The new Fundamental Tree Algorithm for production scheduling of open pit mines

Salih Ramazan

Rio Tinto, GPO Box A42, 152–158 St. Georges Tce, Perth, WA 6000, Australia

Received 10 July 2005; accepted 31 December 2005
Available online 20 March 2006

Abstract

The problem of annual production scheduling in surface mining consists of determining an optimal sequence of extracting the mineralized material from the ground. The main objective of the optimization process is usually to maximize the total Net Present Value of the operation. Production scheduling is typically a mixed integer programming (MIP) type problem. However, the large number of integer variables required in formulating the problem makes it impossible to solve. To overcome this obstacle, a new algorithm termed “Fundamental Tree Algorithm” is developed based on linear programming to aggregate blocks of material and decrease the number of integer variables and the number of constraints required within the MIP formulation. This paper proposes the new Fundamental Tree Algorithm in optimizing production scheduling in surface mining. A case study on a large copper deposit summarized in the paper shows substantial economic benefit of the proposed algorithm compared to existing methods.

Keywords: Large scale optimization; Fundamental Tree Algorithm; Production scheduling; Open pit mine optimization

1. Introduction

Solving the annual production scheduling problem is crucially important for surface mining since it determines the rate and quality of the production involving large cash flows, which can be hundreds of millions of dollars in magnitude. The scheduling problem is determining the optimal sequence of extracting the mineralized material from the ground, so that the total net present value (NPV) from the operation will be maximized, subject to a set of operational and physical constraints.
Subsequently, the orebody model containing the deposit attributes and economic expected value are used to determine final pit limits, which are the limits of the deposit up to which it is economic to mine. *Hochbaum and Chen (2000)* provide detailed discussion of commonly used methods in finding ultimate pit limits.

Annual production scheduling is a decision problem, that is; which blocks within the final pit limits should be mined in which year to maximize the total NPV within a set of defined constraints. The mining blocks are aggregated into larger volumes that each of them can be mined out in a year. The problem is finding the best way to aggregate these blocks so that NPV can be maximized when these volumes are mined.

The main physical constraint in open pit mining is the slope requirement; all the blocks on top preventing mining of a given block must be mined. Fig. 1 shows a cross-sectional view of a hypothetical 2D orebody model containing eight blocks which are square in shape. The block identification numbers are written on the left-top corner and expected economic values are written in the center. If the safe slope angles are assumed to be 45°, blocks 1, 2 and 3 would have to be excavated, or mined, before excavating block 6, or blocks 2, 3 and 4 have to be mined before block 7. In actual mining operations, the blocks that have to be mined before mining a given block are identified by using a cone template whose side angles are equal to the required slope angles for the deposit. If a deposit has different slopes at different regions, multiple cone templates are used. Further discussions and illustrations of the slope angles on a cone can be found in *Ramazan (2001)*.

There are often too many blocks, over 100,000, within the final pit limits to determine the optimum annual production schedule. To reduce the size of the problem, it is a common implementation to partition the material within the final pit limits into smaller volumes called “pushbacks” using one of the existing methods including *Ramazan (1996), Seymour (1995)* and *Whittle (1988)*. The pushbacks are used as a guide in determining annual production scheduling for the duration mine life.

Mixed integer and linear programming models are recognized as having significant potential for optimizing production scheduling in open pit mines, in which the objective is to maximize total discounted profit. However, MIP formulation of the production scheduling problem for open pit mines requires too many binary variables making it very difficult or impossible to solve. For example, if there are 10,000 mining blocks in a pushback to be scheduled over 3 years, it will require 30,000 binary variables to generate the MIP formulation. This would make it very difficult or even impossible to solve the MIP formulation.

Recognizing the strength of operations research techniques in optimization process, *Johnson (1968)* developed an LP scheduling model. He applied *Dantzig and Wolfe (1960)* decomposition principles to decompose the model into a master problem and a set of subproblems, which are solved using a maximum network flow (maxflow) algorithm developed by *Johnson (1968)*. However, this LP approach uses linear variables and leads to the mining of fractional blocks. *Dagdelen (1985)* used the Lagrangian decomposition method to decompose and solve a large MIP problem. The drawback of the approach is that the Lagrangian method might not always converge to an optimum solution if the Lagrange multipliers cannot result in a feasible solution. *Gershon (1983)* presented an LP approach together with MIP models for optimizing mine scheduling, which allowed partial block mining on the condition that the entire block preceding it has been mined. The author suggests that the models for optimizing production scheduling of open pit mines require too many binary variables and cannot be solved.

Alternative efficient methodologies for long-term production scheduling that reduce the required number of binary variables are presented in *Ramazan and Dimitrakopoulos (2004)*. The maximum reduction of the binary variables with the proposed methods is down to the number of ore blocks, or positive economic value blocks, in the model times the production periods minus one. However, this reduction may not be sufficient for some large open pit mines. *Tolwinski (1998)* proposes a method that combines the blocks on the same bench, termed “atoms”, and uses the Lerchs and Grossmann (LG) method (*Lerchs and Grossmann, 1965*) to generate pushbacks combining these atoms. The approach generates a production schedule using
dynamic programming. However, combining blocks into atoms may strongly reduce any possibility of the optimal solution depending on the size of the atoms which is not mentioned. The Milawa algorithm discussed in Whittle (2000) considers a few benches at each pushback as a variable and uses a search technique called the “step and stride” algorithm discussed in Warton (2000) to identify the regions of high value, rather than identifying individual mining blocks. This is a heuristic approach and does not guarantee an optimal solution. Godoy and Dimitrakopoulos (2004) applied simulated annealing optimization method for scheduling a large gold mine. Although the method seems promising, it does not explicitly include grade blending constraints in their model for application to mines with blending problems. Dimitrakopoulos and Ramazan (2004) developed an LP model that considers maximizing the chance of grade blending requirements and the feasibility of mining operations providing equipment access to the blocks. This LP model needs more testing in terms of satisfying the sequencing constraints for the open pit mines that have significant depth, or multiple blocks vertically.

In this paper, a new algorithm called “Fundamental Tree Algorithm” is proposed to reduce the number of binary variables required in MIP formulations and the number of constraints within MIP for optimizing annual production scheduling in open pit mines. The algorithm involves a newly developed linear programming model formulation to combine blocks into Fundamental Trees. A set of combined blocks is termed a “Fundamental Tree” if the combined blocks have the three properties: (i) can be mined without violating the slope constraints; (ii) the total economic value of the combined blocks as a fundamental tree must be positive; and (iii) a Fundamental Tree (FT) cannot be partitioned into smaller trees without violating (i) and (ii). It should be noted that all the blocks within the pushback considered for optimization must belong to a Fundamental Tree. It is shown in this paper why the LP model generates the Fundamental Trees (FTs) with the defined properties. After generating the FTs for a given mineral deposit, an MIP model modified from traditionally known MIP formulations is used to generate annual production schedules for mining.

### 2. Fundamental Tree Algorithm

The proposed Fundamental Tree Algorithm consists of five steps that are schematically illustrated in

![Fig. 2. Steps of the Fundamental Tree Algorithm.](image-url)
block, say \( j \), preventing mining two positive value blocks, say \( i \) and \( k \), under \( j \) and neither of the positive value blocks have enough value to mine \( j \) (\( V_i + V_j < 0 \), and \( V_k + V_j < 0 \)), revenue from both of the blocks \( i \) and \( k \) may need to support the cost of mining \( j \) if \( V_i + V_k + V_j > 0 \). When more than one positive value block supports the mining cost of a negative value block, it is called “joint support” in mining. If there is a block, say \( j \), that is located on top of another block, say \( i \), preventing block \( i \) to be mined, \( j \) is said to be an “overlying block” of \( i \) and \( i \) is said to be an “underlying block” of \( j \). The blocks that are sent to the processing plant to extract the metal from the rock, \( NR_i > PC_i \), are called “ore blocks” and otherwise “waste blocks”. It should be noted that there are more complex methods for classifying a block as ore or waste, but this definition is used herein during scheduling process for simplicity.

2.1. Steps of the FT Algorithm

The steps of the FT Algorithm illustrated schematically in Fig. 2 are discussed below:

**Step 1.** Generate a network for the blocks within a pushback. In the network, the blocks are represented by nodes and the mining slope requirement is represented by the arcs. The arcs in the network represent the node precedence relationship within the pit. An arc is set from each positive value node to all the overlying negative value nodes on the upper levels that have to be removed before accessing the underlying positive value node considered for removal. This node precedence relationship within the network is determined by using one or more cone templates as discussed previously.

**Step 2.** Find the cone value \( CV_i \) for each node \( i \) having positive value within the network. The economic values of all the nodes connected to node \( i \) with an arc set from node \( i \) are summed. This total economic value of the nodes is said to be the cone value of node \( i \).

**Step 3.** Assign coefficients to the positive value nodes according to their cone value. This may also be considered as ranking of these nodes by elevation, or level. On the top most level where one or more positive value nodes exist, the node with the highest cone value is assigned to 1, and the second highest cone value node is assigned to 2, and so on. For instance, if there are 3 positive value nodes on that level, the node with the smallest cone value is assigned to 3. Then, the ranking process moves down one level. If there are some positive value nodes on that level, the node with the highest cone value is assigned to 4. Otherwise, a lower level is searched for positive value nodes. The process is performed for all the positive value nodes within the network. If two or more positive value nodes on the same level have the same cone value (tie condition), the coefficients are assigned randomly; two nodes must not be assigned the same coefficient. For example, nodes \( i \) and \( j \) are on the same level, have the same cone value and \( C_{\text{max}} \) is the highest coefficient assigned previously, \( C_i \) is set to \( C_{\text{max}} + 1 \) and \( C_j \) is set to \( C_i + 1 \). Since nodes \( i \) and \( j \) have exactly the same cone values, the results would be the same if coefficients of \( i \) and \( j \) are exchanged.

**Step 4.** Set up the linear programming formulation as discussed in Section 3. After the problem formulation is ready, it can be solved using a commercially available solver.

**Step 5.** If the number of trees obtained is the same as the number of the trees obtained from the previous solution, go to Step 7 as the problem is considered to be at optimal, and the algorithm is stopped. If the number of trees is higher than the previous solution, keep the currently found connections between nodes and go to Step 6 to generate a new network to be used for iterating the algorithm. Initially, one can assume that whole network is 1 tree. Usually, two or three iterations are required for convergence.

**Step 6.** After obtaining a solution from LP model in Step 5, a new network is generated for this new solution. To generate the new network, first the arcs in the previous network having no flow in the LP solution are deleted. After arc deletion, the nodes that are connected to each other are identified. The deleted arcs that exist in the previous network between the connected nodes are added again to the network. After generating the network, go back to Step 2.

**Step 7.** Stop.
2.2. Application of steps of the FT Algorithm

Application of the Fundamental Tree Algorithm is discussed using an example that can be considered as a cross-sectional view of some blocks on two consecutive elevations, or levels, given in Fig. 1. The node numbers are written on the left top corner of each block and the expected economic value from mining and processing a block is written at the center of each block.

Step 1. A network is generated as shown in Fig. 3. Since nodes 6, 7 and 8 have positive economic values, the arcs are set from these nodes to the nodes on the upper level. For simplicity of the illustration, it is assumed that the blocks are the same in size and have to be mined with 45° slope angle in all directions. The arcs in the figure show the node precedence relationships based on the slope angle requirement. For example, in order to mine block 6, blocks 1, 2 and 3 must be mined, or to mine block 7, blocks 2, 3 and 4 must be mined, and so on.

Step 2. If the economic value of node \( i \) is \( V_i \), then \( CV_i = V_i + V_1 + V_2 + V_3 = +6 - 1 - 2 - 2 = +1 \). Similarly, \( CV_7 = -3 \), \( CV_8 = -2 \).

Step 3. Coefficients \( C_i \) are assigned to positive value nodes according to \( CV_i \) value and the levels where nodes are located. Since \( CV_6 \) is greater than \( CV_7 \) and \( CV_8 \), \( C_6 \) is set to 1; and since \( CV_8 > CV_7 \), \( C_8 \) is set to 3 and \( C_8 \) is 2.

Step 4. The initial problem formulation and its solution are given in Fig. 4a and the network representation of the solution is in Fig. 4b. The LP formulation, terms and notations used on the figure are explained in the next section. Fig. 4b shows that there are now two trees, which is greater than the initial one tree considering the initial assumption that starting network is a tree. Now the algorithm moves to Step 6.

Step 6. After obtaining a solution from the LP model in Step 5, a new network (see Fig. 4b) is generated. To generate the new network, first the arcs in the previous network having no flow in the LP solution are deleted. After arc deletion, the nodes that are connected to each other are identified. The deleted arcs that exist in the previous network between the connected nodes are added again to the network. For example, in Fig. 4b, if node 3 were connected to node 7 by an arc, all the arcs in the initial network would have to be kept since all the nodes would belong to the same tree. That means although there is no flow on

\[
\begin{align*}
\text{Minimise } & \quad f_{61} + f_{62} + f_{63} + 3f_{72} + 3f_{73} + 3f_{74} \\
& + 2f_{83} + 2f_{84} + 2f_{85} \\
\text{Subject To } & \quad f_{61} \leq 6 \\
& \quad f_{72} \leq 3 \\
& \quad f_{84} \leq 4 \\
& \quad f_{31} = 1.001 \\
& \quad f_{51} = 1.001 \\
& \quad f_{51} = 1.001 \\
& \quad f_{61} = 1.001 \\
& \quad f_{62} + f_{72} + f_{82} = 0 \\
& \quad f_{63} + f_{73} + f_{83} - f_{31} = 0 \\
& \quad f_{74} + f_{84} - f_{41} = 0 \\
& \quad f_{85} - f_{51} = 0 \\
& \quad f_{85} - f_{61} - f_{62} - f_{63} = 0 \\
& \quad f_{72} - f_{73} + f_{74} + f_{84} = 0 \\
& \quad f_{85} - f_{63} - f_{84} - f_{85} = 0 \\
& \quad f_{61} - f_{62} = 0 \\
& \quad f_{32} = 0.00200 \\
& \quad f_{33} = 0.00000 \\
& \quad f_{34} = 0.00000 \\
& \quad f_{41} = 1.999998 \\
& \quad f_{45} = 2.001000 \\
\end{align*}
\]

Variable | Value
--- | ---
\( f_{61} \) | 5.003000
\( f_{62} \) | 0.002000
\( f_{63} \) | 0.000000
\( f_{72} \) | 2.001000
\( f_{73} \) | 2.001000
\( f_{74} \) | 2.001000
\( f_{83} \) | 1.999998
\( f_{84} \) | 2.001000

Fig. 3. Network representation of the 2D block model in Fig. 1. A source node \( s \) and a sink node \( t \) are added.

Fig. 4a. The LP fundamental tree problem formulation on the left-hand side in the first iteration and the solution on the right-hand side.
the arc from node 8 to node 3 in the current solution, the arc would still be kept since nodes 3 and 8 would be members of the same tree. After generating the network, go to Step 2.

Step 2. The cone value of node 6 CV₆ is unchanged, 1. Since node 2 and 3 belong to different tree than node 7, there is no arc between node 7 and nodes 2 and 3 in the current network. So, the value of nodes 2 and 3 are not considered in calculating the cone value of node 7. Since there is an arc from node 7 to only node 2, CV₇ = +3 − 2 = +1, CV₈ = +4 − 2 − 2 = 0.

Step 3. Coefficients Cᵢ are assigned to positive value nodes according to new CVᵢ values. C₆ is set to 1; and since CV₇ > CV₈ C₇ is set to 2 and C₈ is 3.

Step 4. The iterated problem formulation and its solution are given in Fig. 5a and the network representation of the solution is in Fig. 5b. The iterated LP formulation is explained in the next section.

Step 5. Since the number of trees from the current solution 3 is greater than the previous number of trees 2, the algorithm should normally move to Step 6. However, since there is no tree containing more than one positive value node in the results, it is impossible for any tree to be partitioned into sub-trees and repetition can be skipped in this case by going to Step 7.

Step 7. Stop the algorithm.
taining a total positive value. It is noted that tree numbers do not necessarily correspond to the sequence in which trees should be mined although it is the case in this small example. The mining sequence of the trees is determined using an MIP production scheduling model that will be discussed in Section 5.

3. The LP formulation to generate fundamental trees

This section discusses the LP formulation of the Fundamental Tree Algorithm. The objective function is minimization of arc connections in the network weighted by the assigned ranks. The objective function is expressed as

$$\text{Min} \sum_i \sum_j C_{ij} f_{ij},$$

(2)

where $C_{ij}$ is the coefficient for node $i$, which is for $n$ positive value nodes discussed in the previous section; $f_{ij}$ is the flow from node $i$ to node $j$; $j$ is the index for the negative value nodes connected to node $i$ with an arc coming from $i$. If there is a flow over an arc, the arc is kept on the network to be generated from the output of this LP model.

If there is one or more positive value nodes on level 1, there is no need to include them in the formulation because there are no arcs formed from these nodes. Since they already possess the defined properties of the fundamental tree, they are only considered in calculating cone values at Step 2, but not in the LP formulation.

The objective function is constructed in a way that the arcs will be set from high cone value nodes to support the negative nodes above them. In a given level, it is considered that the highest cone value node, say node $i$, has the highest chance of supporting all the negative value nodes above node $I$, preventing node $i$ from being mined. Therefore, if the arcs to be constructed are started from the highest cone value node (lowest objective coefficient), considering the model constraints, the number of joint supports for negative value nodes will be minimized. This ranking, or coefficients, in the objective function together with the model constraints has the most effect in establishing the third property of FT, which an FT cannot be partitioned into sub-FTs. The coefficients also have some role in making the FTs obey slope constraints although the effect is not as direct as in Eq. (4). Since the arc connections are prioritized from higher cone value nodes, the Fundamental Trees are generated in a way that higher value blocks become feasible for mining before the lower value blocks for the MIP scheduling model. This is a desirable condition for NPV maximization objective of annual production scheduling.

A positive value node is limited in flow capacity to its value, which is constrained from source node, $s$, to positive nodes. The physical meaning of these constraints for mining is that the expected revenue from mining an ore block cannot support the mining waste where the cost is higher than the revenue. The constraint formulation is expressed as below:

$$f_{si} \leq V_i \quad \text{for all} \ i,$$

(3)

where $f_{si}$ is the flow sent from source node, $s$ to node $i$, $V_i$ is the economic value of block $i$, which is set for only positive value nodes.

Constraints are imposed on the arcs going from negative valued nodes to sink node $t$ to ensure that the cost of mining waste blocks are justified by the income to be generated from mining the ore blocks. A small extra value $\zeta$ is also applied to negative value nodes to ensure that precedence relationship will not be violated by the trees. $\zeta$ is set to a very small number that will not be ignored by the solver such as 0.001. These constraints ensure that minimum economic value of a tree is greater than, or equal to $\zeta$, which is strictly positive. This is the first pre-defined property of a Fundamental Tree.

Without using $\zeta$ value, if an overlying negative node is fully supported by an underlying positive value node, the total value of negative and positive nodes could be zero without requiring a joint support. That would not only generate zero value trees violating the defined property of Fundamental Trees for being strictly positive, but also cause violation in slope requirements, which is further discussed in Section 4.

Note that if $\zeta$ is set to a number too high, the LP model may not produce fundamental trees with the defined properties. The total of the added $\zeta$ values for all the overlying connected nodes should be kept below 1. Otherwise, some trees may violate the last pre-defined property of a Fundamental Tree; one or more trees may be partitioned into sub-trees having the first two pre-defined properties of fundamental trees if $\zeta$ value is set too high. Since the economic values in mining are sufficiently large, thousands in magnitude, they are rounded to integer values.
The functionality of this rounding is discussed in Section 4, Property II. This constraint formulation is expressed as

\[ f_{ji} = -V_j + \xi, \quad (4) \]

where \( \xi \) is a small positive decimal number; \( V_j \) is the value of the negative value node \( j \); and \( t \) is the sink node.

The total flow coming to a negative value node must be equal to the flows leaving that node. If the number of positive value nodes that arcs are set from towards the negative value node \( j \) is \( O_j \), then the mass balance constraints around each negative value node is expressed as

\[ \sum_{i=1}^{O_j} f_{ij} - f_{ji} = 0. \quad (5) \]

The total flow coming to a positive value node \( i \) from the source node must be equal to the total flow leaving that ore block. If the number of waste blocks overlying positive node \( i \) is \( W_i \), the mass balance constraints for the positive value nodes are expressed as

\[ f_{si} - \sum_{j=1}^{W_i} f_{ij} = 0. \quad (6) \]

The initial LP formulation and solution for the example network model given in Fig. 3 are illustrated in Fig. 4a. Fig. 4b is generated by deleting the arcs that are not used by the LP model from the initial network. The iterative LP formulation is generated using the current network of the system as stated in Section 2.1 shown in Fig. 4b. The LP model and the solution are given in Fig. 5a. The solution network in Fig. 5b is generated by deleting the arcs with no flow on them from the previous network. In the example, the number of binaries required is reduced from 8 to 3 by the use of FT.

After generating the fundamental trees for a given orebody model, the annual production scheduling can be formulated as an MIP model treating each tree as a block having certain attributes discussed in Section 5.

4. Properties of the Fundamental Trees generated by the FT Algorithm

The Fundamental Trees have three major properties by definition as given earlier in Section 1. These properties are discussed in this section.

**Property 1.** A tree obtained by the FT Algorithm always has a total cumulative value greater than zero. Eq. (3) ensures that the positive value nodes cannot send more flow to the overlying negative value nodes than their own value. Eq. (4) ensures that all the negative value nodes are totally supported by underlying positive value nodes. For a given tree, constraints in Eqs. (3) and (4) can be written as follows:

\[ \sum_i V_i \geq \sum_i f_{si}, \]

\[ \sum_j (-V_j + \xi) = \sum_i f_{si}. \]

Therefore,

\[ \sum_i V_i \geq \sum_j (-V_j + \xi), \]

where \( \sum_i V_i \) is the sum of all the positive value nodes in a tree, and \( \sum_j (-V_j + \xi) \) is the sum of all the negative value nodes in the tree plus the small epsilon assigned to those nodes. Therefore, the minimum value of a tree is the sum of all the epsilons assigned to the negative value nodes within the tree. Since \( \xi \) is always a positive number, the total economic value of a tree generated by the FT Algorithm is always greater than zero. In the given example, the first and second trees both have a cumulative value of +1 and the third tree has a value of +2.

The property of the trees to be positive may raise the question: what happens if a tree has negative value. It is stated in Section 2 that the FT Algorithm is implemented to the blocks within a pushback that has to be determined using one of the true optimizing methods. This means that the total value of all the blocks, or nodes, considered must be positive. If all the blocks are connected to each other, forming one tree, the total value is positive and the entire pushback forms a feasible solution for the LP formulation. Therefore, there is at least one feasible solution to the LP formulation although it is not the optimal solution in most cases.

**Property 2.** Slope constraints are not violated when fundamental trees are removed from the network one at a time respecting the sequencing between them. It is shown in Property 1 that all the overlying negative value nodes connected to positive value nodes in the initial network must be supported by underlying positive value nodes having arcs towards the negative value nodes, due to the model constraints. Since
Property 1 provides a feasible solution to the Fundamental Tree LP problem, all the overlying negative value nodes will be supported.

The arcs in the initial network are set from each positive value node to all its overlying negative value nodes which are forced to be fully supported. In a situation where a positive value node, say \( i \), cannot fully support all its connected overlying nodes, the overlying node currently considered for support will require extra support from another positive value node, say \( k \). Due to the small \( \xi \), the flow capacity of the positive value node \( i \) cannot be fully consumed for supporting overlying negative nodes without requiring support from another positive value node \( k \). It is because of the fact that the flow capacities (economic values of blocks) of the nodes are rounded to integers before setting the LP formulations whilst \( \xi \) is a small decimal number. Therefore, either the positive value node \( i \) will have to be able to support all the overlying negative nodes, or it will have to require support from another positive value node \( k \) to the same overlying negative value node where the flow capacity of the node \( i \) is fully consumed. This characteristic of \( \xi \) makes the resultant trees always obey the slope constraints.

Property 3. A tree found by the FT Algorithm cannot contain a subset of trees that also can be a fundamental tree. If a sub-tree exists that is also a fundamental tree, it means there is a set of one or more positive value nodes in the sub-tree that has sufficient flow capacity to support all its overlying waste. If this were so, the LP solution would have identified the sub-tree as a fundamental tree due to the optimality of the objective function. It is always better to send the flows from the smaller coefficient node in the objective function than sending some of the flows from a higher coefficient node since each coefficient in the objective function is unique.

5. Optimization of annual production scheduling using fundamental trees

MIP scheduling formulations of the trees are done by treating each FT as a block having certain amount of ore tons with average grade of commodities and possibly some waste tons. Since binary variables are assigned to the FT’s instead of blocks, the number of sequencing constraints, given in Eqs. (8) and (9) in Section 5.1, are also reduced significantly. Traditionally, sequencing constraints are written for all the blocks below the first level in the model whilst they are written only for FTs that are substantially less than the number of blocks with the proposed method. The substantial reduction in MIP problem size by applying FT Algorithm reduces the required solution time significantly and that makes it possible to apply MIP model to large open pit mines.

5.1. MIP production scheduling formulation with Fundamental Trees

The optimization model is maximization of the NPV, or discounted economic value, to be generated from the mining operation and hence for scheduling \( N \) Fundamental Trees over \( Y \) years, the objective function can be expressed as

\[
\text{Max } \sum_{p=1}^{Y} \sum_{i=1}^{N} \{ (\text{DEV}_{pi}) \times O_{ip} + (\text{DWC}_{pi} \times W_{ip}) \}, \tag{7}
\]

where \( O_{ip} \) is defined as a binary variable for \( p < Y \), valued 1 if the ore ton in FT \( i \) is scheduled for period \( p \), and 0 otherwise; when \( p = Y \), \( O_{ip} \) is the percentage of the block scheduled in year \( p \); \( W_{ip} \) is a linear variable representing the tons of the waste to be mined at period \( p \) from fundamental tree \( i \); \( \text{DEV}_{pi} \) discounted economic value to be generated from mining and processing all the ore tons in year \( p \) from FT \( i \); \( \text{DWC}_{ip} \) discounted average cost of mining one ton waste from FT \( i \) in year \( p \). If the schedule of the mine until the last year is optimized using binary variables, the last year’s variables do not need to be defined as binary since the slope requirement will be satisfied from the previous years. Further discussions on this issue can be found in Ramazan and Dimitrakopoulos (2004).

If the economic discount rate is \( d \), discounting factor in year \( p \), \( \text{DF}_{p} \), can be calculated by

\[
\text{DF}_{p} = 1/(1.0 + d)^{p}. \tag{7a}
\]

The economic value to be generated by processing all the ore tonnage at time 0 from FT \( i \), \( \text{DEV}_{0i} \), can be determined by

\[
\text{DEV}_{0i} = \sum_{j} V_{j},
\]

where \( j \) is the index for all the ore blocks that belongs to FT \( i \). The objective function coefficient \( \text{DEV}_{pi} \) can be found by
DEV\textsubscript{pi} = DEV\textsubscript{0} \ast DF\textsubscript{p}. \hfill (7b)

Similarly, the cost of mining the waste material can be calculated using Eq. (1) and DWC\textsubscript{ip} is determined by

DWC\textsubscript{ip} = DWC\textsubscript{00} \ast DF\textsubscript{p}, \hfill (7c)

where DWC\textsubscript{00} = (∑\textsubscript{h=1}hV\textsubscript{h})/T\textsubscript{Wi}, which is the cost of mining one ton of waste material; \( h \) is the index for waste blocks that belongs to FT \( i \) and \( T\textsubscript{Wi} \) is the total tons of waste material within FT \( i \).

The number of binary variables (NB) required for the MIP model using FTs is equal to the number of FTs multiplied by one less the total time period to be scheduled, which can be expressed as, \( NB = N \ast (Y - 1) \).

It is very important to maintain correct slope angles for the walls surrounding mined out areas in open pit mines. The slope cannot be too steep to prevent the side walls collapsing towards the zones of mining activity and it cannot be too shallow so as to avoid extracting too much waste material which can substantially increase the mining cost. Precedence of the FTs in mining operations can be identified using cone templates as in moving cone method (Lemieux, 1979). If \( j \) is the index for the fundamental trees that have to be removed before being able to remove tree \( i \) in period \( p \), the sequencing formulations is written for each \( j \) as follows:

\[ O\textsubscript{ip} - \sum_{i=1}^{p} O\textsubscript{j} \leq 0. \hfill (8a) \]

The sequencing formulation can also be written as in Eq. (8b), which is 1 constraint for all overlying FTs instead of one constraint for each overlying FT. However, this type of formulation containing fewer constraints in this way does not always reduce solution time compared to Eq. (8a). The two types of formulations are discussed in Ramazan and Dimitrakopoulos (2004) in terms of solution times. If \( N_j \) is the number of overlying FTs for FT \( i \), Eq. (8b) is expressed as

\[ N_j O\textsubscript{ip} - \sum_{j=1}^{N_j} \sum_{i=1}^{p} O\textsubscript{j} \leq 0. \hfill (8b) \]

Another set of constraints that may also be considered as sequencing is that before mining the ore tonnages from a given FT, all the waste tonnages must be mined out in that FT. If \( T\textsubscript{Wi} \) is the total tons of waste material within tree \( i \), these constraints can be set as follows:

\[ T\textsubscript{Wi} O\textsubscript{ip} - \sum_{i=1}^{p} W\textsubscript{it} \leq 0. \hfill (9) \]

To achieve a possible low cost at the processing plants, the ore material sent to the processing plant is sometimes constrained to have a grade (gram per ton, ounce per ton, or % metal contained) within a range of values. Assume that \( G_i \) is the average grade of FT \( i \), and \( T\textsubscript{Oj} \) is the total ore tonnage in FT \( i \), \( G\textsubscript{min} \) and \( G\textsubscript{max} \) are the periodical minimum and maximum grade range requirements at the processing plant. Then, the constraint formulations for \( n \)-fundamental trees and a period \( p \) is expressed as

\[ \sum_{i=1}^{N} G_i T\textsubscript{Oj} O\textsubscript{ip} \geq G\textsubscript{min}, \]

which can be expressed in linear form as

\[ \sum_{i=1}^{N} (G_i - G\textsubscript{min}) T\textsubscript{Oj} O\textsubscript{ip} \geq 0, \hfill (10a) \]

\[ \sum_{i=1}^{N} (G_i - G\textsubscript{max}) T\textsubscript{Oj} O\textsubscript{ip} \leq 0. \hfill (10b) \]

The processing plant is limited by the amount of ore material it can process during a time period. A lower bound may also be applied in some cases to ensure that the processing plant will not be idle, and to avoid or reduce undesirable fluctuations in the production. For \( n \)-FTs and a period \( p \), the limitations of the processing mill capacity as lower and upper bounds can be written as follows:

\[ \sum_{i=1}^{N} T\textsubscript{Oj} O\textsubscript{ip} \geq PC\textsubscript{min}, \hfill (11a) \]

\[ \sum_{i=1}^{N} T\textsubscript{Oj} O\textsubscript{ip} \leq PC\textsubscript{max}, \hfill (11b) \]

where \( PC\textsubscript{min} \) and \( PC\textsubscript{max} \) are the minimum and maximum capacity of mill.

Since the production scheduling is performed within the final pit, or a pushback, constraints are needed to ensure that all the material considered in optimization will be mined out. These constraints ensure that ore tons in each FT must be mined in a single period:

\[ \sum_{p=1}^{y} O\textsubscript{ip} = 1. \hfill (12) \]
Stripping ratio is defined as the ratio of total amount of waste material mined to the amount of ore produced. At some mining operation, stripping ratio is also limited at a maximum value. If the stripping ratio is SR, for a period $p$ considering $n$-FT, the constraints not to exceed a given stripping ratio limit is defined as

$$SR = \frac{\sum_{i=1}^{N} W_{ip}}{\sum_{i=1}^{N} T_{Oi} O_{ip}}.$$ 

Then

$$\sum_{i=1}^{N} W_{ip} - SR_{\text{max}} \sum_{i=1}^{N} T_{Oi} O_{ip} \leq 0,$$  

(13)

where $SR_{\text{max}}$ is the maximum stripping ratio allowed during any year.

The total amount of material mined during a year cannot be more than the total mining capacity of all the equipments, MCap. This requirement in open pit mining operations is expressed by

$$\sum_{i=1}^{N} (W_{ip} + T_{Oi} * O_{ip}) \leq \text{MCap}.$$  

(14)

In some of the mining operations, the total tonnage of waste to be stripped out in a period is restricted not to exceed a waste stripping amount $WC_{\text{max}}$. For $n$-FTs and a given period $p$, the formulation is

$$\sum_{i=1}^{N} W_{ip} \leq WC_{\text{max}}.$$  

(15)

6. Application of the MIP model with fundamental trees

The application of MIP model in annual production scheduling of open pit mines is shown using the example containing three FTs. Although some of the constraints in this application may seem redundant, or the result may also be obvious, all the constraints assumed to exist in this case are written for illustration. The FTs determined previously are illustrated in Fig. 6 representing them as blocks. The scheduling requirements for the mine are set out in Table 1. Assuming a 10% economic discount rate, discount factors for the two years are found as shown in Eq. (7a); $DF_1 = 1/(1.10)^1 = 0.91$ and $DF_2 = 1/(1.10)^2 = 0.83$. Using these factors, discounted economic values for each FT and each time period $p$ can be calculated by Eq. (7b) as below:

$$DEV_{11} = DEV_{10} \times DF_1 = 6 \times 0.91 = 5.46$$
for FT = 1, $p = 1$,

$$DEV_{12} = DEV_{10} \times DF_2 = 6 \times 0.83 = 4.98$$
for FT = 1, $p = 2$,

$$DEV_{21} = DEV_{20} \times DF_1 = 3 \times 0.91 = 2.73$$
for FT = 2, $p = 1$,

$$DEV_{22} = DEV_{20} \times DF_2 = 3 \times 0.83 = 2.49$$
for FT = 2, $p = 2$,

$$DEV_{31} = DEV_{30} \times DF_1 = 4 \times 0.91 = 3.64$$
for FT = 3, $p = 1$,

$$DEV_{32} = DEV_{30} \times DF_2 = 4 \times 0.83 = 3.32$$
for FT = 3, $p = 2$.

Discounted cost of mining one ton of waste material for the example problem is calculated as follows:

$$DWC_{01} = 5/30 = 0.167, \quad DWC_{02} = 2/10 = 0.2$$
and $DWC_{03} = 2/10 = 0.2$.

Using Eq. (7c),

$$DWC_{11} = 0.167 \times 0.91 = 0.152,$$
and similarly,

\[ \text{DWC}_{12} = 0.138, \quad \text{DWC}_{21} = 0.182, \]
\[ \text{DWC}_{22} = 0.166, \quad \text{DWC}_{31} = 0.182 \quad \text{and} \]
\[ \text{DWC}_{32} = 0.166. \]

By inserting the calculated values to the objective function in Eq. (7), the following objective function is obtained for the example problem:

\[
\begin{align*}
\text{Max } & 5.46O_{11} + 4.98O_{12} + 2.73O_{21} + 2.49O_{22} \\
& + 3.64O_{31} + 3.32O_{32} - 0.152W_{11} - 0.138W_{12} \\
& - 0.182W_{21} - 0.166W_{22} - 0.182W_{31} - 0.166W_{32}.
\end{align*}
\]

The sequencing constraints between the second tree and the first tree given in Eq. (8a) for the example are expressed as \( O_{21} - O_{11} \leq 0 \), that means if the ore from tree 2 is taken in period 1, the ore from tree 1 must also be taken in period 1. The same constraint for the second period is written as \( O_{22} - O_{11} - O_{12} \leq 0 \), that means if the ore from tree 2 is excavated in period 2, the ore from tree 1 must be excavated in period 1, or period 2. Similarly, the constraints for the third tree is written as \( O_{31} - O_{21} \leq 0 \) and \( O_{32} - O_{21} - O_{22} \leq 0 \). Since the constraints from second tree to first tree are written, the constraints between the third tree and first tree are not necessary.

The constraints in Eq. (9) to represent the requirement of mining waste material within each tree before being able to mine ore material are written for the example problem as \( 30 \ast O_{11} - W_{11} \leq 0 \), which means if the ore within FT1 is mined in the first period \( (O_{11} = 1) \), then, the waste variable of the first tree in the first period, \( W_{11} \), must be equal to the waste tons in that tree, which is 30 tons.

The same constraint for the second period is written as \( 30 \ast O_{12} - W_{12} - W_{11} \leq 0 \), which means if the ore material in the first tree is mined in the second period, waste material in that tree can be mined in the first period, or second period. The same constraints for the 2nd and 3rd FTs can be written similarly as follows:

\[
\begin{align*}
10 \ast O_{21} - W_{21} & \leq 0, \\
10 \ast O_{22} - W_{21} - W_{22} & \leq 0, \\
10 \ast O_{31} - W_{31} & \leq 0, \\
10 \ast O_{32} - W_{31} - W_{32} & \leq 0.
\end{align*}
\]

The grade constraints given in Eqs. (10a) and (10b) are illustrated for the problem as below:

\[
\begin{align*}
(4.4 - 4.0) \ast 18 \ast O_{11} + (3.8 - 4.0) \ast 10 \ast O_{21} + (4.5 - 4.0) \ast 10 \ast O_{31} & \geq 0, \\
(4.4 - 4.0) \ast 18 \ast O_{12} + (3.8 - 4.0) \ast 10 \ast O_{22} + (4.5 - 4.0) \ast 10 \ast O_{32} & \geq 0, \\
(4.4 - 4.5) \ast 18 \ast O_{11} + (3.8 - 4.5) \ast 10 \ast O_{21} + (4.5 - 4.5) \ast 10 \ast O_{31} & \leq 0, \\
(4.4 - 4.5) \ast 18 \ast O_{12} + (3.8 - 4.5) \ast 10 \ast O_{22} + (4.5 - 4.5) \ast 10 \ast O_{32} & \leq 0.
\end{align*}
\]

The processing capacity is assumed to be limited with an upper and lower bound in this model. The requirements for this processing plant can be expressed using Eqs. (11a) and (11b) as

\[
\begin{align*}
18O_{11} + 10O_{21} + 100O_{31} & \geq 10, \\
18O_{12} + 10O_{22} + 100O_{32} & \geq 10, \\
18O_{11} + 10O_{21} + 100O_{31} & \leq 20, \\
18O_{12} + 10O_{22} + 100O_{32} & \leq 20.
\end{align*}
\]

The constraints in Eq. (12) ensures that an FT must be mined in one of the periods. We suggest these equality constraints be written before other constraints due to efficiency of the problem although we are following a different order in this example to follow the equations given in Section 5.1. The constraints are applied for the example problem as

\[
\begin{align*}
O_{11} + O_{12} & = 1, \\
O_{21} + O_{22} & = 1, \\
O_{31} + O_{32} & = 1.
\end{align*}
\]

Eq. (13) representing the limit on the stripping ratio in each period is applied to the example problem as

\[
\begin{align*}
W_{11} + W_{21} + W_{31} - 2 \ast 18 \ast O_{11} - 2 \ast 10 \ast O_{21} - 2 \ast 10 \ast O_{31} & \leq 0, \\
W_{12} + W_{22} + W_{32} - 2 \ast 18 \ast O_{12} - 2 \ast 10 \ast O_{22} - 2 \ast 10 \ast O_{32} & \leq 0.
\end{align*}
\]

The total capacity of the equipments in removing ore and waste material are limited during a given year at a maximum of 50 tons. This requirement given in Eq. (14), written for the example problem as

\[
\begin{align*}
W_{11} + W_{21} + W_{31} + 18 \ast O_{11} + 10 \ast O_{21} + 10 \ast O_{31} & \leq 50, \\
W_{12} + W_{22} + W_{32} + 18 \ast O_{12} + 10 \ast O_{22} + 10 \ast O_{32} & \leq 50.
\end{align*}
\]
Solving the MIP problem results in mining FT 1 during the first year \((O_{11} = 1 \text{ and } W_{11} = 30)\), and FT 2 and FT 3 are mined during the second year \((O_{22} = 1, O_{32} = 1, W_{22} = 10 \text{ and } W_{32} = 10)\). The objective value of the MIP is 3.39, which is the optimal NPV that can be generated from mining the example orebody model. Using the FT Algorithm, the number of binaries is reduced from 16 to 6.

7. Case study

One of the case studies is performed on a large copper mine in Southern Peru that contains about 1 million blocks in the orebody model and has two ore processors, leaching process and mill plant. The proposed Fundamental Tree Algorithm reduced the number of blocks within the ultimate pit limits from 38,457 to 5512 fundamental trees. The MIP scheduling model is applied to develop a production schedule over 8 years for the mine. This was impossible to perform without applying the Fundamental Tree Algorithm due to large number of binary variables and large number of sequencing constraints required in formulating the problem. In one of the pushbacks scheduled over 4 years, the number of blocks were about 12,600. This would normally require 37,800 binary variables in the traditional MIP formulations, but FTA combined the blocks into 1640 trees, resulting the MIP scheduling model containing 4920 binary variables being solved in about 36 minutes with 5% gap.

The number of constraints is also significantly reduced by using FTs instead of blocks in the MIP formulation. Traditionally, sequencing constraints would be required for about 12,350 blocks excluding the top level. The proposed LP algorithm requires the sequencing constraints for only 1640 FTs, which is about 7.5 times fewer constraints.

In mining, a gap less than 10% is often accepted as a good solution considering the uncertainty in estimated block grades as input to the model. For this single project, the method produced $25M (~7\%) higher NPV than the best NPV generating schedule obtained among three commonly used traditional software packages including MINTEC’s M821V, Earthworks’ NPV scheduler, and Whittle’s Milawa schedulers in the Four-X program. The NPV values are calculated after designing the haul roads and smoothing the pits that makes the profit achievable through the actual mining operation. A detailed description of the proposed scheduling process, calculations and various maps can be found in Ramazan (2001) and the three traditional scheduling details are given in Bernabe (2001).

8. Conclusions

In this paper, a mathematical programming model is developed using linear variables to produce Fundamental Trees for any given orebody model for open pit mines. This model does not include any integer variables so that it solves large problems without running into time constraints. The number of integer variables required for the MIP scheduling formulation is decreased substantially using the fundamental trees. Since the number of FTs produced is significantly less than the number of blocks in the model, the number of constraints are also substantially less than the traditional MIP scheduling formulations. Therefore, large open pit production scheduling can be optimized by MIP modelling for any objective such as maximizing NPV of a given mine project, or to solve hard ore quality and grade blending problems.

It should be noted that alternative solutions may be available to find the fundamental trees, which means the same number of fundamental trees may be determined with different configurations of connected blocks. However, determining all the possible configurations of the trees and measuring their effect on the scheduling are not performed. A significant difference in the scheduling results for different block configuration of fundamental trees is not expected due to the structure of the FTA that makes the higher cone value blocks feasible to mine before the other blocks.

Acknowledgements

The author express special thanks to Prof. Roussos Dimitrakopoulos for reviewing this paper. The project was partly sponsored by Natural Resources Research Institute of Minnesota.

References


Salih Ramazan holds a PhD degree in mining engineering with a minor in computer science and mathematics, and MSc degree in mining engineering from Colorado School of Mines, USA, and ME degree in Geostatistics from Ecole Nationale Superieure des Mines de Paris, France. He is specialized in operations research methods especially development of efficient large size mixed integer and stochastic programming models and applications to production scheduling of large open pit mines. He has worked as a research scientist for 4 years in developments of new stochastic optimization methods and algorithms at the University of Queensland, Australia, and is currently a mine planning engineer at Kalgoorlie Consolidated Gold Mines, Australia.