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

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:

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

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

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

Subject to:

2.2.2 Grade blending constraints

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

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

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).

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

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.

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

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).

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

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

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 (***N*, *a***) by the set of vertices, ***N*** (blocks); connected by ordered pairs of elements called arcs, ***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.

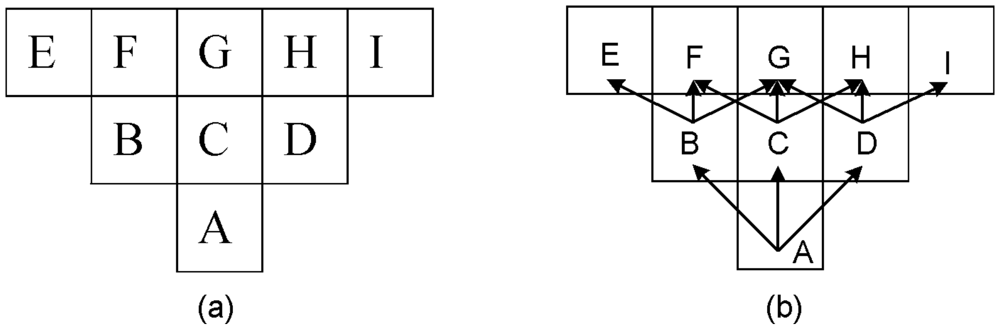


Figure 1 – Directed graph of extraction precedence

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.

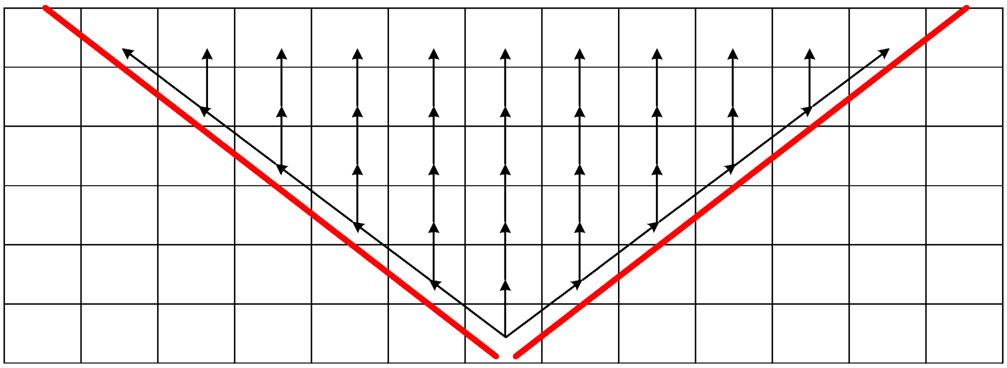


Figure 3 – Inverted cone and the directed graph for 45 degree slope.

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.

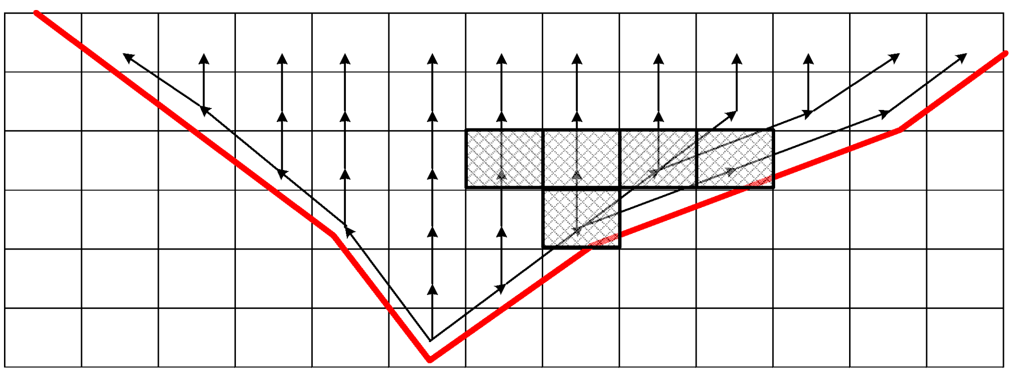


Figure 4 – Inverted cone and the directed graph defining variable slopes.

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

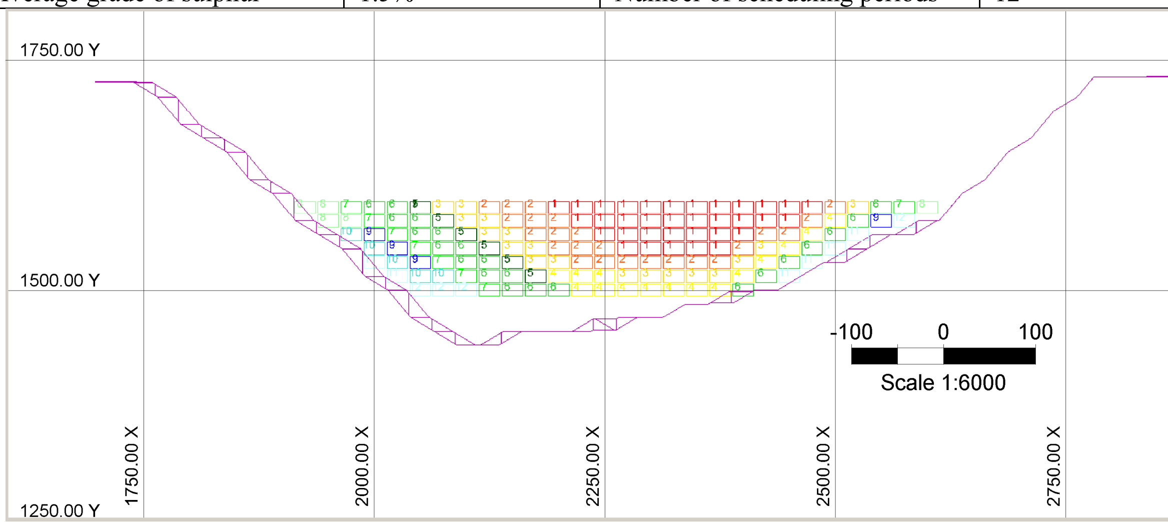


Figure 5 – Cross section 98400 showing the push-back and the schedule, looking east (meters).

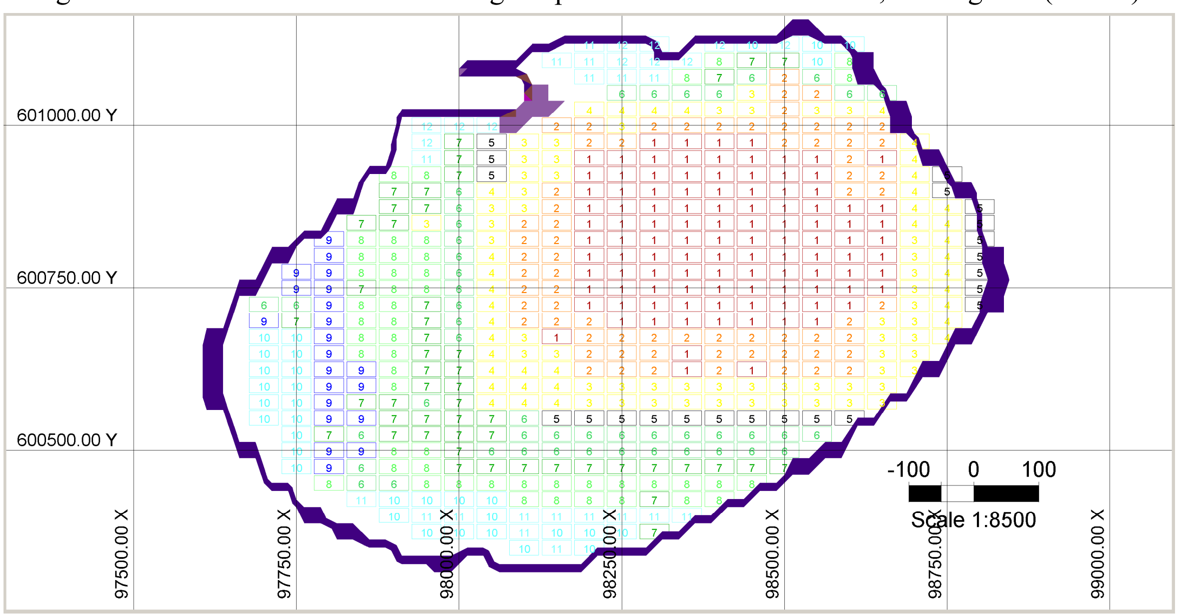
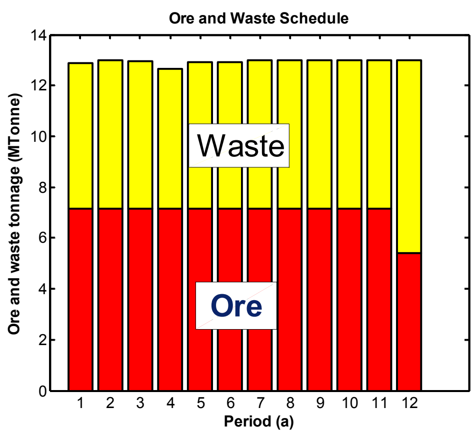


Figure 6- Plan view of schedule on bench 1570 (metes).



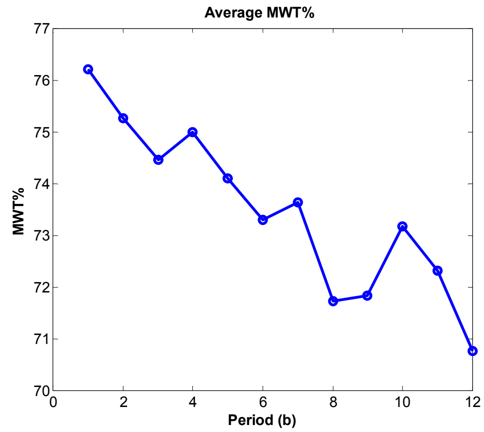
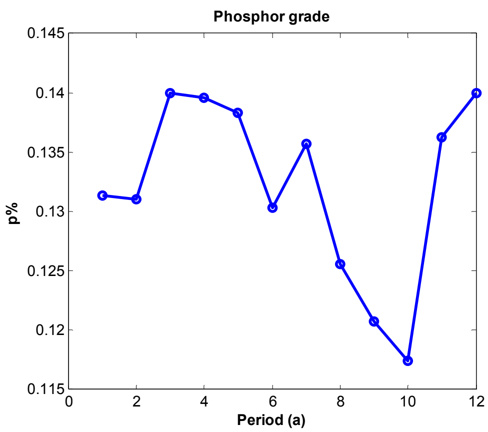


Figure 7 – (a) scheduled ore and waste over twelve periods; (b) average grade of ore in each period.



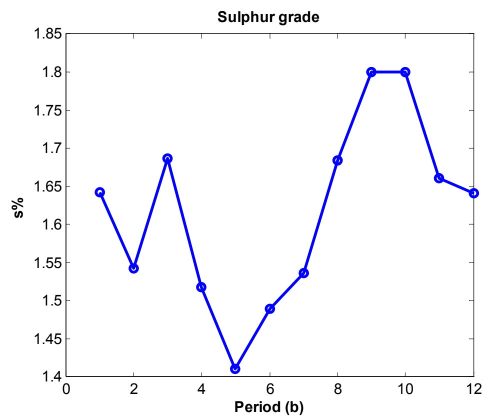


Figure 8 – (a) average grade of phosphor in each period; (b) average grade of sulfur in each period.

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.