A HYBRID LOCAL SEARCH/MIXED INTEGER PROGRAMMING APPROACH TO OPEN PIT CONTROLLED PHASE-DESIGN

Hooman Askari-Nasab, Mining Optimization Laboratory (MOL). * Corresponding Author: hooman@ualberta.ca, +1 (780) 492 4053.

Clemens Mieth, School of Mining and Petroleum Engineering, University of Alberta, Edmonton, Canada.

Mohammad Tabesh, School of Mining and Petroleum Engineering, University of Alberta, Edmonton, Canada.

Abstract

The life-of-mine production schedule defines the strategy of displacement of ore, waste, and overburden over the mine-life. Long-term production schedules are the backbone of short-term planning and day-to-day mining operations. The objective of production scheduling is to determine the sequence of extraction and displacement of material in order to optimize management’s objective over the time-horizon of interest. The objective of this paper is to develop, verify, and present new optimal methodologies for three interrelated key component of open-pit mine planning: controlled optimal phase-design, characterization of selective mining-units, and long-term production scheduling optimization. A new hybrid solution methodology for open-pit phase-design using integer programming and a local search heuristic is presented. The phase-design methodology maximizes the net value of the mine with control over the tonnage of mineralized material and waste within each pushback. The first step in the phase-design module is based on the linear relaxation model of an integer programming formulation for open pit push-back optimization. The linear relaxation provides an approximation to the optimal solution for the push-back and it is also used to reduce the problem size. If the residual problem is small enough it is solved directly using an IP-solver. Otherwise, a greedy heuristic with a statistical component is used to create an initial solution followed by a local search heuristic to improve the greedy solution. Next, a hierarchical clustering approach with size and shape control is presented, which aggregate blocks into minable polygons constrained within the pushback boundaries; and finally, a mixed integer linear programming mathematical model is introduced, which uses the generated pushbacks and aggregates as the planning units to provide near-optimal practical life-of-mine schedules. The production scheduling tool allows the mine planner to optimize large-scale real-size multi-pit multi-process planning problems. Also, the model inherently solves the cut-off grade optimization problem. A case-study of an iron-ore deposits presented to illustrate practicality of the developed methodologies, and also to compare the results against industrial conventional practices to assess validity, performance, and strengths of the developed methodologies.

INTRODUCTION

The first step in a mining operation is to build a block model that represents the characteristics of the deposit in a three dimensional grid. Grades and rock characteristics are estimated based on drill-hole data using Geostatistical methods. Mine planning is a process in which the order of extraction of blocks is determined. The planning process is usually performed in a top-down multi-step fashion with different resolution in each step. The first step is to determine the ultimate pit limit (UPL) based on the economic value of the blocks. The goal of this step is to determine which blocks to extract to maximize the profit while respecting slope constraints. Two widely used techniques can be found in the literature: Lerchs and Grossmann (LG) algorithm (Lerchs and Grosmann, 1965) and the moving cone algorithm (Pana, 1965). Linear programming (LP) and graph maximum closure techniques are also introduced in the literature to find the optimum pit limits but are not as commonly used as LG and the moving cone algorithm. Most of the current software and literature use the LG algorithm since it can rapidly find the ultimate pit limit. The blocks falling outside the ultimate pit limits are then removed from the planning problem to reduce the size of the problem. On the other hand, there have been attempts to determine the ultimate pit limit and the production schedule simultaneously. However, Caccetta and Hill (2003) prove that simultaneous optimization of production schedule and ultimate pit limit does not contribute to the net present value (NPV) of the operation since the optimum pit limit found based on maximizing NPV will always fall inside the predetermined ultimate pit limit.

The next step of mine production planning is to determine the phases of extraction also known as pushbacks. The traditional approach for determining the pushbacks, called parameterization approach, is to iteratively change the value of the blocks using a revenue factor and determine the ultimate pit limit based on the modified block values. The result of parameterization is a series of nested pits and the user or an automatic procedure will choose a number of pits. The incremental tonnage between two subsequent chosen pits represents a pushback. The pits determined via the parameterization approach suffer an important setback. Regardless of the step-size of changing the block value, there is a possibility that there is no appropriate selection of pushbacks such that the material tonnage is uniformly distributed between the pushbacks. Changing the block value by a very small amount can force the ultimate pit limit optimizer to find a pit that is significantly larger than the previous one. It is also possible that a large portion of the generated pits have very close tonnages such that you cannot choose the desired number of pushbacks from them. This phenomenon is called the gap problem in the literature and can affect the quality of the production schedule. We introduce a pushback design procedure in this paper that can overcome the gap problem and can determine pushbacks with controlled rock and ore tonnage.

Determining life of mine production schedule is the next step in which the order of extraction of blocks from the mine and their destination is determined. This process has been widely studied in the mining literature and various approaches have been introduced. These approaches can be classified into heuristics, meta-heuristics, dynamic programming and mixed integer linear programming (MILP). Most of these mine planning techniques try to find the best production schedule that maximizes the NPV with respect to mining and processing constraints. However, there are instances that different objective functions, such as minimizing the deviations from the target production, minimizing the total cost and minimizing the production grade deviations, are considered. For a good review on various models developed to solve the open pit production planning problem the readers can refer to Osanloo et al (2008) and Newman et al (2010)

PUSHBACK DESIGN

In this section, we introduce a new pushback design algorithm based on mathematical programming that can determine pushbacks with controlled ore and waste tonnages. The algorithm starts by forming a binary integer programming (BIP) model that assigns blocks to pushbacks such that the tonnage of material and the tonnage of ore in each pushback do not exceed predetermined values. The next step is applying reduction techniques to reduce the size of the problem. The remaining problem is then solved using common MILP solvers. However, if the size of the remaining problem is still large and imposes long CPU processing times a two-step heuristic approach is used to determine the pushback assignment.

Mathematical Formulation

In this section, we propose a binary integer programming (BIP) model which assigns blocks to pushbacks while respecting maximum ore and rock tonnage and slope constraints. The BIP model is developed to find out if block *i* is contained in pushback *j*. This model can be formed and solved iteratively to assign blocks to pushbacks. Therefore, we solve the model for the j^{th} pushback, remove the blocks determined to belong to this pushback from the block model, and solve the BIP for $j+1^{th}$ pushback. It is assumed that j^{th} pushbackis extracted prior to$j+1^{th}$ pushback. Thus, the slope constraints are respected if we remove all the blocks assigned to pushback *j* when we are assigning blocks to pushback *j+1*.This BIP has significant structural properties that can be used for finding bounds and initial solutions in reasonable time. Readers can refer to Mieth (2012) for a detailed study on the structural properties of the pushback design BIP. **Figure 1** shows the pseudo code for the push-back design.

$PB^0 = \emptyset$, i=1, set capacity, set C^u, set block_data

while $\bigcup_{j=0}^{i-1} PB^j \neq C^u$ **do**

    $[PB^i]$ = solve_PB(capacity, block_data)

    block_data = remove_PB(block_data, PB^i)

    i = i + 1

end while

end

Figure 1 – Pseudo code for pushback creation (Mieth, 2012).

<u>Sets</u>

N	set of all nodes in the precedence graph representing all the blocks in the block model
E	set of all edges in the precedence graph
PB^j	set of blocks in pushback *j*
C^u	set of blocks in the ultimate pit

Parameters

p_i	economic value of block *i*
t_i	overall tonnage of block *i*
o_i	ore tonnage of block *i*
mc_j	maximum tonnage of material in pushback *j*
pc_j	maximum ore tonnage in pushback *j*

Decision variables

x_{ij}	binary integer variable indication if block *i* belongs to pushback *j*

Mathematical formulation

$$\max \sum_{i \in N} p_i x_{ij} \tag{1}$$

s.t.

$$x_{ij} - x_{kj} \leq 0 \;\; \forall (i,k) \in E \tag{2}$$

$$\sum_{i \in N} t_i x_{ij} \leq mc_j \tag{3}$$

$$\sum_{i \in N} o_i x_{ij} \leq pc_j \tag{4}$$

$$x_{ij} \in \{0,1\} \;\; \forall i \in N \tag{5}$$

The objective function of the mathematical model is to maximize the profit by including the blocks with the highest value in the pushback (Equation (1)). Since there is an assumption that j^{th}pushback is extracted prior to $j+1^{th}$ pushback assigning higher value blocks to pushback *j* results in higher NPV. Equation (2) enforces the slope precedence constraint i.e. a block is allowed to be included in pushback *j* if all of its predecessors are also included in this pushback. Note that some predecessors of block *i* could have been assigned to pushback *j-1* and their corresponding constraints are not included in the j^{th} instance of the formulation. Equations (3) and (4) enforce the maximum rock and ore tonnage of material contained in the pushback. Equation (5) indicates that x_{ij} variables can only take 0 or 1.

Problem Reduction

The BIP formulation we present is similar to a very famous operations research problem called the Knapsack Problem (KP). KP is originally defined as the decision of choosing the items to carry in a knapsack to maximize the total utilization. Each item has a volume and the total volume of the Knapsack is limited to a specific number. Although KP does not account for precedence between the objects (Equation (2)), an extension to KP called the Precedence Constrained Knapsack Problem (PCKP) considers the same type of precedence between the objects. KP, PCKP and other extensions of KP are widely studied in the literature and proven to be at least NP-Hard (Johnson and Niemi, 1983). Therefore, there is a need to come up with procedures to reduce the size of the problem and to use heuristics to get near-optimal solutions to our BIP. The structural properties of the aforementioned BIP and the reduction procedures are explained in Mieth (2012) in details but briefly explained in this section.

The first step of problem reduction is to eliminate all the blocks falling outside the ultimate pit limits. In our approach, ultimate pit limit is determined using a maximum closure technique and all the blocks not chosen to be included in ultimate pit limit are removed from the set and not considered in the consequent steps of the procedure.

In the next step we use a combination of linear relaxation and Lagrangean relaxation to find the upper and lower limits of the pushback problem using the hybrid method introduced in Bienstock and Zuckerberg (2010). The two bounds calculated serve not only as the bound for the pushback value but also in a spatial context. Therefore, the blocks inside the lower bound and the blocks outside the upper bound can be discarded from the problem.

Exact Solution

If the residual problem is small enough, it is solved using a general LP solver (we used CPLEX (IBM, 2010)). The variables corresponding to the blocks inside the lower bound are prefixed to 1 since they are certainly included in the pushback. The variables corresponding to the blocks outside the upper bound are totally discarded in order to improve the memory usage of the program. The residual problem is then sent to solver to get the optimal solution to the pushbacks design problem. If the processing time exceeds a predetermined value the exact solution procedure is terminated and a greedy heuristic along with a local search algorithm become responsible for determining the blocks to be included in the pushback.

Greedy Heuristic

Greedy Heuristic is a famous heuristic for solving the Knapsack problem. The idea behind this algorithm is to calculate a cost-benefit ratio for all the items and choose the items with highest values to be included in the Knapsack. In our case, the value of the blocks divided by their tonnage is a quality measure for choosing the best blocks to include. However, since there are precedence constraints in the model, the value of a block is calculated as its own value plus the values of all the blocks that have to be extracted prior to extracting that block. The same calculation is done for the tonnages. Consequently, the block quality if defined as the total value of the block and all its predecessors divided by the total tonnage that has to be extracted prior to extracting that block. As its name implies, the Greedy Heuristic chooses the blocks in a greedy manner that does not always result in the optimum solution. Thus, a local search step is added to improve the solution determined using the Greedy Heuristic.

Local Search

If the value of the generated solution with greedy heuristic is close to the upper bound calculated the solution is accepted. Otherwise, a local search heuristic is used to improve the quality of the solution in a few iterations. Let us define S^i as the set of blocks included in the pushback in the i^{th} iteration and V^i as the set of blocks with positive value not included in S^i. In every step of the local search, the distance between the blocks from V^i and the closest block to them from S^i is calculated. A neighborhood BIP with the blocks from V^i closest to S^i and their predecessor blocks is formed to respect the slope constraints. The BIP is then solved using exact algorithms and the local search continues until it reaches the maximum number of iterations. For a detailed explanation of the local search algorithm used the readers are referred to Mieth (2012). The pushbacks generated at the end of this step are then used in clustering and also forming the MILP as described in the following sections.

CLUSTERING

Clustering is a general term for a group of algorithms that group similar objects together. This grouping is usually done via defining a similarity or dissimilarity measure. The method we use in this project is called hierarchical clustering. As its name implies, hierarchical clustering forms a hierarchy of objects based on their similarity. The similarity index and the way our algorithm works is thoroughly explained in Tabesh and Askari-Nasab (2011). However, a shape refinement procedure is added to the hierarchical clustering algorithm to produce clusters with better shapes and better controlled sizes. This procedure is run after the blocks are clustered in every bench and consists of disaggregating small clusters and also removing sharp corner from the clusters. On the other hand, it has to be considered that the refinement procedure does not account for similarity and consequently over-smoothing the clusters results in decreasing the overall intra-cluster similarity.

PRODUCTION PLANNING MILP

In order to determine the long-term production plan of a mine, an MILP is formed which uses the intersections of the designed pushbacks and the benches as mining units. The mining units are called panels or bench-phases. The processing decisions are made based on the mining cuts formed using hierarchical clustering. The cuts are formed in a way that they mostly consist of one rock type. The MILP tries to maximize the NPV of the mining operation while respecting mining and processing capacity constraints, slope constraints and plant head grades of metal and deleterious material. The MILP is based on the one proposed in Tabesh and Askari-Nasab (2011) with a modification of the definition of decision variables.

$$\max \sum_{t=1}^{T}\left(\sum_{k=1}^{K}(v_k^t \times x_k^t) - \sum_{p=1}^{P}(q_p^t \times y_p^t)\right) \tag{6}$$

$$gl^{t,e} \le \sum_{k=1}^{K} g_k^e \times o_k \times x_k^t \Big/ \sum_{k=1}^{K} o_k \times x_k^t \le gu^{t,e} \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{7}$$

$$pl^t \le \sum_{k=1}^{K} o_k \times x_k^t \le pu^t \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{8}$$

$$ml^t \le \sum_{p=1}^{P} (o_p + w_p) \times y_p^t \le mu^t \qquad \forall t \in \{1,...,T\} \tag{9}$$

$$\sum_{k \in K_p} o_k \times x_k^t \le \left(o_p + w_p\right) \times y_p^t \qquad \forall p \in \{1,...,P\}, \quad t \in \{1,...,T\} \tag{10}$$

$$b_p^t - \sum_{i=1}^{t} y_s^i \le 0 \qquad \forall p \in \{1,...,P\}, \quad t \in \{1,...,T\}, \quad s \in C_p \tag{11}$$

$$\sum_{i=1}^{t} y_p^i - b_p^t \le 0 \qquad \forall p \in \{1,...,P\}, \quad t \in \{1,...,T\} \tag{12}$$

$$b_p^t - b_p^{t+1} \le 0 \qquad \forall p \in \{1,...,P\}, \quad t \in \{1,...,T-1\} \tag{13}$$

Where

- $x_k^t \in [0,1]$ is a continuous variable, representing the portion of mining-cut k to be extracted as ore and processed in period t.
- $y_p^t \in [0,1]$ is a continuous variable, representing the portion of panel p to be mined in period t, fraction of y characterizes both ore and waste included in the panel.
- $b_p^t \in \{0,1\}$ is a binary integer variable controlling the precedence of extraction of panels. b_p^t is equal to one if extraction of panel p has started by or in period t, otherwise it is zero.
- C_p is the set of the panels that have to be extracted prior to panel p
- K_p is the set of the mining-cuts within panel p
- v_k^t is the discounted revenue generated by selling the final product within mining-cut k in period t minus the extra discounted cost of mining all the material in mining-cut k as ore and processing it.
- q_p^t is the discounted cost of mining all the material in panel p as waste.
- o_k is the ore tonnage in mining-cut k.
- w_p is the waste tonnage in panel p.

- g_k^e is the average grade of element e in ore portion of mining-cut k.
- $gu^{e,t}$ and $gl^{e,t}$ are the upper bound and lower bound on acceptable average head grade of element e in period t in percent.
- pu^t and pl^t are the upper bound and lower bound on processing capacity of ore in period t in tonnes.
- mu^t and ml^t are the upper bound and lower bound on mining capacity in period t in tonnes.

Equation (6) is the objective function that consists of two parts. The first part is the discounted revenue generated from processing the cuts in the processing plant. The second part of the objective function is the cost associated with extracting material from panels. The relation between the ore tonnage extracted from the cuts and the total tonnage extracted from the corresponding panels is modeled in equation (10). Obviously, the total tonnage processed from the cuts in each period cannot exceed the tonnage extracted from their panels in the same period. Equations (8)and (9) are the mining and processing capacity constraints. Equation (7) controls the head grade of the material sent to the processing plants for different elements of interest. Equations (11) to (13) are responsible for slope constraints between the panels.

Case Study

A case study from an iron ore mine with approximately 177,000 blocks in the ultimate pit limits used to illustrate the performance of our multi-step production planning procedure. The ultimate pit limit contains approximately 4 billion tonnes of material with nearly 700 thousand tones of mineralized material. The distribution of rock types are shown from a plan view and two cross sections in **Figure 2** to **Figure 4**. Ore rocks are shown with darker shade. The ore body is tabular and goes deep (**Figure 4**) which makes it very hard for parameterization approach to find appropriate pushbacks.

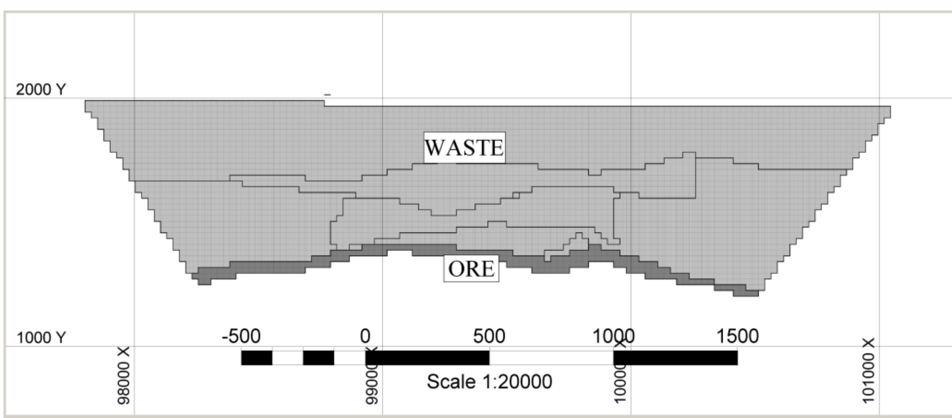

Figure 2 – Sample vertical section looking North at 600140 Northing.

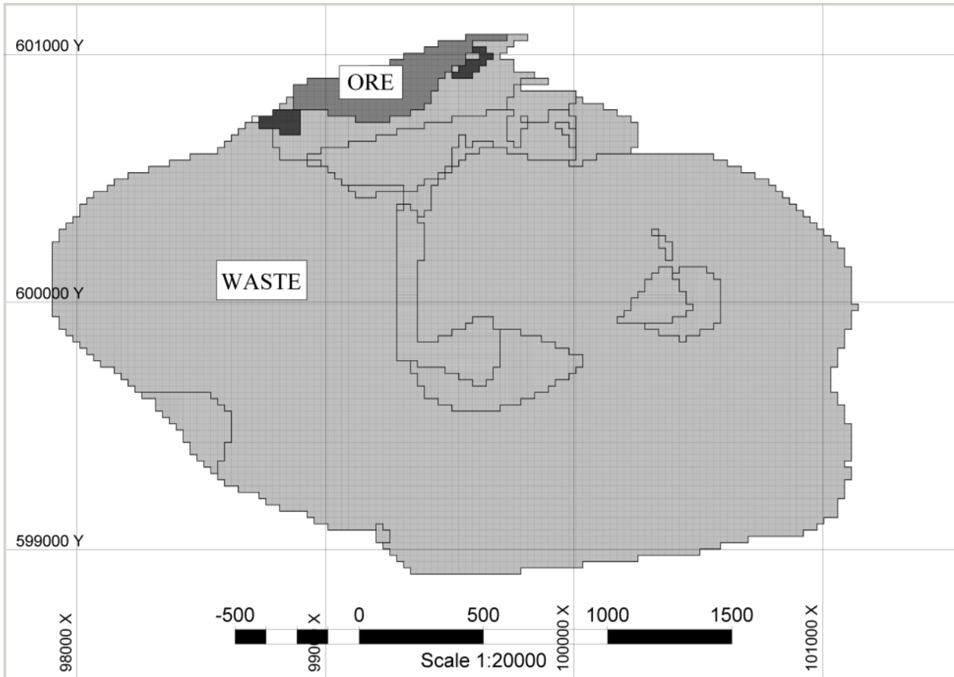

Figure 3 – Sample plan view at 1580 meters elevation.

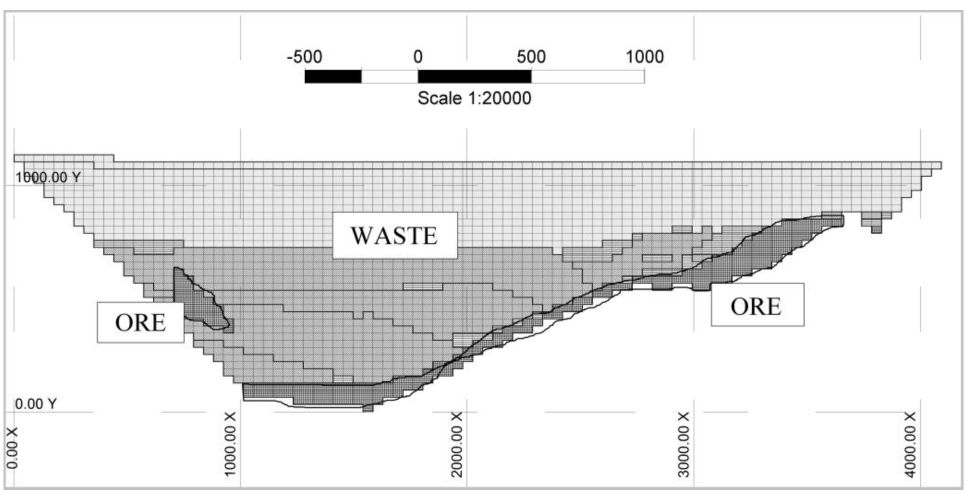

Figure 4 – Sample vertical section looking West at 98100 meters Easting.

The block model is first divided into a number of pushbacks using the pushback design procedure. The panels are then formed as the intersection of pushbacks and benches and are used as the extraction units. The generated pushbacks have more reasonable shapes, compared to the results of the parameterization approach, and are uniform in their tonnages. **Figure 5** shows the comparison of tonnages when we aim at five pushbacks using our algorithm and the parameterization approach. **Figure 6** compares the results for eight pushbacks.

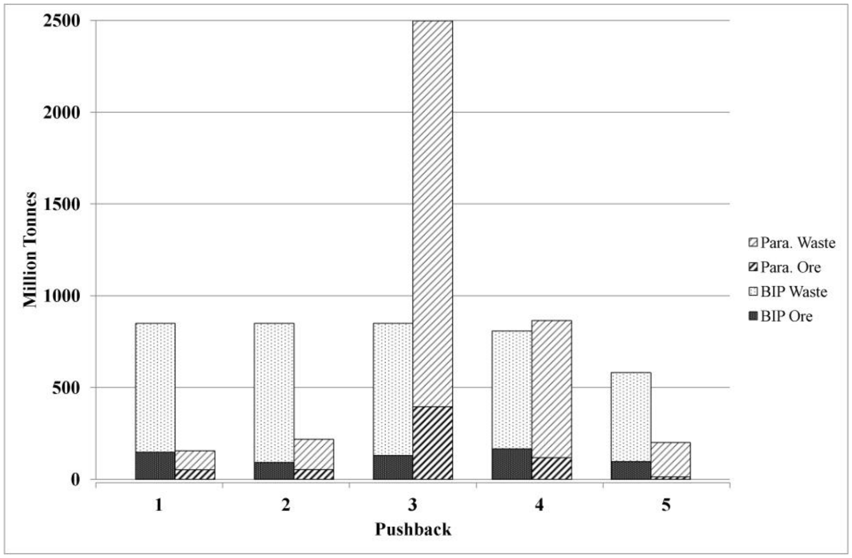

Figure 5 – Comparison of 5 pushbacks.

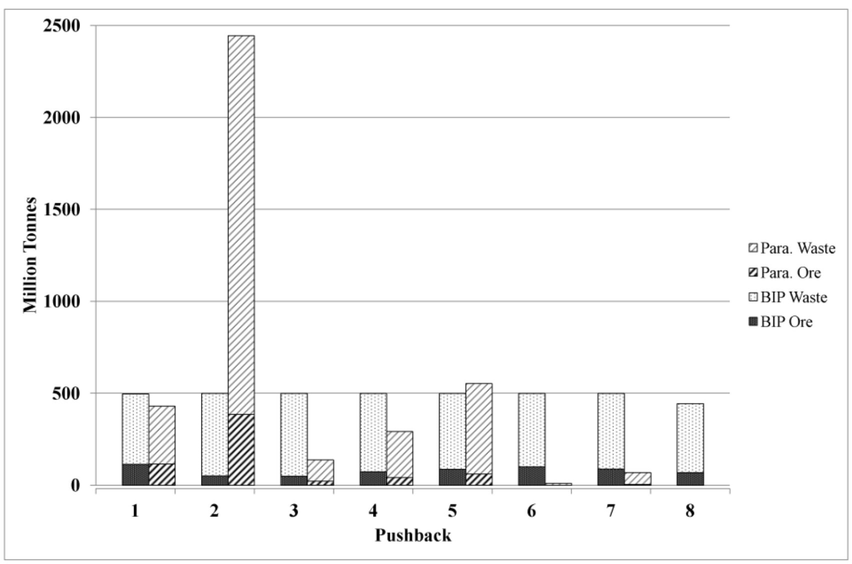

Figure 6 – Comparison of 8 pushbacks.

The shapes of the generated pushbacks are also important from the mining operations point of view. Very narrow pushbacks are not favorable since they practically cannot be mined. On the other hand, very large pushbacks result in large panels and prevent the schedule from getting to the ore faster.

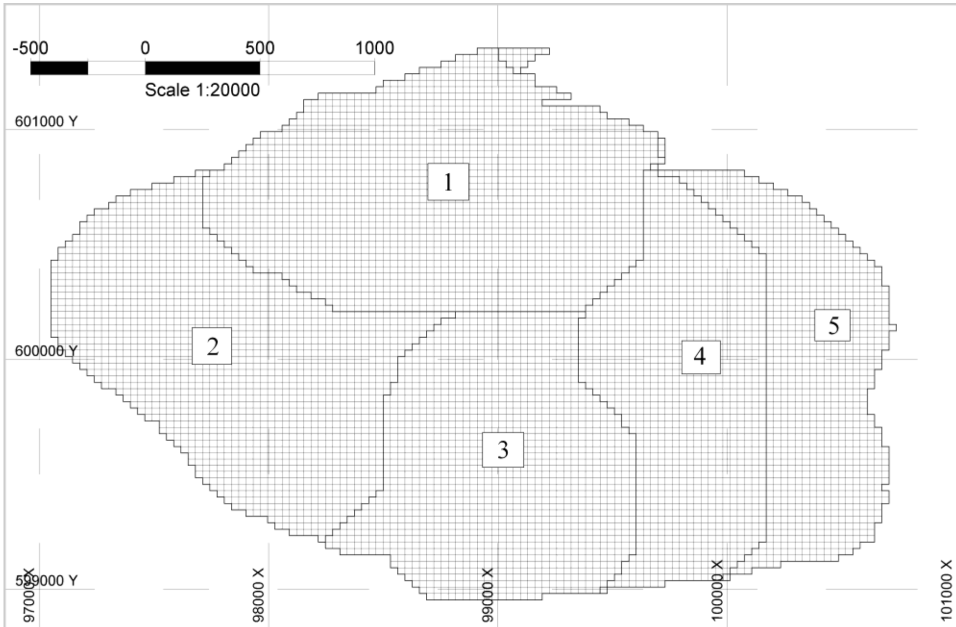

Figure 7 – Sample plan view from the BIP/Heuristic pushbacks.

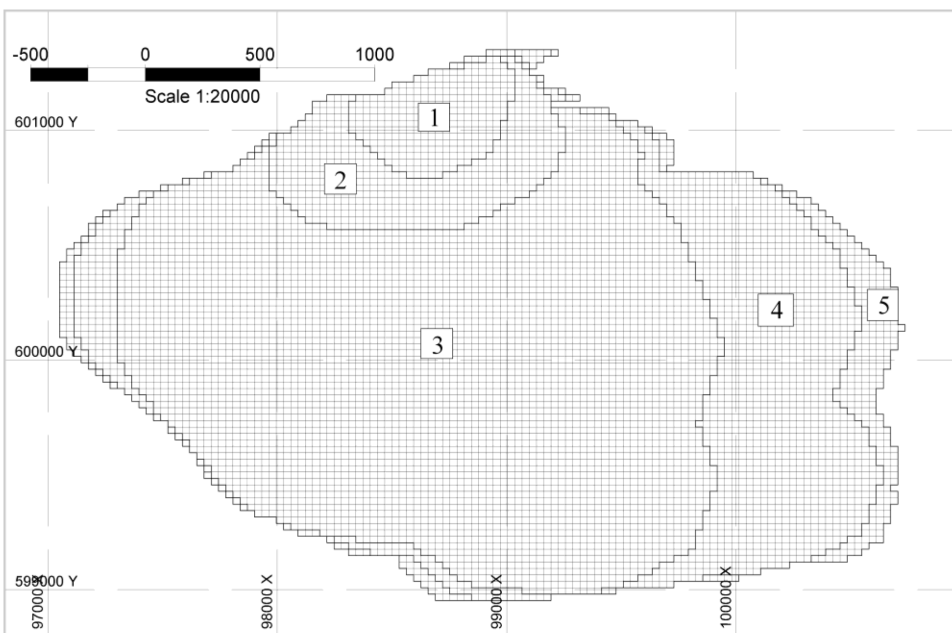

Figure 8 – Sample plan view from the parameterisation pushbacks.

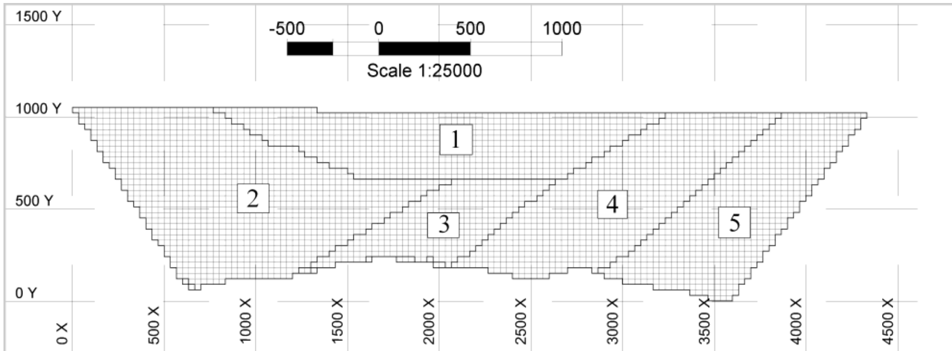

Figure 9 – Sample vertical section from the BIP/Heuristic pushbacks.

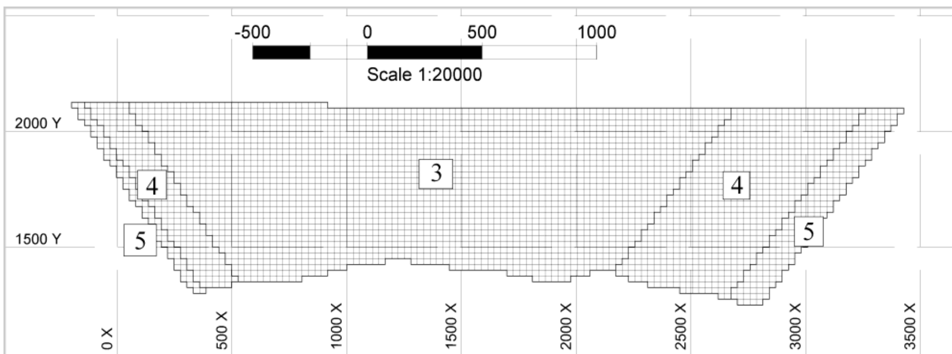

Figure 10 – Sample vertical section from the parameterisation pushbacks.

Afterwards, the hierarchical clustering algorithm is used to form mining-cuts which are the basis for making ore and waste decisions. The determined pushbacks have to be taken into account during the clustering algorithm so that we do not end up with clusters overlapping two or more panels. We call this feature clustering within pushback boundaries. An MILP is then formulated to determine the optimum life-of-mine production plan using the generated cuts and panels. We set the mining capacity to 120 MT/yr for the first 9 years, 150 MT/yr from year 10 to 20 and 100 MT/yr from year 20 to 35. The processing capacity is 10 MT/yr from year 4 to 14 and 20 MT/yr from year 15 to 35. We ran the solver to 2% optimality gap. The results are summarized in **Table 1**. The highest NPV belongs to the MILP based on BIP/Heuristic pushbacks which is $6.34 billion. However, having 5 pushbacks results in 0.5% smaller NPV but offers a time saving of 82%. MILPs based on 8 and 5 pushbacks from the parameterization approach have lower NPVs and do not contribute to the processing times. We used the same parameter settings in Whittle (Gemcom, 2012) and used the Milawa algorithms to get the production schedule. The result is shown in **Table 2**. It can be seen that the MILP results in significantly higher NPV.

Table 1 – MILP Summary.

Pushback Algorithm	No. of Pushbacks	No. of Panels	No. of Cuts	NPV ($B)	MILP CPU Times (s)
BIP/Heuristic	5	173	3,764	6.31	383
	8	263	3,642	6.34	2,161
Parameterization	5	151	4,605	5.83	303
	8	244	10,555	6.29	2,314

Table 2 – Whittle production scheduling results.

Pushback Algorithm	Scheduling Algorithm	NPV ($B)	
		5 Pushbacks	8 Pushbacks
BIP/Heuristic	MILP	6.31	6.34
Parameterization		5.83	6.29
Parameterization	Milawa NPV	1.85	4.73
Parameterization	Milawa Balanced	1.85	2.86

CONCLUSION

We present a pushback design algorithm based on binary integer programming, greedy heuristic and a local search. The results from our pushback design procedure are more uniform in tonnage and more practical in shape from the operational point of view compared to pushbacks generated using parameterization approach. We used intersection of pushbacks and benches as our extraction units. We also used a clustering algorithm to form groups of blocks with similar rock types as the units for making ore and waste decisions. We then developed a life-of-mine production scheduling MILP which uses the bench-phases and mining-cuts as the production units and gets near-optimal production schedules in reasonable time. The performance of our multi-step approach is tested on an iron ore deposit with 177,000 blocks in the final pit. The comparison shows significant improvement against Whittle (Gemcom, 2012) regarding the NPV and the generated schedule.

REFERENCES

Bienstock, D. and Zuckerberg, M. (2010). Solving LP Relaxations of Large-Scale Precedence Constrained Problems. Integer Programming and Combinatorial Optimization SE - 1. F. Eisenbrand and F. B. Shepherd. Springer Berlin Heidelber. pp. 1-14.

Caccetta, L. and Hill, S. P. (2003). An Application of Branch and Cut to Open Pit Mine Scheduling. Journal of Global Optimization. v. 27(2-3). pp. 349-365.

Gemcom Software International (2012). Whittle strategic mine planning software. Version 4.4

IBM, C. (2010). CPLEX. New York. Armonk.

Johnson, D. S. and Niemi, K. A. D. (1983). On Knapsacks, Partitions, and a New Dynamic Programming Technique for Trees. Mathematics of Operations Research. v. 8(1) pp. 1-14.

Lerchs, H. and Grosmann, L. (1965). Optimum Design of Open Pit Mines. CIM Transaction. v. 68. pp. 17-24.

Mieth, C. (2012). Pushback-design using mixed integer linear optimization. Industrial Engineering. Freiberg. TU Bergakademie Freiberg. 111p.

Newman, A. M., Rubio, E., Caro, R., Eurek, K. (2010). A Review of Operations Research in Mine Planning. Interfaces. v. 40(3). pp. 222-245.

Osanloo, M., Gholamnejad, J., Karimi, B. (2008). Long-term open pit mine production planning: a review of models and algorithms. International Journal of Mining, Reclamation and Environment. v. 22(1). pp. 3 - 35.

Pana, M. T. (1965). The Simulation Approach to Open-Pit Design. Transactions of the Short Course and Symposium on Computers and Computer Applications in Mining and Exploration, Tucson, Arizona. University of Arizona.

Tabesh, M. and Askari-Nasab, H. (2011). Two-stage clustering algorithm for block aggregation in open pit mines. Transactions of the Institutions of Mining and Metallurgy, Section A: Mining Technology. v. 120(3). pp. 158-169.