

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.

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:

$$\text{maximize } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (1)$$

subject to the following constraints:

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

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \quad (3)$$

$$\cdot \quad \quad \quad \cdot$$

$$\cdot \quad \quad \quad \cdot$$

$$\cdot \quad \quad \quad \cdot$$

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

$$\text{and all } x_i \geq 0 \quad (5)$$

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

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

$$\text{subject to:} \quad Ax \leq b \quad (7)$$

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

Here x is an n -dimensional column vector, c^T is an n -dimensional row vector, A is an $m \times n$ 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} , $i=1, 2, \dots, m$; $j=1, 2, \dots, n$ to:

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

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

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

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

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):

$$\text{maximize} \quad z = \sum_{k=1}^p a_k x_k \quad (13)$$

$$\text{subject to:} \quad -x_j + x_k \leq 0 \quad \text{for all } k \text{ and } j \in D_k \quad (14)$$

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

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):

$$\text{maximize} \quad aX \quad (16)$$

$$\text{subject to:} \quad AX \leq 0 \quad (17)$$

$$IX \leq 1 \quad (18)$$

$$X \geq 0 \quad (19)$$

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

A = $q \times p$ matrix of the block precedence relationships,

I = $p \times p$ identity matrix,

a = row vector of economic block values,

1 = $p \times 1$ 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):

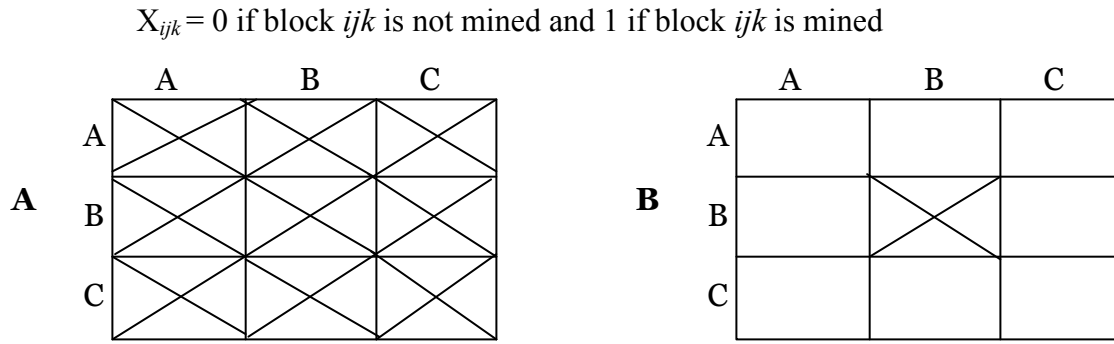


Figure 2: Plan view of the top blocks and bottom blocks in the model
(Modified after Gershon, 1983)

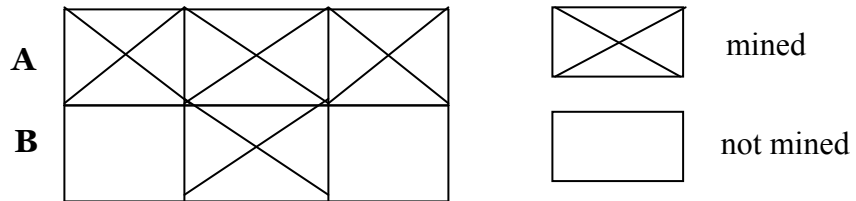


Figure 3: Cross-section of the block model

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):

$$-X_{AAA} + X_{BBB} \leq 0 \quad (20)$$

$$-X_{ABA} + X_{BBB} \leq 0 \quad (21)$$

$$-X_{ACA} + X_{BBB} \leq 0 \quad (22)$$

$$-X_{BAA} + X_{BBB} \leq 0 \quad (23)$$

$$-X_{BBA} + X_{BBB} \leq 0 \quad (24)$$

$$-X_{BCA} + X_{BBB} \leq 0 \quad (25)$$

$$-X_{CAA} + X_{BBB} \leq 0 \quad (26)$$

$$-X_{CBA}+X_{BBB} \leq 0 \quad (27)$$

$$-X_{CCA}+X_{BBB} \leq 0 \quad (28)$$

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):

$$\min f^T x \quad (29)$$

$$\text{subject to: } Ax \leq b \quad (30)$$

$$Aeq.x = beq \quad (31)$$

$$x \in (lb, ub) \quad (32)$$

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.

$$x3 = \begin{bmatrix} 1 & 0 & . & . & . & . & 0 \\ 0 & 1 & 0 & . & . & . & 0 \\ . & 0 & 1 & 0 & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & 0 & 1 & 0 \\ 0 & 0 & . & . & . & 0 & 1 \end{bmatrix} \quad x4 = \begin{bmatrix} -1 & 0 & . & . & . & . & 0 \\ 0 & -1 & 0 & . & . & . & 0 \\ . & 0 & -1 & 0 & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & 0 & -1 & 0 \\ 0 & 0 & . & . & . & 0 & -1 \end{bmatrix}$$

These three matrices $x3$, $x4$ and S are vertically concatenated to become a 117×18 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 81×1 column vector of zeros for the mining sequence matrix represented by $x5$, an 18×1 column vector of ones for the $x3$ matrix denoted by $x6$ and another 18×1 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 117×1 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.

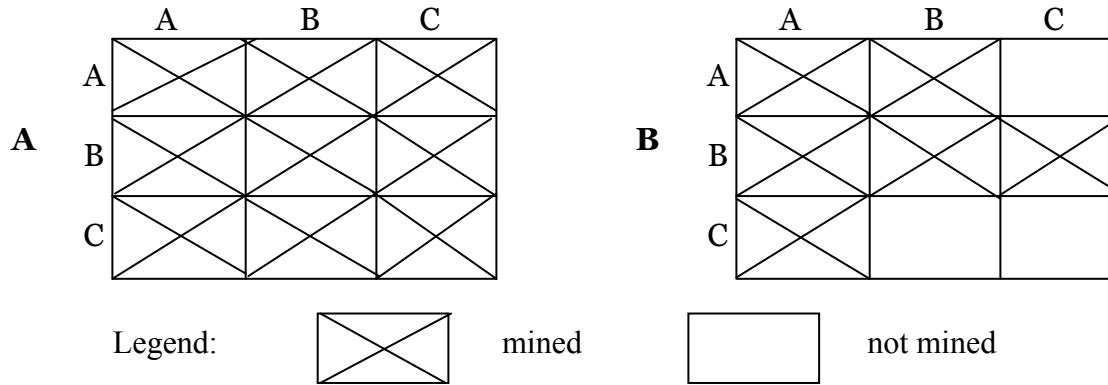


Figure 4: Plan view of the top blocks and bottom blocks in the model after optimization

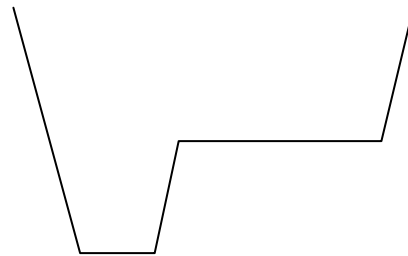


Figure 5: Cross-sectional view of the ultimate pit limit for the model after optimization
(Not to scale)

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 1×120 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 5088×120 . 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 5088×1 . 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.

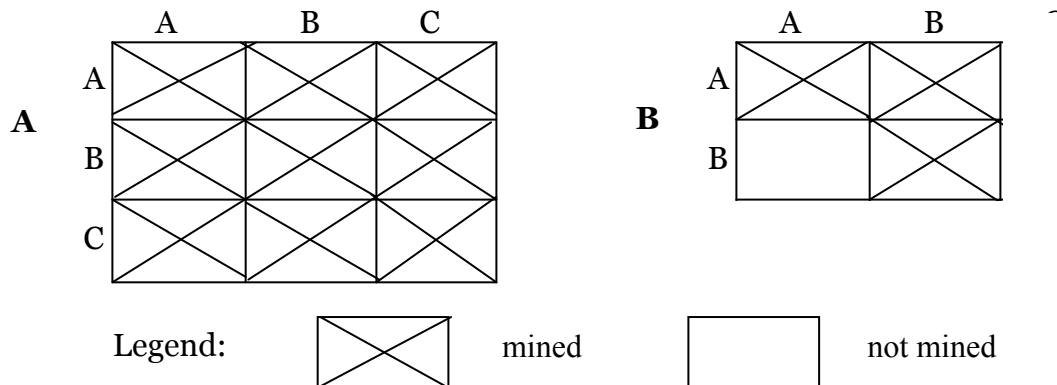


Figure 6: Plan view of some of the top blocks mined and the bottom block that was not mined in the model after optimization

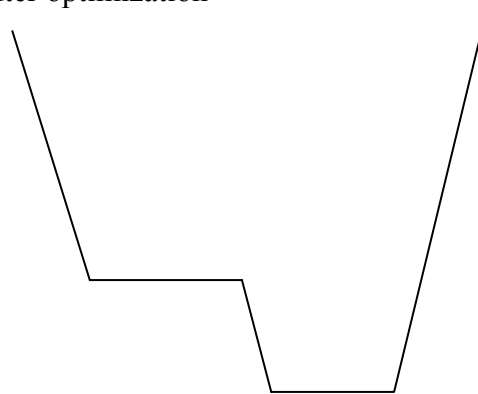


Figure 7: Cross-sectional view of the ultimate pit limit for the model after optimization
(Not to scale)

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
2. Minimum requirements
3. Getting started
4. Results
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.