an acceptable blend (one with acceptable aggregate ratios) can be improved by treating the problem as an optimization problem. This will reduce the iterative part of the process and provide a means to manage the HMA cost, and stockpile levels and consequently, provide a competitive edge for a contractor.

3. Mix design optimization

The mix design optimization problem can be formulated as an LP problem. The goal is to minimize the cost of the mix while meeting all the gradation limits set by the agency, the Bailey principles constraints and feasibility constraints (these include solutions that are non-negative and between 0 and 100). A typical asphalt plant will be fed through tunnels with draw points below each HMA stockpile or through feed bins. The number of bins or stockpiles feeding the plant depends on the plant manufacturer and model, and plant layout. Mineral filler or baghouse fines can be used in the mix design through screw conveyors feeding into the plant. Nonetheless, each plant has a finite number of feed stockpiles/bins. The mix design problem is to determine the percentage contribution of each stockpile/bin material to the total aggregate mix going into the asphalt plant. Consequently, the decision variables for this problem are the percentage contribution of each stockpile/bin material in the HMA mix.

3.1. Objective function

The objective for the mix design problem is to minimize the cost of the mix. Equation (4a) shows the objective function. However, to develop a generic algorithm, the algorithm has to take into consideration the case where some of the bins are not utilized for a particular design. In such cases, Equation (4b) becomes the objective function instead.

$$Min \quad z = cx \tag{4a}$$

$$Min \quad z = c_{eff}x \tag{4b}$$

where c is the 1 x n vector of costs, x is the n x 1 vector of bin percentages (decision variables), $c_{eff} = c.*c_u$ is the effective cost vector , c_u is the 1 x n vector of bins in use [0,1], and n is the number of bins/stockpiles.

3.2. Constraints

3.2.1 Percentage Constraint

This constraint is an equality constraint to ensure that the sum of the decision variables is equal to 100 (Equation 5).

$$A_{1eq}x = b_{1eq} \tag{5}$$

where $A_{1eq} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$, is a 1 x n vector of ones and $b_{1eq} = 100$.

3.2.2 Unused Bins Constraint

Equation (6) is a set of equality constraints to set the percentage of any unused bins to zero. This is necessary for a robust numerical algorithm capable of solving problems with different bin sizes. Hence, the program does not have to be developed with the same number bins/stockpiles as a particular example.

$$x_{(i)} = 0; \qquad \text{for } c_{eff(i)} = 0 \tag{6a}$$

$$A_{2eq}x = b_{2eq} \tag{6b}$$

where $x_{(i)}$ is element "i" of the vector x^4, A_{2eq} is a matrix with the same number of rows as unused bins/stockpiles and b_{2eq} is a vector of zeros.

3.2.3 Gradation Constraints

Equation (7) represents the gradation constraints. These constraints can be 0 to 100 or agency specifications on particular sieve sizes.

$$A_1 x \le b_1 \tag{7a}$$

$$A_1 = \begin{bmatrix} M \\ -M \end{bmatrix} \qquad b_1 = \begin{bmatrix} U \\ -L \end{bmatrix} \tag{7b}$$

where M is an m x n matrix of gradations, m is the number of sieve sizes included in the model, U is an m x 1 vector of the upper gradation limits, and L is an m x 1 vector of the lower gradation limits.

3.2.4 NMPS Constraint

Equation (8) represents the constraint to ensure the design NMPS is maintained by the solution. This is done by changing the upper bound of the next sieve below the NMPS sieve to 90%. Equation (8) uses a numbering convention where sieves are numbered from the largest sieve size (i.e. the largest sieve size is number 1).

$$U_{(nmps+1)} = 90 \tag{8}$$

3.2.5 CA Ratio Constraint

Two modeling approaches were taken to model the CA ratio constraint – modeling as a range or a specific value. If the CA ratio is modeled as a range, then the constraint is represented by the two inequalities shown in Equation (9). Equation (10) shows Equation (9) in matrix form. The half-sieve and PCS are determined from Table 2 and the given NMPS.

$$\sum_{j=1}^{n} m_{(a,j)}\left(1+CA_u\right) - m_{(b,j)} \le 100CA_u \tag{9a}$$

$$\sum_{j=1}^{n} m_{(b,j)} - m_{(a,j)}\left(1+CA_l\right) \ge 100CA_l \tag{9b}$$

[4] This convention of defining elements of a vector is used throughout this paper

where m(i,j) is the element of matrix M occupying row i and column j[5], a is the sieve number for the half-sieve, b is the sieve number for the PCS, CAl and CAu are the lower and upper bounds of the CA ratio, respectively.

$$A_2 x \le b_2 \tag{10a}$$

$$A_2 = \begin{bmatrix} A_2^1 \\ A_2^2 \end{bmatrix}$$

$$a_{2(i,j)}^1 = m_{(a,j)}\left(1 + CA_u\right) - m_{(b,j)} \tag{10b}$$

$$a_{2(i,j)}^2 = m_{(b,j)} - m_{(a,j)}\left(1 + CA_l\right)$$

$$b_2 = \begin{bmatrix} 100CA_u \\ -100CA_l \end{bmatrix} \tag{10c}$$

Alternatively, the CA ratio can be modeled as a specific value. This will represent the case where, for example, the engineer is trying to correct for field deviation and therefore has a specific CA ratio value for the mix. Equation (11) represents the CA ratio constraint in such a case.

$$A_2 x = b_2 \tag{11a}$$

$$\begin{aligned} a_{2(i,j)} &= m_{(a,j)}\left(1 + CA\right) - m_{(b,j)} \\ b_2 &= 100CA \end{aligned} \tag{11b}$$

3.2.6 FA$_c$ Constraint

The CA ratio modeling approach was used to model the FAc constraint as well. Equations (12) and (13) represent the inequality and equality constraint, respectively.

$$A_3 x \le b_3 \tag{12a}$$

$$A_3 = \begin{bmatrix} A_3^1 \\ A_3^2 \end{bmatrix}$$

$$a_{3(i,j)}^1 = m_{(c,j)} - FA_{cu} m_{(b,j)} \tag{12b}$$

$$a_{3(i,j)}^2 = FA_{cl} m_{(b,j)} - m_{(c,j)}$$

$$b_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{12c}$$

where FAcl and FAcu are the lower bound of the FAc ratio, respectively, c is the sieve number for the SCS, and the previously defined conventions for matrix elements apply.

[5] This convention for defining elements of a matrix is used throughout this paper.

$$A_3 x = b_3 \tag{13a}$$

$$\begin{aligned} a_{3(i,j)} &= m_{(c,j)} - FA_c m_{(b,j)} \\ b_3 &= 0 \end{aligned} \tag{13b}$$

3.2.7 FA$_f$ Constraint

The same modeling approach as before was adopted to model the FAf constraint too. Equations (14) and (15) represent the inequality and equality constraints, respectively.

$$A_4 x \le b_4 \tag{14a}$$

$$A_4 = \begin{bmatrix} A_4^1 \\ A_4^2 \end{bmatrix}$$

$$\begin{aligned} a_{4(i,j)}^1 &= m_{(d,j)} - FA_{fu} m_{(c,j)} \\ a_{4(i,j)}^2 &= FA_{fl} m_{(c,j)} - m_{(d,j)} \end{aligned} \tag{14b}$$

$$b_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{14c}$$

where FAfl and FAfu are the lower bound of the FAf ratio, respectively, d is the sieve number for the TCS and the previously defined conventions for matrix elements apply

$$A_4 x = b_4 \tag{15a}$$

$$\begin{aligned} a_{4(i,j)} &= m_{(d,j)} - FA_f m_{(c,j)} \\ b_4 &= 0 \end{aligned} \tag{15b}$$

3.2.8 Lower and Upper Bound Constraints

There are technological and regulatory reasons why an engineer will require lower and upper limits on the percentage from a particular bin/stockpile. For instance, many departments of transportation have a maximum percentage of recycle asphalt product (RAP) that can be used in a mix. Also, a mix design that requires 2% of a particular stockpile/bin may be difficult to achieve since it requires a very low conveyor belt speed. Equation (16) represents the constraint for the lower and upper bounds imposed on the solution. By ensuring that no element in l is less than 0, the non-negativity constraint of an LP problem is satisfied by the lower bound constraint. Hence the non-negativity constraint is not explicitly built into the LP model in this work.

$$\begin{aligned} & l \le x \le u \\ & l_{(i)} \ge 0 \\ & u_{(i)} \le 100 \end{aligned} \tag{16}$$

where l and u are the vectors with the lower and upper bounds on the solution, respectively.

3.3. Overall Problem

The overall optimization problem can be stated as Equation (17). The remaining challenge is the numerical model to assemble the matrices A and Aeq and the vectors b and beq. The next section discusses the developed numerical procedures and solution algorithms.

$$Min \quad z = c_{eff} x$$

Subject to:

$$\begin{aligned} &A_{eq} x = b_{eq} \\ &Ax \le b \\ &l \le x \le u \end{aligned} \tag{17}$$

4. Numerical Modeling

MATLAB 7.4 (Mathworks, 2007) was chosen as the programming platform for this work because of the large library of efficient numerical routines, flexible data storage and management, and efficient linear programming solution procedure amenable to programming generic problems. The “linprog” function in MATLAB (Mathworks, 2007) finds the minimum of the problem in Equation (18).

$$\begin{aligned} &\min_{x} f^T x \\ &st \\ &Ax \le b \\ &A_{eq} x = b_{eq} \\ &l \le x \le u \end{aligned} \tag{18}$$

where f, x, b, beq, l, and u are vectors and A and Aeq are matrices.

Linprog allows three different solution algorithms – large scale and medium scale optimization, and a variant of the Simplex method. The large scale optimization algorithm is based on the linear interior point solver (Zhang, 1995), a variant of Mehrotra's predictor-corrector algorithm (Mehrotra, 1992), a primal-dual interior-point method. The medium scale algorithm uses an active set method which is also a projection method (Gill et al., 1981). This method is a variant of the Simplex method (Dantzig et al., 1955). The simplex method is a variant of the popular simplex method introduced by Dantzig et al. (1955). It solves an auxiliary piecewise linear programming problem to find an initial feasible solution (Mathworks, 2007). A MATLAB (Mathworks, 2007) user can choose which algorithm linprog uses by changing the function options.

The model presented in Section 3 was developed into the Asphalt Mix Design Optimization (AMIDO) program in MATLAB program using the numerical algorithm shown in Figure 1.

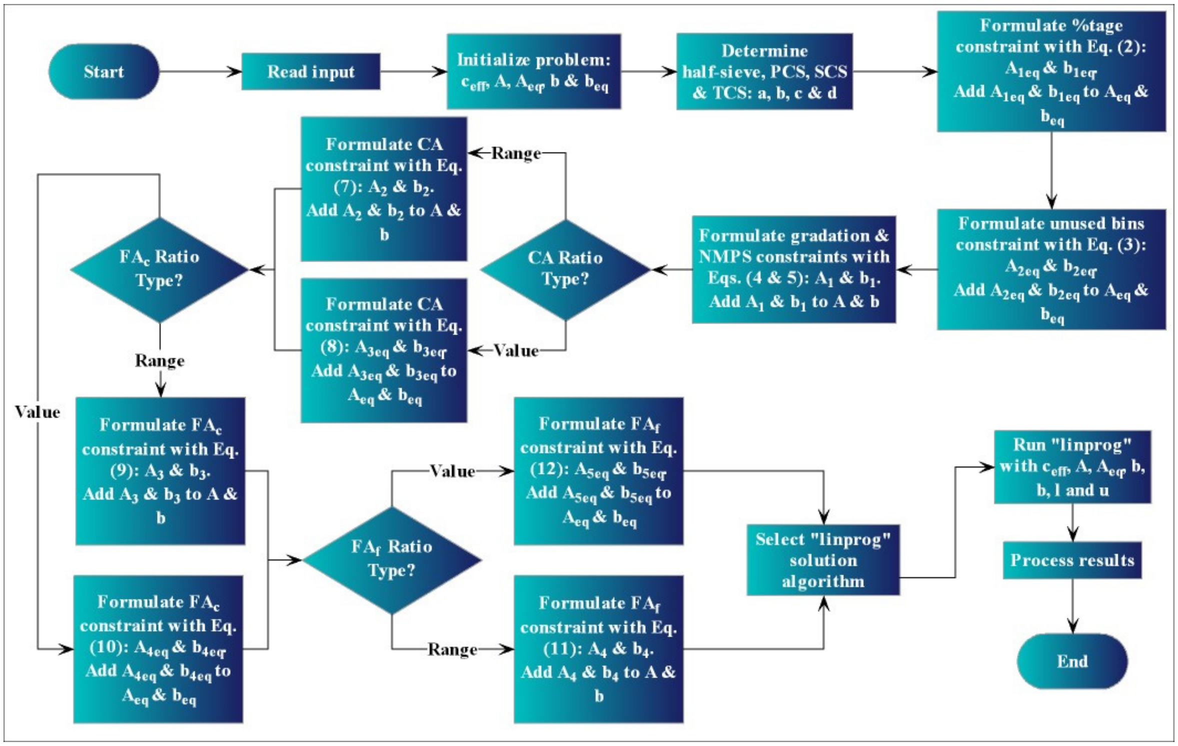

Start
Read input
Initialize problem: c_{eff}, A, A_{eq}, b & b_{eq}
Determine half-sieve, PCS, SCS & TCS: a, b, c & d
Formulate %tage constraint with Eq. (2): A_{1eq} & b_{1eq}, Add A_{1eq} & b_{1eq} to A_{eq} & b_{eq}
Formulate CA constraint with Eq. (7): A_2 & b_2. Add A_2 & b_2 to A & b
Range
FA$_c$ Ratio Type?
CA Ratio Type?
Formulate gradation & NMPS constraints with Eqs. (4 & 5): A_1 & b_1. Add A_1 & b_1 to A & b
Formulate unused bins constraint with Eq. (3): A_{2eq} & b_{2eq}, Add A_{2eq} & b_{2eq} to A_{eq} & b_{eq}
Formulate CA constraint with Eq. (8): A_{3eq} & b_{3eq}, Add A_{3eq} & b_{3eq} to A_{eq} & b_{eq}
Value
Range
Value
Formulate FA$_c$ constraint with Eq. (9): A_3 & b_3. Add A_3 & b_3 to A & b
Value
Formulate FA$_f$ constraint with Eq. (12): A_{5eq} & b_{5eq}, Add A_{5eq} & b_{5eq} to A_{eq} & b_{eq}
Run "linprog" with c_{eff}, A, A_{eq}, b, b, l and u
FA$_f$ Ratio Type?
Select "linprog" solution algorithm
Process results
Formulate FA$_c$ constraint with Eq. (10): A_{4eq} & b_{4eq}, Add A_{4eq} & b_{4eq} to A_{eq} & b_{eq}
Range
Formulate FA$_f$ constraint with Eq. (11): A_4 & b_4. Add A_4 & b_4 to A & b
End

Figure 1 AMIDO algorithm.

The program consists of 12 MATLAB functions in all – one master function and 11 sub-routines that are called from the master function to perform specific tasks. Taking advantage of MATLAB's interface with MS Excel, the program was developed to take MS Excel input files. This is because most engineers and technicians perform mix design in MS Excel and the data is easily available. Once the program is called, the first step is to read the input from the specified file. The sub-routine for reading the input file, expects 6 bins/stockpiles (including baghouse fines and RAP if used) and the gradations for list of sieve sizes in Table 1 (percent passing those sieves). This limitation can be easily removed by changing the code in this sub-routine alone. The input file provides information on gradations of each bin/stockpile, whether a bin/stockpile is to be used for the problem or not, user defined bin/stockpile descriptions, gradation specifications (limits), CA, FAc, and FAf ratios targets and whether they are to be ranges or a specific value, minimum and maximum percentages from each bin/stockpile, the cost of the material in each bin/stockpile and the NMPS sieve.

Once the data has been read into the program, the NMPS sieve specified is used in conjunction with Table 2 to determine the half sieve, the PCS, SCS and TCS. Then the algorithm formulates the percentage and unused bins constraints using the appropriate equations and adds the resulting matrix and vector to Aeq and beq, respectively (Figure 1). Subsequently, the gradation and NMPS constraints are formulated by the same subroutine and the resulting matrix and vector added to A and b, respectively. The next three constraints to be formulated depend on the user input. AMIDO is designed for the user to specify for each of the ratios whether or not there is a range or a specific value. From the input, the program decides which equations to use to formulate the constraints and whether the resulting matrix and vector are added to A and b or Aeq and beq. Once the whole problem has been formulated, the user is prompted to select one of the three (3) available solution algorithms. AMIDO then proceeds to run the "linprog" function with the provided input. Linprog returns a solution and an exitflag which is used to test if the program converged to a solution. If a solution was obtained, then the postprocessor function in AMIDO is invoked to generate a plot of the designed gradations. Also, the program reports the optimal value (the optimal value is the cost of a 100 units of the mix and does not include the production cost), the design bin/stockpile percentages, the CA, FAc, and FAf ratios for the blend, and the CPU run time for solving the LP problem.

5. Numerical Examples

The example used by Vavrik et al. (2002a) to illustrate the Bailey method is used to verify the code in AMIDO (see Table 4). Vavrik et al. (2002a) designed the mix with gradation control only on the 0.075-mm sieve (a target of 4.5% for the specification between 3.5-6.0%) and without regard to the cost of the various stockpiles. The designed mix was 13.2% coarse aggregate, 40.2% intermediate aggregate, 44% slag sand and 2.7% mineral filler (Vavrik et al., 2002a). The resulting CA, FAc and FAf ratios were 0.45, 0.41 and 0.44, respectively with a NMPS of 9.5 mm. First, the problem was solved with AMIDO using the Simplex algorithm option with $0.00/ton cost for all the aggregate to mimic the disregard for cost in the original solution. This problem is a multiple optima LP problem with several feasible solutions. However, it was used to test the ability of linprog to return valid solutions even for such problems. Subsequently, aggregate cost of $10.50, $10.50, $9.30 and $0.00 per ton were assumed for the coarse aggregate, intermediate aggregate, slag sand and mineral filler[6], respectively. The problem was then solved with the Missouri Department of Transportation (MO DOT) SP095 gradation specification (Results & Discussions

The trial results are shown in Table 6 and 7. Table 6 shows the stockpile design percentages, aggregate ratios and the aggregate cost per ton of mix while Table 7 shows the gradations of the mixes. Mix 1 is the Vavrik et al. (2002a) example with $0.00/ton cost for each stockpile. As stated earlier, this is a multiple optima problem and the program returns the first feasible solution. As shown in Table 6, the design differs from the Vavrik et al. (2002a) even though the two mixes have the same aggregate ratios. This problem shows the equality constraints in the AMIDO algorithm work well, since the aggregate ratios of the mix were the same as the specified ratios. A lab technician or an engineer could therefore easily reproduce the properties of a well performing mix by running AMIDO with the specific mix aggregate ratios. In this specific case, the VMA of the new mix may be different from the Vavrik et al. (2002a) mix because of the dissimilar materials in the stockpiles (slag sand and crushed rock). However, if the shape and texture of the

[6] Mineral filler was assumed to baghouse fines and thus the $0.00 cost.

stockpiles were similar one would expect the VMA and other volumetric properties of the mix to be the similar since the aggregate ratios are the same (Brown et al., 2001).

) using the Simplex algorithm again. The Bailey method aggregate ratios were set to the recommended ranges according to Table 3.

Table 4 Stockpile material gradations (after Vavrik et al., 2002a).

Sieve #	*Sieve Size*	*Coarse Aggregate*	*Intermediate Aggregate*	*Slag Sand*	*Mineral Filler*
1	1 1/2"	100.0	100.0	100.0	100.0
2	1"	100.0	100.0	100.0	100.0
3	3/4"	100.0	100.0	100.0	100.0
4	1/2"	94.0	100.0	100.0	100.0
5	3/8"	38.0	99.0	100.0	100.0
6	#4	3.0	30.0	99.0	100.0
7	#8	1.9	5.0	79.9	100.0
8	#16	1.8	2.5	48.8	100.0
9	#30	1.8	1.9	29.0	100.0
10	#50	1.8	1.4	14.2	100.0
11	#100	1.8	1.3	8.8	98.0
12	#200	1.7	1.2	3.0	90.0

Table 5 MO-DOT SP095 Gradation Specifications.

Sieve #	*Sieve Size*		*Specification*
	US Standard	*(mm)*	*% passing by weight*
1	1 1/2"	37.5	
2	1"	25.0	
3	3/4"	19.0	
4	1/2"	12.5	100
5	3/8"	9.5	90-100
6	#4	4.75	90 max
7	#8	2.36	32-67
8	#16	1.18	
9	#30	0.600	
10	#50	0.300	
11	#100	0.150	
12	#200	0.075	2-10

6. Results & Discussions

The trial results are shown in Table 6 and 7. Table 6 shows the stockpile design percentages, aggregate ratios and the aggregate cost per ton of mix while Table 7 shows the gradations of the mixes. Mix 1 is the Vavrik et al. (2002a) example with $0.00/ton cost for each stockpile. As stated earlier, this is a multiple optima problem and the program returns the first feasible solution. As shown in Table 6, the design differs from the Vavrik et al. (2002a) even though the two mixes have the same aggregate ratios. This problem shows the equality constraints in the AMIDO algorithm work well, since the aggregate ratios of the mix were the same as the specified ratios. A lab technician or an engineer could therefore easily reproduce the properties of a well performing mix by running AMIDO with the specific mix aggregate ratios. In this specific case, the VMA of the new mix may be different from the Vavrik et al. (2002a) mix because of the dissimilar materials in the stockpiles (slag sand and crushed rock). However, if the shape and texture of the stockpiles were similar one would expect the VMA and other volumetric properties of the mix to be the similar since the aggregate ratios are the same (Brown et al., 2001).

Mix 2 is the resulting mix from introducing aggregate cost into the problem. The solution respects all the constraints set forth in the problem, thus verifying the program. It is obvious from Table 6 that more of the cheaper stockpiles (slag sand and MF/baghouse fines) are used in the mix than when cost was not an issue. Consequently, the cost of Mix 2 is lower than Vavrik et al. (2002a) mix. Applying the same stockpile costs as in this problem, the Vavrik et al. (2002a) mix will use $9.70 of aggregate per ton of mix compared to $9.11 for Mix 2. This difference could be even more significant depending on the costs of the different stockpiles. On a big paving job, the 59 cents difference could be a substantial advantage. Since Mix 2 was using 6.5% of baghouse fines (Table 6), it was assumed the plant may only have 5% fines available. This assumption was then used to set-up another input file for AMIDO resulting in Mix 3. Again, an optimal solution was returned respecting all the previous constraints and the new constraint. The new solution however, does not have significantly different aggregate ratios from Mix 2. As would be expected, the new mix is slightly more expensive ($9.27/ton) but still 43 cents cheaper than the Vavrik et al. (2002a) mix.

Applying the algorithms proposed in this paper has obvious advantages in optimizing asphalt mix design by taking full advantage of the guidance to aggregate gradation selection provided by the Bailey method. This approach can be used to manage cost and stockpile inventories. Many asphalt producers are aggregate producers as well. The site has to be managed from a systems approach perspective covering the mining, and aggregate and asphalt plants. The properties of the reserve and the aggregate plant often dictate which stockpiles are scarce and which are abundant. Since aggregate markets are local, excess inventory of a particular stockpile can often not be easily disposed off. In such a situation, asphalt mix design needs to be optimized to help manage stockpile inventory while producing the same quality mixes. Often times, aggregate plant personnel complain about mix designs that do not take into consideration the aggregate plant inventories. Most lab technicians and QC/QA engineers try to take into consideration the aggregate plant stockpile inventories but without tools such as AMIDO, they rely solely on experience and inadvertently produce sub-optimal mixes. Also, the classical Bailey method process requires additional lab testing (unit weight for each coarse and fine aggregate - excluding MF, bag house fines, and RAP) which is completely avoided in the approach presented here. By concentrating on the aggregate ratios and not the design unit weight, the algorithm presented here avoids the testing and the blending by volume which is unfamiliar for most lab technicians and engineers. This is likely to increase the use of the Bailey method in mix designs. However, when the stockpiles have significantly different unit weights, blending by volume may be the best approach to design asphalt mixes for aggregate interlock using the Bailey method. Finally, it is not easy to know whether the target aggregate ratios and the specifications are in conflict. Hence an engineer could be trying to achieve mix properties (aggregate ratios) that are infeasible with the available stockpile gradations. Using the algorithms in AMIDO, the program returns an infeasible solution at the first trial saving hours of trial and error to arrive at the same conclusion.

Table 6 AMIDO designed mixes

	Mix 1	Mix 2	Mix 3
Coarse aggregate (%)	12.7	15.8	15.8
Intermediate aggregate (%)	41.3	19.2	20.1
Slag sand (%)	43.4	58.5	59.1
MF/Baghouse fines (%)	2.6	6.5	5.0
CA ratio	0.45	0.550	0.550
FA_c ratio	0.41	0.443	0.426
FA_f ratio	0.44	0.500	0.467
Cost/ton ($/ton)	0.00	9.11	9.27

Table 7 AMIDO results: mix gradations

Sieve #	*Sieve Size*		*% Passing*		
	US Standard	*(mm)*	*Mix 1*	*Mix 2*	*Mix 3*
1	1 1/2"	37.5	100.0	100.0	100.0
2	1"	25.0	100.0	100.0	100.0
3	3/4"	19.0	100.0	100.0	100.0
4	1/2"	12.5	99.2	99.0	99.1
5	3/8"	9.5	91.7	90.1	90.0
6	#4	4.75	58.3	70.7	70.0
7	#8	2.36	39.6	54.5	53.5
8	#16	1.18	25.1	35.8	34.6
9	#30	0.600	16.2	24.1	22.8
10	#50	0.300	9.6	15.4	14.0
11	#100	0.150	7.2	12.1	10.6
12	#200	0.075	4.4	8.1	6.8

7. Conclusions

The Bailey method provides useful guidance for aggregate gradation selection for interlock during asphalt mix design. The method however, does not take into account the aggregate cost and the stockpile inventory management goals of the plant. This work presents an LP model to minimize the cost of the designed HMA mix while respecting all the constraints of the Bailey method principles; and a numerical solution algorithm to solve such the model. The numerical algorithm is implemented in MATLAB (Mathworks, 2007) and takes full advantage of the linprog function. The developed algorithm is programmed into an Asphalt Mix Design Optimization (AMIDO) program. A numerical example has been used in this paper to verify the algorithm with both equality and inequality constraints for the Bailey method aggregate ratios. The results indicate that the mix designed with AMIDO is 59 cents/ton cheaper than the mix designed with just the Bailey method. This difference may provide competitive advantage to a contractor during a competitive bid. Furthermore, AMIDO provides a means to manage stockpile inventory for a systems management approach to quarry management. The proposed method will save time because of the reduced number of trials to find an acceptable starting blend for asphalt mix design. Also, the methodology avoids the extra lab testing and the unfamiliar blending by volume required by the Bailey method. This will improve the chance of engineers using the Bailey method in mix design. However, in situations where the stockpiles have significantly different unit weights, blending by volume may be the best way to design a mix. The work improves asphalt mix design methodology for dense graded mixes and could potentially be modified for other mixes.

8. References

Brown, R. Huber, G., Michael, L., Sines, R., Dukatz, E., Scherocman, J., D'Angelo, J. And Williams, C. (2001), *SUPERPAVE Mixture Design Guide*, WesTrack Forensic Team Consensus Report, Mitchell, Terry (ed), Federal Highway Administration, DC, USA 17 pp.

Coree, B. J. and Hislop, W. P. (1999), “Difficult Nature of Minimum Voids in Mineral Aggregate: Historical Perspective”, Transportation Research Record, No. 1681, Transportation Research Board, DC, USA pp. 148-156.

Dantzig, G.B., A. Orden, and P. Wolfe, (1955) "Generalized Simplex Method for Minimizing a Linear from Under Linear Inequality Constraints," *Pacific Journal of Math., Vol. 5(2)*, CA, USA, pp. 183–195.

Gill, P. E., Murray, W., and Wright, M.H., (1981), *Practical Optimization*, Academic Press, London, UK.

Goode, J. F. and Lufsey, L. A. (1962), “A New Graphical Chart for Evaluating Aggregate Gradations”, *Proceedings of the Association of Asphalt Paving Technologists*, *Vol. 31*, MN, USA pp. 176-207.

Hinrichsen, J. A. and Heggen, J. (1996), ”Minimum Voids in Mineral Aggregate in Hot-Mix Asphalt Based on Gradation and Volumetric Properties“, Transportation Research Record, No. 1545, Transportation Research Board, DC, USA pp. 75-79

Kandhal, P. S. and Keohler, W. S. (1985), „“Marshall Mix Design Method: Current Practices“, *Proceedings of The Association of Asphalt Paving Technologists, Vol. 54*, Association of Asphalt Paving Technologists, MN, USA pp. 284-303.

Kandhal, P. S. and Mallick, R. B. (2001), “Effect of Mix Gradation on Rutting Potential of Dense-Graded Asphalt Mixtures”, *Transportation Research Record, No. 1767*, Transportation Research Board, DC, USA pp. 146-151.

Kandhal, P. S., Dunning, M., Vavrik, W., Hugo, F., Wasil, B., Weishahn, L., Hobson, K., Nady, R. and Allen Cooley Jr., L. (2002), “Investigation of the Restricted Zone in the SUPERPAVE Aggregate Gradation Specification”, *Proceedings of the Association of Asphalt Paving Technologists, Vol. 71*, Association of Asphalt Paving Technologists, MN, USA pp. 479-534.

Kandhal, P. S., Foo, K. Y. and Mallick, R. B. (1998), “Critical Review of Voids in Mineral Aggregate Requirments in SUPERPAVE”, *Transportation Research Record, No. 1609*, Transportation Research Board, DC, USA pp. 21-27

Kennedy, T. W., Huber, G. A., Harrigan, E., Cominsky, R. J., Hughes, C. S., Von Quintus, H. and Moulthrop, J. S. (1994), *Superior Performing Asphalt Pavements (SUPERPAVE): The Product of the SHRP Asphalt Research Program*, Transportation Research Board, Report No. SHRP-A-410, DC, USA 156 pp.

Mathworks Inc, (2007), MATLAB 7.4 (R2007a), MA, USA.

McGennis, R. B., Anderson, R.M., Kennedy, T.W. and Solaimanian, M. (1995), *Background of SUPERPAVE Asphalt Mixture Design and Analysis*, Federal Highway Administration, Report No. FHWA-SA-95-003, DC, USA 172 pp.

Mehrotra, S., (1992)"On the Implementation of a Primal-Dual Interior Point Method," *SIAM Journal on Optimization, Vol. 2,* PA, USA pp. 575–601.

Mohammad, L. N. and Shamsi, K. A. (2007), “A Look at the Bailey Method and Locking Point Concept in SUPERPAVE Mixture Design”, Practical Approaches to Hot-Mix Asphalt Mix Design and Production Quality Control Testing, Transportation Research Circular, No. E-C124, Transportation Research Board, DC, USA pp. 24-32.

Richardson, C. (1912), *The Modern Asphalt Pavement*, 2nd Edition, John Wiley & Sons, NY, USA

Vavrik, W. R., Huber, G., Pine, W. J., Carpenter, S. H. and Bailey, R. (2002a), “Bailey Method for Gradation Selection in Hot-Mix Asphalt Mixture Design”, *Transportation Research Circular, No. E-C044*, Transportation Research Board, DC, USA 33 pp.

Vavrik, W. R., Pine, W. J. and Carpenter, S. H. (2002b), “Aggregate Blending for Asphalt Mix Design Bailey Method”, *Transportation Research Record, No. 1789*, Transportation Research Board, DC, USA pp. 146-153

Vavrik, W. R. (2000), Asphalt Mixture Design Concepts to Develop Aggregate Interlock, PhD Dissertation, University of Illinois at Urbana-Champaign, IL, USA 172 pp.

Zhang, Y., (1995)"Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment," *Technical Report TR96-01*, Department of Mathematics and Statistics, University of Maryland, Baltimore, MD, USA.

Life cycle assessment of belt conveyor and truck haulage systems in an open pit mine

Kwame Awuah-Offei[1], David Checkel[2], and Hooman Askari-Nasab[3]

Abstract

Life Cycle Assessment (LCA) is an emerging environmental management tool for examining environmental impact from cradle-to-grave. In this paper, LCA is used to evaluate the acidification potential (AP) and global warming potential (GWP) of belt conveyor and truck haulage systems in a surface gold mine using the ISO 14040 standards. The results show that for AP the truck system has 32 kg of equivalent SO_2 per functional unit compared to 8 kg for the conveyor system. However, for GWP, the conveyor has 2,820 kg of equivalent CO_2 per functional unit compared to 648 kg for the truck option.

1. Introduction

The choice of an appropriate materials handling system for a surface mine is a complex task with many variables. Such factors include the topography of the area, the lengths and inclinations of haulage routes, weather conditions, types of excavators and loading equipment, nature of the materials to be handled (size, density, wetness, stickiness, abrasiveness, corrosiveness and temperature) (Sweigard, 1992). In most operations, the task is reduced to a choice between belt conveyor and truck haulage systems as these have a wide range of flexibilities and advantages over other haulage systems. Some of the advantages of belt conveyors are that they: (i) are continuous materials handling systems with a high range of capacities; (ii) have low operating costs and safety of operation; (iii) have high versatility, reliability and are more environmentally friendly than other haulage systems. This view on environmental friendliness, however, is because only the environmental impact of the use of the belts conveyors as against that of the truck haulage system are often considered rather than assessing the environmental impacts of both systems over their entire life cycles in terms of capital and operating impacts as is done in life cycle assessments.

[1] Corresponding Author: Assistant Professor, Dept. of Mining & Nuclear Engineering, University of Missouri-Rolla, 226 McNutt Hall, Rolla, MO 65401. Phone: +1(573)341-6438; Fax: +1(573)341-6934; E-mail: kwamea@umr.edu

[2] Professor, , Mechanical Eng. Dept., University of Alberta, 4-9 Mechanical Engineering Building Edmonton, AB, Canada T6G 2G8. Phone: (780)492-2340; Fax: (780)492-2200, E-mail: dave.checkel@ualberta.ca

[3] Assistant Professor, Department of Civil & Environmental Engineering, University of Alberta, Edmonton, Alberta, Canada, T6G 2W2. Phone: (780) 492-4053; Fax: (780) 492-0249; E-mail: hooman@ualberta.ca

Life cycle assessment (LCA) is a technique for quantifying and interpreting environmental impacts associated with a product system over the life of the product from cradle to grave. This technique has been used to assess the environmental impacts of several products and systems including computer monitors (Socolof et al., 2005), agricultural production systems (Basset-Mens et al., 2007), solar thermal systems (Battisti and Corrado, 2005) and municipal waste systems (Chaya and Gheewala, 2007), just to name a few. Unfortunately, there is limited mining application of life cycle assessment to evaluate environmental burdens of a product or process. For instance a search in EI Compendex with the keywords life "cycle assessment" yields 1,650 results whereas the same search yields 4 results (35 results show up but only 4 are relevant) when combined with the keyword "mining". This lack of mining LCAs undermines the many life cycle inventories that have been developed since every product system consumes the products of mining. Some studies have been done applying LCA to mining (Suppen et al., 2006; Durucan et al., 2006; and Spitzley and Duane, 2004). Suppen et al. (2006) examines the application of LCA in mining on the basis of the recoginition of LCA as an important tool to promote sustainable development in Latin America and the Caribbean. The work proposes two approaches – a national life cycle inventory for base minerals and an integrated life cycle model for the management of mining processes. Durucan et al. (2006) provides comprehensive life cycle models of both surface and underground mining in an European context. Spitzley and Duane (2004) deal with the issue of land-use impacts in LCA methodology and deduced equivalency factors for gold, bauxite and copper mining.

The objective of this work was to develop an LCA method to evaluate belt conveyor and truck haulage systems based on their environmental impacts by conducting an LCA of the two systems for a hypothetical hard rock gold mine which is located in Alberta, Canada. The major environmental impacts of hauling operations on a mine are vibration, noise, greenhouse gas emissions, acid rain precursor emissions and particulate matter emissions. Since vibration and particulate matter emission effects tend to be on a local scale and mining operations are usually located in remote areas, the global warming impacts (GWP) and acidification impacts (AP) were used as measures of environmental impacts in this work. Consequently, the emission levels of carbon dioxide (CO_2), methane (CH_4), sulphur dioxide (SO_2) and oxides of nitrogen (NO_x) were monitored in this work. The LCA was conducted using the ISO 14040 standards (ISO TC 207, 1997).

2. Product system

It is assumed that a new orebody has been discovered 12 km (7.5 miles) away from an existing mine's processing facilities[4]. The waste stripped from the surface mine will be dumped about 4 km (2.5 miles) away from the new pit while the ore will be transported over a distance of 15 km to the old Run-of-Mine (ROM) Pad. Table 1 gives the vital parameters of the hypothetical pit. In such a case, the desire will be to choose an appropriate hauling system which takes into account both the technical, economic and environmental impacts of the operations. This paper evaluates the environmental impacts of the proposed truck and belt conveyor haulage systems for the new pit. The two systems

[4] This hypothetical problem is setup to mimic closely a real-life situation in Ghana, W. Africa. The only reason it is set-up to be hypothetical is to conduct the study for Alberta, Canada.

were designed to achieve the desired 4,000 t/hr. The truck haulage system comprises ten CAT 793 trucks with estimated availability and utilization of 85% each. These trucks have a carrying capacity of 218 tonnes (240 tons). The belt conveyor system is made up of Goodyear's Pylon Plus 1000/5 belts to convey both crushed ore and waste materials. The availability of the conveyor system is assumed to be 85%.The functional unit (the common measure of service provided) was defined as "hauling 4,000 t/hr of rock from pit to dump sites (Waste Dump and ROM Pad)".

Table 1 Important parameters of proposed pit

Parameter	***Value***
Productivity (tonnes/hr)	4,000
Average Grade of ore (g/tonne)	4
Average stripping ratio (dimensionless)	3:1
Life of mine (yr)	20
Haul distance to Waste Dump (km)	4
Haul distance to ROM Pad (km)	15
Scheduled number of working days per year	331
Number of shifts per day (dimensionless)	2
Scheduled shift duration (hr)	8

The product system for any product or system contains an infinite number of unit processes. In order to build the product system for each of the options, each branch was structured to mimic the available data. Figure 1 shows the original product system for the truck haulage system while Figure 2 shows that for the belt conveyor system. According to the ISO 14040 standards, scoping should be quantitative where possible (ISO TC 207., 1997). The Relative Mass-Energy-Economic Value (RMEE) method of scoping is the preferred method of identifying the significant unit processes because it is quantitative, repeatable and fair to both systems (Raynolds and Checkel, 2000). However, the RMEE method is deficient when applied to high mass and low energy products like rock. To be able to use RMEE in this work, energy was neglected energy as a cut-off criterion since the functional unit product has negligible energy content. Secondly, different cut-off ratios were used for mass and economic values to reflect the different scales on which the unit processes relate to the functional unit with respect to energy and economic value. Finally, a very low cut-off ratio for mass was used because the functional unit product (4,000 tonnes of rock) is far greater than any unit process product and renders scoping with mass otherwise irrelevant. The cut-off ratio for mass was thus set at 0.001% whereas that for economic value was set at 0.1%. Figure 3 and 4 show the product systems of both options after scoping.

3. Life cycle inventory analysis

Most of the data used is relevant to the province of Alberta and thus the uncertainties are low (≤ 25%) (Atkinson, 1992; Frizzel and Martin, 1992; Knights and Boerner, 2001; Anon., 2002c; Anon., 2003a; Anon., 2003b). The results of the inventory analysis are shown in Table 2 and 3.

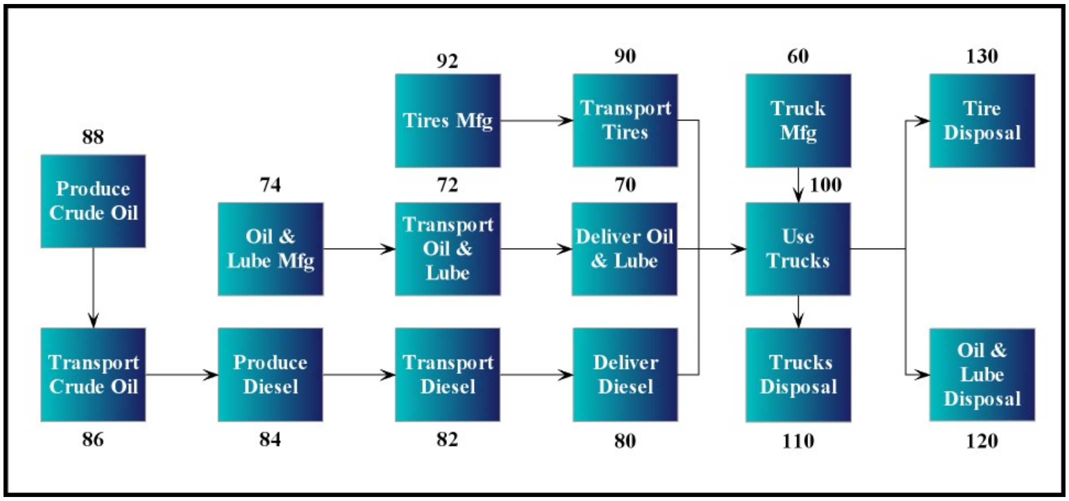


Figure 1 Original product system for truck haulage system.

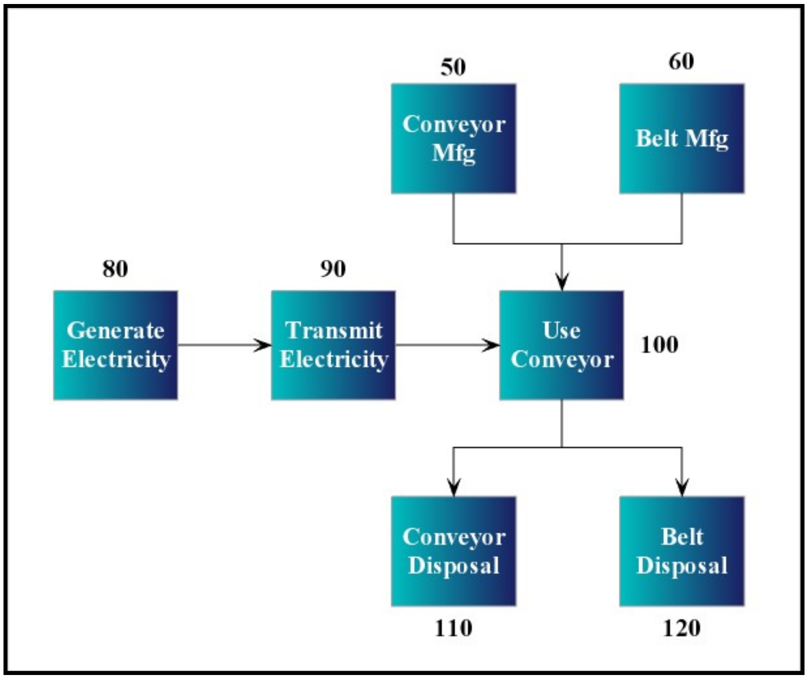


Figure 2 Original product system for belt conveyor system.

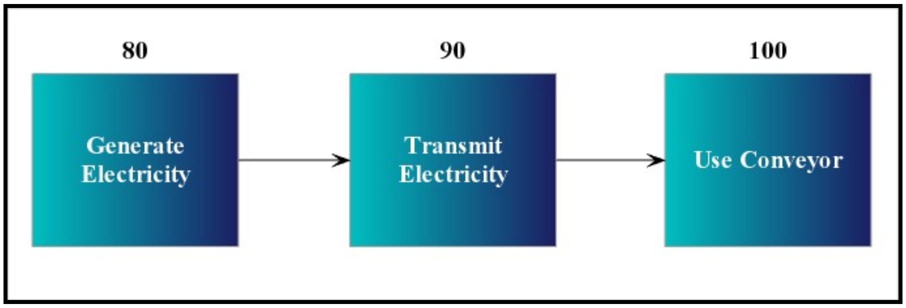


Figure 3 Product system for belt conveyor after modified-RMEE scoping.

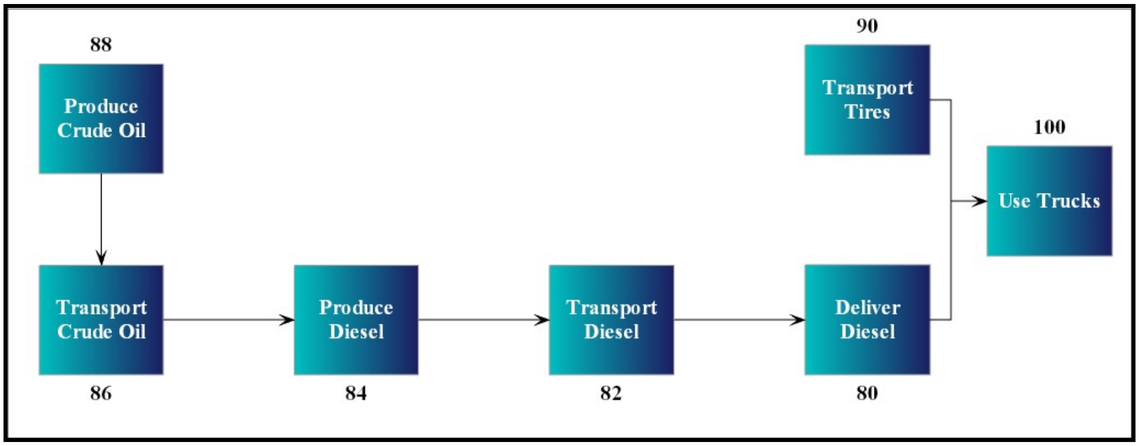


Figure 4 Product system for truck haulage system after modified-RMEE scoping.

In conducting the inventory analysis on the belt conveyor system, it was assumed that the average efficiency of the electric motors that drives the belt conveyors is 90% and that the transmission losses over the grid from the coal fired plants to the mine is 2%. The power requirement for the conveyor system is calculated as outlined by Sweigard (1992). In the inventory calculations involving the truck haulage option, it is assumed that the mechanical availability and utilization of the trucks average 85% each. It is also assumed that the average thermal efficiency of the CAT 3516B engine is 25%. Fuel consumption estimates were obtained both from Finning Canada Ltd. and the CAT Performance Handbook (Anon., 2002b).

Table 2 Inventory Matrix for the Truck Option.

Unit Process #	***100***	***90***	***80***	***82***	***84***	***86***	***88***
Unit Process Output	4000000 kg	80 kg	218.86 kg	220 l	261 l	696 l	696 l
CH_4 Emissions (kg)	0.00	0.0014	0.00	0.0001	0	0.0182	2.7562
CO_2 Emissions (kg)	428	6.50	0.00	2.42	19.68	3.14	124.84
N_2O Emissions (kg)	0.00	0.0002	0.00	0.00007	0.00	0.00	0.00
NO_x Emissions (kg)	42.8	0.055	0.00	0.011	0.019	0.085	0.38
SO_2 Emissions (kg)	0.00	0.000047	0.00	0.0057	0.065	0.0056	1.97

Table 3 Inventory Matrix for the Conveyor Option.

Unit Process #	***100***	***90***	***80***
Unit Process Output	4000000 kg	3756 kW-hr	3833 kW-hr
CH_4 Emissions (kg)	0.00	0.00	5.25
CO_2 Emissions (kg)	0.00	0.00	2667
N_2O Emissions (kg)	0.00	0.00	0.077
NO_x Emissions (kg)	0.00	0.00	6.59
SO_2 Emissions (kg)	0.00	0.00	3.23

4. Impact assessment and discussions

The two categories of impacts evaluated are Global Warming Potential (GWP) and Acidification Potential (AP). GWP equivalent factors over 100 years as reported by the

Intergovernmental Panel on Climate Change's Third Assessment Report (Anon., 2002c) were used in the analysis. For AP, NO_x is given an equivalence factor of 0.7 of SO_2. Monte Carlo simulation was used to evaluate the uncertainty in the impacts (Andrae et al., 2004; Loyd and Ries, 2007). The uncertainty analysis was carried out using @Risk3.5, a commercial risk analysis software (Palisade, 1997). Two thousand iterations were done during the simulation to estimate uncertainty parameters of both impacts. All input data was specified as samples from a normal distribution and the standard deviations were measures of both data variability as well as uncertainty due to data relevance. The uncertainty in the equivalence factors were not modeled in this work. The results are shown in Figure 5 and 6 which show the results for GWP and AP, respectively. Table 4 summarizes the results of the simulation showing the expected values, the standard deviations and covariance of the impacts for both options.

Figure 5 shows the expected values of the GWP in kg of equivalent CO2 and the 95% confidence ranges for the two systems. Clearly, the belt conveyor option with an expected value of 2,820 kg of CO2 per functional unit has greater global warming impacts compared to the truck option with 648 kg of CO2 per functional unit. At 95% confidence, the conveyor option has greater global warming impacts than the truck option. It is also observed that the conveyor option is has higher variability as shown by the higher covariance (Table 4). This is due to the higher variability in the power plant data (Anon., 2002c).

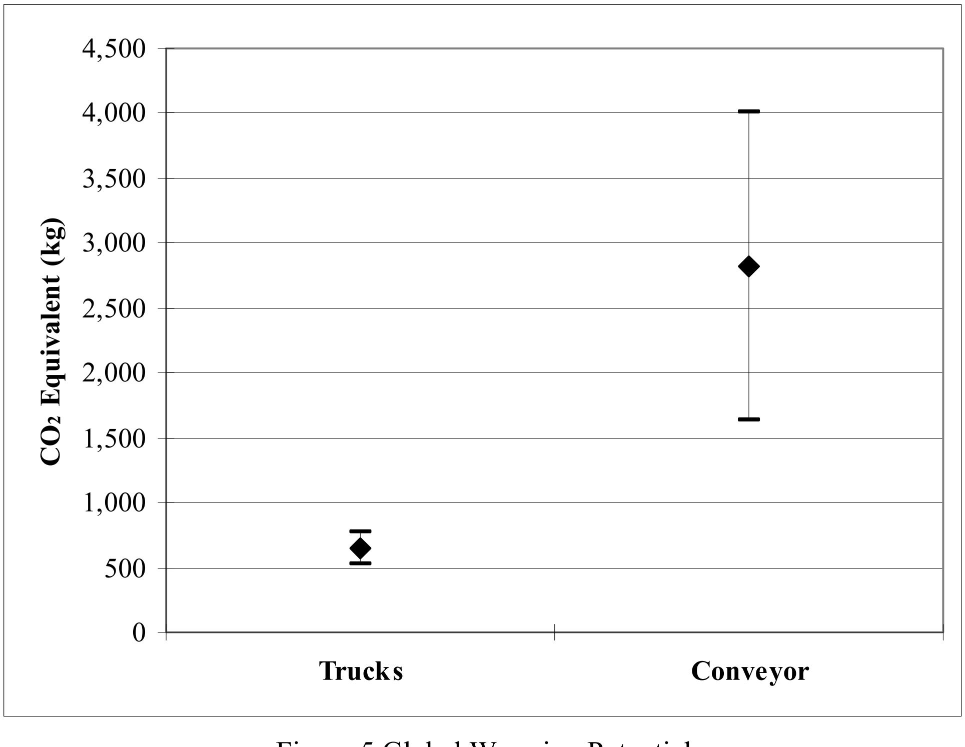


Figure 5 Global Warming Potential

Figure 6 shows the expected values of the AP in kg of equivalent SO_2 and the 95% confidence ranges of both options. In this case however, the truck option has higher acidification potential at 32 kg of SO_2 per functional unit compared to 8 kg of SO_2 per functional unit. At 95% confidence, the expected ranges of the two values do not overlap indicating higher impacts regardless of the uncertainty. Again, Table 4 shows the variability in the belt conveyor option (covariance of 17.1%) is higher than that of the truck option (covariance of 9.5%).

Table 4 Monte Carlo Estimates of Impact Uncertainties

Statistic	***Global Warming Potential (kg of CO_2)***		***Acid Rain Precursors (kg of SO_2)***	
	Trucks	Belt Conveyor	Trucks	Belt Conveyor
Expected Value (Mean)	648	2,820	32	8
Standard Deviation	62.61	604.92	3.03	1.37
Covariance	9.7%	21.5%	9.5%	17.1%

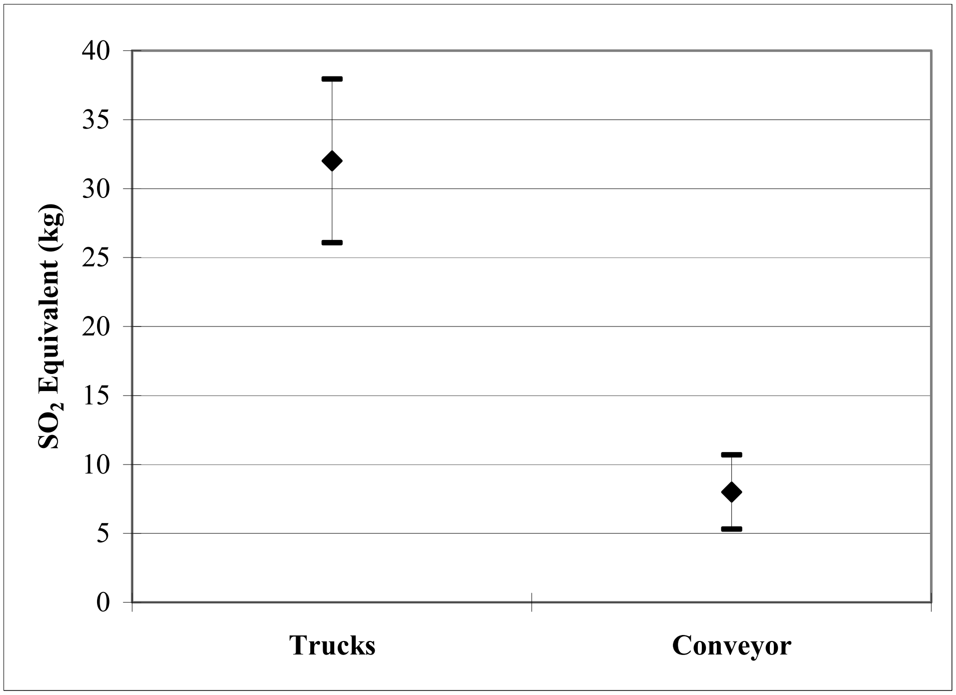


Figure 6 Acidification Potential

Sensitivity analysis in @Risk uses multivariable regression to rank input variables (in this case emission factors and product outputs) in order of which input variable affects the total output most. Table 5 shows the three most important input variables for each estimate of expected value in Table 4. It is interesting to note, that the regression coefficient between GWP for the belt conveyor system and CO2 emission factor for electricity generation (unit process 80) and that between AP for the trucks and the NOx emission factor for the functional unit were really high (0.883 and 0.907, respectively). Also, of interest is the fact that all 3 top variables for conveyor option, for both impacts, are related to the power generation unit process (UP#80). This is because this is the only unit process in the product system with emissions of interest.

Since the range of expected values at 95% confidence, for the two options do not overlap there is no pressing need for improvement of estimates. The only other area for improvement will be the individual options. The belt conveyor option is an obvious candidate for improvement because the power generation unit process (UP# 80) emits all the pollutants. Consequently, a natural gas fired power plant was considered instead of the original coal fired plant. Figure 7 and 8 show that the natural gas fired plant results in lower GWP and AP than the coal fired plant which is much more prevalent in Alberta, Canada. The GWP is reduced from 2,820 kg of equivalent CO2 per functional unit to 2,031 kg, reduction of 28%. The AP is also reduced from 8 kg of equivalent SO2 per functional unit to 5 kg, a reduction of 37.5%. Even though these improvements do not necessarily overturn the results of comparative analysis carried out in this work, it shows that using a natural gas fired plant (or say a hyrdo-electric plant) could significantly improve the life cycle impacts of a conveyor belt system.

Table 5 Summary of Results of @Risk Sensitivity Analysis

Impact	*Option*	*Input Variable*	*Regression Coefficient*
Global Warming Potential (kg of CO_2 Equivalents)	Trucks	CO_2 Emission Factor for UP 100	0.669
		CO_2 Emission Factor for UP 88	0.506
		Product Output for UP 88	0.453
	Conveyor	CO_2 Emission Factor for UP 80	0.883
		Product Output for UP 80	0.458
		CH_4 Emission Factor for UP 80	0.051
Acid Rain Precursors (kg of SO_2 Equivalents)	Trucks	NO_x Emission Factor for UP 100	0.907
		SO_2 Emission Factor for UP 84	0.389
		SO_2 Emission Factor for UP 88	0.154
	Conveyor	NO_x Emission Factor for UP 80	0.678
		Product Output for UP 80	0.562
		SO_2 Emission Factor for UP 80	0.457

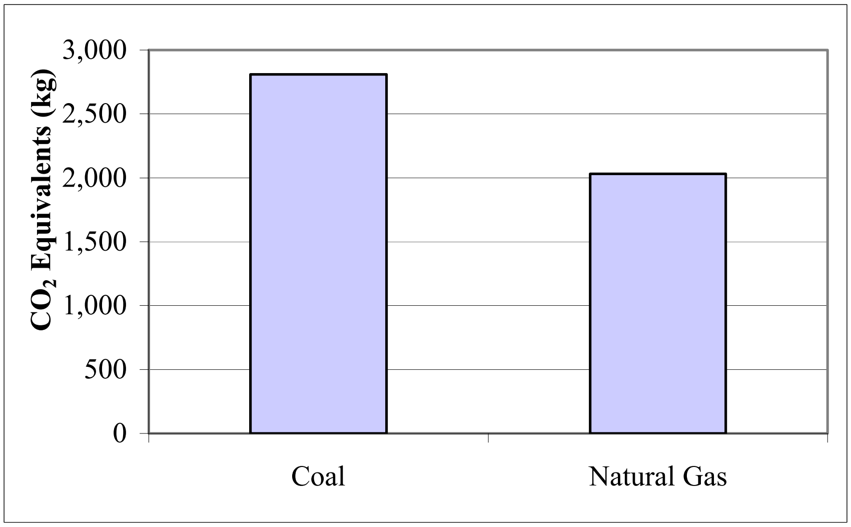


Figure 7 Global Warming Potential for Coal and Natural Gas Plants

5. Conclusions

The environmental impacts of mining have in recent times received increasing attention from society. The mining industry has responded by adopting available tools (including structured environmental management systems like ISO 14001 certified systems) to assess and manage environmental impacts better. Life Cycle Assessment is a new methodology for comprehensive evaluation of the environmental impacts of a system or product. This technique though has not been applied extensively mining. This work presents an example of engineering evaluation using LCA. A hypothetical surface mine has been successfully used to illustrate this approach in evaluating belt conveyor and truck haulage systems. The results show that for acidification potential the truck haulage system has 32 kg of equivalent SO2 per functional unit while the conveyor system has 8 kg. However, for global warming potential, the belt conveyor option has 2,820 kg of equivalent CO2 per functional unit while the truck option has 648 kg. The result of Monte Carlo simulation shows that the uncertainty in the impacts does not result in overlapping boundaries at the 95% confidence level. Consequently, at that level of confidence, the truck option has lower GWP and the conveyor option has lower AP. Using natural gas instead of coal to generate electricity at power generating stations improves the impact of the belt conveyor system on the environment. The long held impression that belt conveyor systems are environmentally friendlier than truck haulage systems is not entirely true. To categorically answer that question, more impact categories will have to be added or an evaluation of the relative importance of the two impacts will have to be carried out.

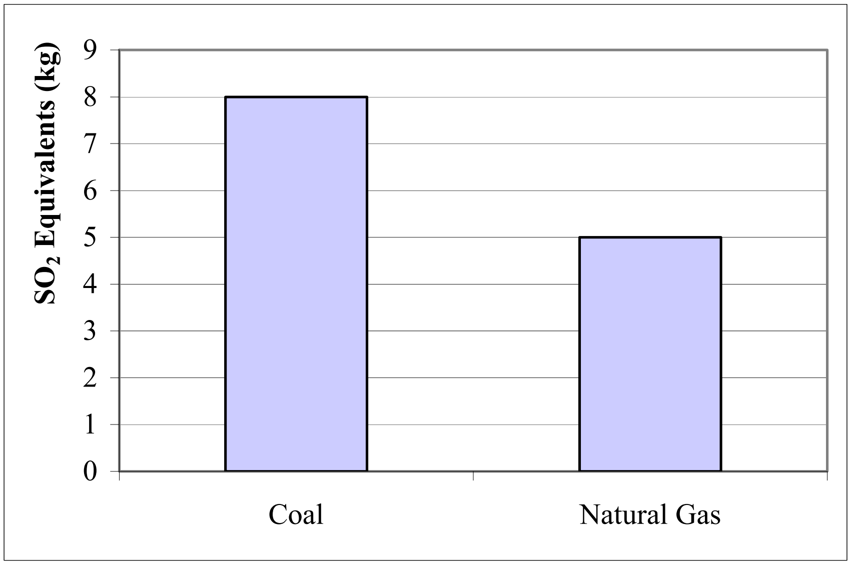


Figure 8 Acid Rain Precursor Impact for Coal and Natural Gas Plants

6. References

Andrae, A. S. G., Moller, P., Anderson, J. and Liu, J., 2004, "Uncertainty Estimation by Monte Carlo Simulation Applied to Life Cycle Inventory of Cordless Phones and Microscale Metallization Processes", IEEE Transactions of Electronics Packaging Manufacturing, Vol. 27, pp. 233-245.

Anon., 2002a, "Greenhouse Gases and Global Warming Potential Values", *USEPA*, 11p.

Anon., 2002b, "Caterpillar Performance Handbook", *33rd ed., Caterpillar Inc.*, pp. 20-1-20-36.

Anon., 2002c, Pembina Institute for Appropriate Development, Alberta, Canada webpage, www.lcva.ca.

Anon., 2003a, Caterpillar Inc. webpage, www.cat.com

Anon., 2003b, Goodyear Tire and Rubber Company, USA, webpage, www.goodyear.com

Atkinson, T., 1992, "Design and Layout of Haul Roads", SME Mining Engineering Handbook, 2nd ed., Vol. 2., (H. L. Hartman, sen. ed.), pp. 1334-1342.

Basset-Mens, C., Van der Werf, H.M.G, Robin, P., Morvan, T. H., Hassouna, M., Paillat, J.-M, and Vertès, F., 2007, 'Methods and Data for the Environmental Inventory of Contrasting Pig Production Systems", Journal of Cleaner Production, Vol. 15, pp. 1395-1405.

Battisti, R. and Corrado, A., 2005, "Environmental Assessment of Solar Thermal Collectors With Integrated Water Storage" Journal of Cleaner Production, Vol. 13, pp. 1295-1300.

Chaya, W. and Gheewala, S. H., 2007, "Life Cycle Assessment of MSW-To-Energy Schemes in Thailand", Journal of Cleaner Production, Vol. 15, pp. 1463-1468.

Frizzell, E. M. and Martin, T. W., 1992, "In-pit Crushing and Conveying", SME Mining Engineering Handbook, 2nd Ed., Vol. 2, (H. L. Hartman, se. ed.), pp. 1343-1351.

ISO TC 207, 1997, ISO 14040: Environmental Management - Life Cycle Assessment: Principles And Framework.

Knights, P. F. and Boerner, A. L., 2001, "Statistical Correlation of Off-Highway Tire Failures With Openpit Haulage Routes", Mining Engineering, Vol. 53, pp. 51-56.

Lloyd, S. M. and Ries. R, (2007), "Characterizing, Propagating, and Analyzing Uncertainty in Life-Cycle Assessment: A Survey of Quantitative Approaches", Journal of Industrial Ecology, Vol. 11, pp. 161-180.

Palisade, 1997, @RISK: Advanced Risk Analysis for Spreadsheets. © Palisade Corporation, Newfield, NY, USA

Raynolds, M., Fraser, R. and Checkel, D., 2000, "The Relative Mass-Energy-Economic Value (RMEE) Method for System Boundary Selection – Part I: A Means to Systematically and Quantitatively Select LCA Boundaries", Int. Journal of Life Cycle Assessment, pp. 96-104.

Socolof, M. L., Jonathan G. Overly, J. G., and Geibig, J. R., 2005, "Environmental Life-Cycle Impacts of CRT and LCD Desktop Computer Displays", Journal of Cleaner Production, Vol. 13, pp. 1281-1294.

Sweigard, R. J., 1992, "Materials Handling: Loading and Haulage", SME Mining Engineering Handbook, 2nd ed., Vol. 2, (H. L. Hartman, sen. ed.), pp 761-782.

Guidelines for using TOMLAB on Linux clusters

Hooman Askari-Nasab
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

This is a document on how to run and compile TOMLAB/CPLEX projects on the University Linux clusters. The servers are referred to as the AICT statistical and numerical servers.

1. Compiling TOMLAB/CPLEX files on the numerical server:

Starting MATLAB:

1- Login to the numerical server num.srv.ualberta.ca through putty.

2- Set the shared folder of tomlab on the linux path as follows:

→ setenv LD_LIBRARY_PATH /afs/ualberta.ca/home/h/o/hooman/tomlab/shared
→ setenv LD_LIBRARY_PATH /afs/ualberta.ca/home/h/o/hooman/tomlab/shared:$LD_LIBRARY_PATH

3- Start up MATLAB
→ /usr/local/bin/matlab

4- Change directory to the tomlab folder within matlab:
afs/ualberta.ca/home/h/o/hooman/tomlab

5- Type startup at the matlab prompt. Make sure there is no error running tomlab.

6- Locate the folder containing the final tomlab code.

7- Set the profiler on in the tomlab function running cplex.

```
profile on
..
profile report
```

8- Run TOMLAB/CPLEX code and make a minimal copy of the TOMLAB/CPLEX files required reported by the profiler. Data files *.mat files should be copied as well (see TOMLAB SAL documentation). Don’t forget to copy the tomlab.lic file.

9- This is a list of the minimal files that we got in our run. This list needs to be updated anytime that you want to run the code. Table 1 should be used just as a guideline only.

Table 1 – Files required by TOMLAB/CPLEX to compile the MILP on the Linux cluster

tomRun	cplexTL	cplex	cplexmex	tomlab
tomlablic	PrintResult	optParamSet	mipAssign	checkType
defblbu	optParamDef	DefPar	cplexStatus	xprint
StateDef	ProbDef	LineParamDef	checkAssign	tomFiles
mkbound	SOLGet	iniSolveMini	ResultDef	getfield
cpx2retvec	xprinte	xprinti	endSolveMini	etime
blanks				

10- Copy the folder containing all the required files to the server. example:
→ /afs/ualberta.ca/home/h/o/hooman/mipTest

11- Logout of matlab and logback in to make sure tomlab startup files are not on the path any more.

12- Test the code to see if it runs in MATLAB on the server without starting up TOMLAB. It is possible that some files are missing. One needs to find them with trial and error and replace them.
→ Note: don't copy the log10.m file from TOMLAB directory this is a MATLAB function.

→ Note: To find a file for instance etime.m do the following:
local_shared/matlab2008b> find . -name etime.m

13- Before compiling the files make a copy of the folder mipTest. Other files are going to be generated within this folder.

14- Change directory to the folder:
→ /afs/ualberta.ca/home/h/o/hooman/mipTest

→ Copy all the content of the tomlab/shared folder into
/afs/ualberta.ca/home/h/o/hooman/mipTest

15- Open the matlab file and comment out the profile on and profile report that you added to the file before. Matlab compiler can't compile the profiler.

16- Create a folder mipTest on /scratch_shared/hooman/mipTest
mkdir /scratch_shared/hooman/mipTest

17- Copy the Tomlab compiled folder to /scratch_shared/hooman/mipTest

→ cp –r mipTest/* /scratch_shared/hooman/mipTest

18- One must create an "mcr cache" directory under /scratch_shared/hooman on the server. This only has to be done once. For example, execute

```
→ mkdir -p /scratch_shared/hooman/my_matlab_mcr_cache
```

Then, in your PBS script, add the following line just before the cd command line.

```
→ export MCR_CACHE_ROOT=/scratch_shared/hooman/my_matlab_mcr_cache
```

This line will have to appear in all your PBS scripts that run Matlab executables.

interactively to check the compiled file before submitting as a batch this could be done on the linux command as:

```
→ setenv MCR_CACHE_ROOT /scratch_shared/hooman/my_matlab_mcr_cache
```

19- Compile the matlab files from the linux command prompt:

```
→ scratch_shared/hooman/mipTest> mcc -m mip.m
```

the following Warnings are generated. Couldn't figure out what they are. But it seems everything is working.

```
 Warning: Duplicate directory name:
/data1/local_shared_nfs/matlab2008b/toolbox/local.
Warning: Name is nonexistent or not a directory:
/data1/local_shared_nfs/matlab2008b/toolbox/compiler/patch.
Warning: Name is nonexistent or not a directory:
/data1/local_shared_nfs/matlab2008b/toolbox/compiler/patch.
Warning: Name is nonexistent or not a directory:
/data1/local_shared_nfs/matlab2008b/toolbox/compiler/patch.
```

20- Test running the compiled file interactively in the linux prompt environment.

```
→ scratch_shared/hooman/mipTest> ./run_mip.sh /local_shared/matlab2008b
```

21- It is possible that still some files are missing in tomlab. Search for the files reported and add them to the directory. Example:

```
------------------------------------------
Setting up environment variables
---
LD_LIBRARY_PATH is
.:/local_shared/matlab2008b/runtime/glnxa64:/local_shared/matlab2008b/bin/glnxa64:/
local_shared/matlab2008b/sys/os/glnxa64:/local_shared/matlab2008b/sys/java/jre/glnx
a64/jre/lib/amd64/native_threads:/local_shared/matlab2008b/sys/java/jre/glnxa64/jre/li
b/amd64/server:/local_shared/matlab2008b/sys/java/jre/glnxa64/jre/lib/amd64/client:/l
ocal_shared/matlab2008b/sys/java/jre/glnxa64/jre/lib/amd64
m =  77736
n =  19232
Mine Scheduling Problem. Variables 19232. Constraints 77736
??? Undefined function or variable 'tomlabVersion'.
MATLAB:UndefinedFunction
```

22- Search for the file missing and add it to the directory. Recompile and run again till there is no problem.

23- Another error it looks for ProbCheck.m function. Find it in Tomlab and add it to the directory. Recompile the folder.

24- If there is no error of missing files. The code will look for the tomlab.lic file in the cache.

```
Mine Scheduling Problem. Variables 19232. Constraints 77736
Error 2100: Cannot open file
/scratch_shared/hooman/my_matlab_mcr_cache/.mcrCache7.9/mip_B40/mip/toml
ab.lic for reading.
Set environment variable TOMLAB_LICENSE_FILE to path of your license file
or name it tomlab.lic and put it in your tomlab installation directory.
??? Error using ==> tomlablic
License error.
```

Locate and copy the tomlab.lic file to the folder

→ hooman/my_matlab_mcr_cache> cp tomlab.lic /scratch_shared/hooman/my_matlab_mcr_cache/.mcrCache7.9/mip_B40/mip/

The .mcrCache7.9 folder is hidden and the mip_B40 folder is a folder that is created with every compilation run with a different name. So this must be checked

25- Run the code interactively again and make sure it works.

→ scratch_shared/hooman/mipTest> ./run_mip.sh /local_shared/matlab2008b

26- Write a .pbs script file on the server as follows (use gedit command to create the file):

```
c #!/bin/bash
#PBS -S /bin/bash
#PBS -q mainq
#PBS -N jobx
#PBS -l nodes=1:ppn=1
#PBS -m be
#PBS -M hooman@ualberta.ca
export MCR_CACHE_ROOT=/scratch_shared/hooman/my_matlab_mcr_cache
cd /scratch_shared/hooman/mipTest
./run_mip.sh /local_shared/matlab2008b
```

→ Note: mipTest is the folder that you have the tomlab compiled files in. Check to see what is the .sh file name here it was run_mip.sh

→ Note: if you create the file on a windows system you need to convert it to Unix file format with:

```
dos2unix -o mipTest.pbs
```

27- Finally submit the job:

```
/scratch_shared/hooman> qsub mipTestPBS
3663.sage.nic.ualberta.ca
```

28- Check the status of the job:

```
/scratch_shared/hooman> qstat
```

2. References

[1] Holmström, K., (1989-2009), "TOMLAB /CPLEX - v11.2", ver. Pullman, WA, USA: Tomlab Optimization.