A Hybrid Solution Methodology to Long-Term Open Pit Production Planning

Mohammad Tabesh, Clemens Mieth, and Hooman Askari-Nasab
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

The life-of-mine production schedule determines the order of extraction of materials and their destination over the mine-life. The objective of this paper is to develop, verify, and present a multi-step methodology for three interrelated key components of open-pit mine planning: controlled optimal phase-design, characterization of selective mining-units, and long-term production scheduling optimization. A hybrid solution methodology for open-pit phase-design using integer programming and a local search heuristic is presented. Next, a hierarchical clustering approach with size and shape control, which aggregates blocks into minable polygons constrained within the pushback boundaries, is presented; and finally, a mixed integer linear programming mathematical model, which uses the generated pushbacks and aggregates as the planning units to provide near-optimal practical life-of-mine schedules, is introduced. The production scheduling tool allows the mine planner to optimize large-scale real-size multi-pit multi-process planning problems, where the planner has control over mining-fleet capacity, production capacity and grade blending requirements. In addition, the model inherently solves the cut-off grade optimization problem. Two case-studies of real-size deposits are presented to illustrate practicality of the developed methodologies, and also to compare the results against industrial conventional practices to assess validity, performance, strengths, and limitations of the developed methodologies.

1. Introduction

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 approach with different resolution of details required in each planning step. The process consists of determining pit limits, designing pushbacks, determining the optimal cut-off grades and scheduling the mining and production processes. Various studies have tackled each of these steps as individual problems. However, the objective of this paper is to propose a multi-step approach that benefits from the relations between the steps and finds near-optimal solutions using mathematical programming.

The first step in mine planning procedure is to determine the ultimate pit limit based on the economic value of the blocks. The goal of this step is to determine which blocks to extract in order to maximize the profit while respecting slope constraints. Two widely used techniques can be found in the literature: the Lerchs and Grossmann (LG) algorithm (Lerchs and Grosmann, 1965)

Askari-Nasab, Hooman (2013), Mining Optimization Laboratory (MOL) – Report Five, © MOL, University of Alberta, Edmonton, Canada, Pages 230, ISBN: 978-1-55195-327-4, pp. 1-23.

This paper has been submitted to International Journal of Mining and Mineral Engineering (IJMME).

and the moving cone algorithm (Pana, 1965). The LG algorithm is a graph theory based algorithm in which weights are assigned to graph nodes based on block values and the maximum weight of the graph is found. On the other hand, the moving cone algorithm is a simulation-based heuristic that iteratively moves a cone with respect to slope constraints on the block model in order to find the optimum pit limits. 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 the LG and moving cone algorithms. Most of the current software and literature use the LG algorithm since it can find the optimum ultimate pit limit in a reasonable time. 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 the production schedule and the 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 in mine production planning is to determine the phases of extraction, also known as pushbacks. The common approach for determining the pushbacks, called the parameterization approach, is to iteratively change the value of the blocks using a revenue factor and to determine the ultimate pit limit based on the modified block values. The result of parameterization is a series of nested pits, of which the user or an automatic procedure must choose a certain number. The incremental tonnage between two chosen nested pits represents a pushback. The pits determined via the parameterization approach face an important limitation. Regardless of the step-size chosen, 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 finder to find a pit that is significantly larger than the previous one. It is also possible that a large number of the generated pits have very close tonnages in a way that one cannot choose the desired number of pushbacks from them. This phenomenon, which can affect the quality of the production schedule, is called the "gap problem" in the literature. In Section 2 of this paper, we introduce a pushback design procedure that can overcome the gap problem and can determine pushbacks with controlled rock and ore tonnage.

Determining the 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. Various methods have been introduced: heuristics, meta-heuristics, dynamic programming (DP) and mixed integer linear programming (MILP). Although heuristic and meta-heuristic methods are widely used in commercial software, their lack of generality and their inability to incorporate problem specific features have motivated researchers to look for more flexible solution procedures. Samanta, Bhattacherjee, & Ganguli (2005) and Zhang (2006) grouped blocks into larger units and found a production schedule using the Genetic algorithm. However, they were not able to measure the quality of their schedule and did not provide any details about their aggregation techniques.

The first and most famous mathematical model for representing open pit production planning was introduced by Johnson (1969). The proposed model considered many constraints and possibilities but was too large to be solved directly. Researchers in the mine planning area have since proposed various mathematical programming approaches. For a good review of the history of operations research in mine planning, the readers can refer to Osanloo, Gholamnejad, & Karimi (2008) and Newman, Rubio, Caro, Weintraub, & Eurek (2010). Among the numerous researchers who worked on the open pit production planning problem, some have used aggregates as the planning units to reduce the problem size. Gershon (1983) labeled layers of material as ore or waste and used zero-one decision variables to determine the extraction period for each layer. Klingman & Phillips (1988) followed the same approach but let the mathematical formulation decide whether a layer is economical to process. This layering technique is limited to low-depth layered deposits such as Phosphate and Limestone. Ramazan (2007) defined the concept of fundamental trees as mineable groups of blocks in order to solve the MILP with smaller number of variables without sacrificing optimality. Boland, Dumitrescu, Froyland, & Gleixner (2009) used aggregated units to solve the

MILP and proposed a disaggregation technique to find the block level schedule based on the aggregated schedule. However, they did not propose any aggregation techniques. Askari-Nasab & Awuah-Offei (2010) and Tabesh & Askari-Nasab (2011) used hierarchical clustering to form aggregates of blocks as planning units. Badiozamani & Askari-Nasab (2012) and Ben-Awuah & Askari-Nasab (2012) used the same clustering approach to deal with the production planning problem in oil sands mines.

DP is also favored due to its easy implementation and flexibility. However, DP approaches are not usually able to handle large-scale problems due to the number of iterations they need. Another shortcoming shared by heuristic and DP methods is the fact that a user does not own a tool to estimate how far from the optimal solution he has gotten. Therefore, most researchers tend to use MILP to model the production planning problem. MILP models can easily incorporate new constraints imposed by technical and economic conditions of a mining operation and can provide the user with a measure to evaluate the quality of the solution they get at each step. However, MILP models face the limitation that they need huge amounts of time and resources to be solved. Note that in its simplest form, the open pit production planning MILP model can be reduced to Precedence-Constrained Knapsack Problem which has been proven to be NP-Hard (Caccetta and Kulanoot, 2001).

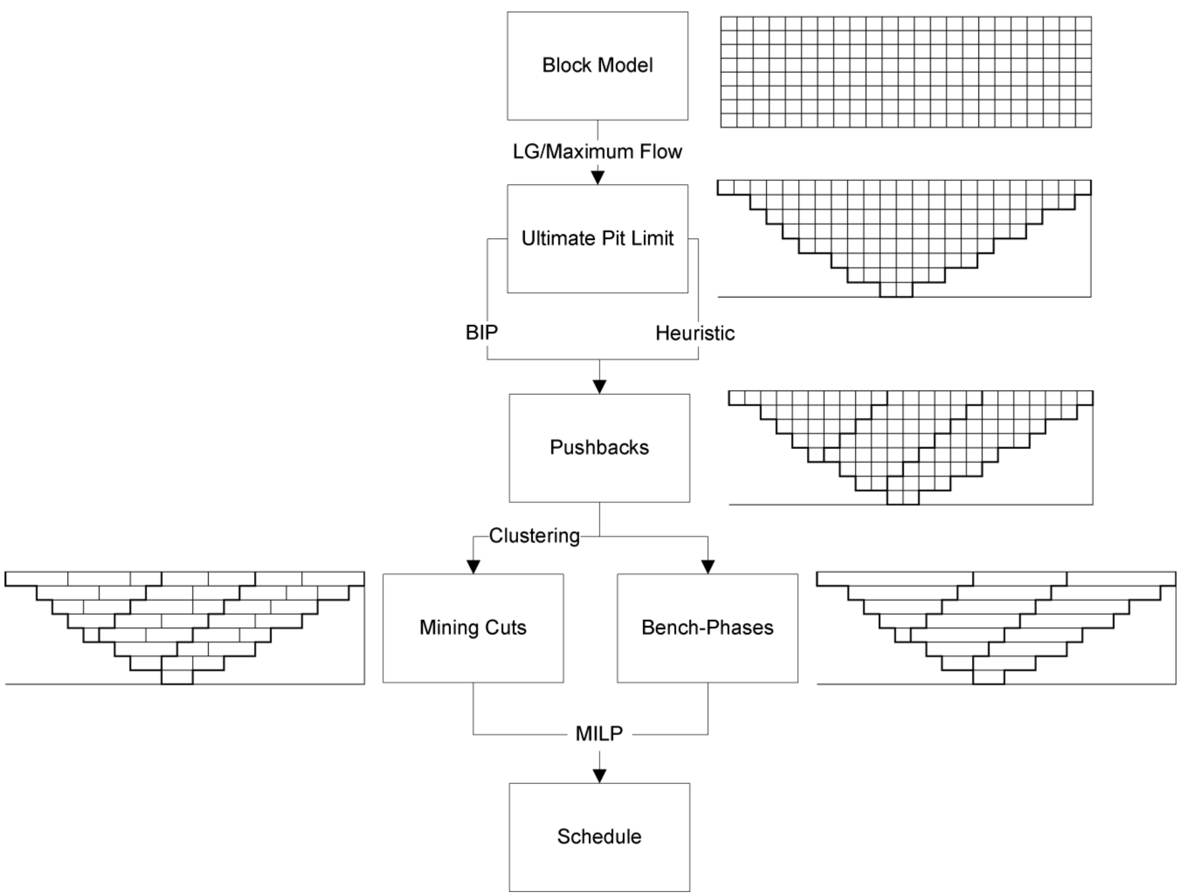

Figure 1. Mine Planning Flowchart

Although the open pit production planning problem has been studied for a long time, there is no globally recognized solution approach that can solve the problem with a reliable solution in a reasonable time. Mathematical programming techniques are time consuming while heuristics and meta-heuristics are not flexible to case-specific technical and financial characteristics and do not ensure the user of the quality of the solution they propose. On the other hand, even if it were computationally possible to determine the mine plan at the block level without breaking it down, the solution would be a scattered impractical schedule that is not favorable from the operations' point of view.

Therefore, in this paper, we are breaking down the production planning problem into multiple steps, as illustrated in Figure 1 in order to deal with smaller, more tractable problems. We will be using two different resolutions of aggregates to form the MILP in order to provide practical schedules. The first set contains the panels (bench-phases), which are the intersections of the benches and the pushbacks determined in Section 2. These panels are used to make extraction decisions. We then use a modified hierarchical clustering approach to form smaller units within those panels to make processing decisions in the MILP. The clustering algorithm is briefly explained in Section 3. For a more detailed presentation of the clustering algorithm, the readers are referred to Tabesh and Askari-Nasab (2011) and (2012). The mathematical formulation is presented in Section 4. Case studies of the application of the model are presented in Section 5. The conclusion and future works are presented in Section 6.

2. Pushback design using integer programming

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 common approach is to use parameterization for finding pushbacks. In order to find the pushbacks, a number of nested pits are determined using the Lerchs and Grosmann (1965) (LG) algorithm. In this approach, the value of the blocks is iteratively changed using a revenue factor. The ultimate pit limit is then found using the LG algorithm and based on the modified block values. Starting from a small revenue factor, e.g. 0.1, the smallest nested pit is formed. Consequent larger pits are determined in the same way using larger revenue factors. This process is continued until the revenue factor reaches the maximum revenue factor defined or all the blocks with positive values are contained in the final pit. A number of pushbacks are then chosen, from the nested pits generated, by the user or an automatic algorithm.

The pushbacks determined using parameterization share the following advantages: they all respect slope constraints and they are nested pits i.e., every pushback is contained within the next larger pushback. However, the parameterization approach faces an important limitation: the size of the pushbacks is not directly relative to the step size used in parameterization of the revenue. Consequently, regardless of the step size, it is possible to end up having pushbacks that are either very close in size or have a significant difference in the tonnage of ore and waste they contain. This phenomenon, which is referred to as the "gap problem" in the literature, can cause limitations for the next steps of planning. As mentioned earlier, we are using two sets of aggregates as planning units. The first set consists of the intersections of the benches and the pushbacks, called panels, and the second set consists of clusters of blocks within those panels. Large panels, because of large pushbacks, can decrease the NPV since they prevent the schedule from getting to the bottom of the pit, where blocks with higher value can be found, in early years. On the other hand, very small panels are not practical to be mined as independent mining units. Therefore, there exists a need to have pushbacks of controlled size.

In order to overcome the gap problem in the parameterization approach, we developed a pushback design algorithm that consists of a mathematical model as well as two variable reduction techniques. 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 processing times, a two-step heuristic approach is used to determine the pushback assignment.

2.1. Mathematical formulation

In this section, we propose a binary integer programming (BIP) model that assigns blocks to pushbacks while respecting maximum ore and rock tonnage and slope constraints. The BIP model is developed to determine 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} pushback is 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 $j+1$. The iterative solving process is illustrated in Figure 2. This BIP has significant structural properties that can be used for finding bounds and initial solutions in a reasonable time. Readers can refer to Mieth (2012) for a detailed study on the structural properties of the pushback design BIP.

```
PB^0 = ∅ , i=1, set capacity, set C^u, set block_data
while ∪_{j=0}^{i-1} PB^j ≠ 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 2. Pseudo code for pushback creation (Mieth, 2012)

2.1.1. Sets

N	set of all nodes in the precedence graph representing all the blocks in the block mode
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

2.1.2. Parameters

p_i	economic block 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

2.1.3. Decision variables

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

2.1.4. Mathematical formulation

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

s.t.

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

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

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

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

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.

2.2. Problem reduction

The BIP formulation we present is similar to a very famous operations research problem called the Knapsack Problem (KP)(Kellerer et al., 2004). 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 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 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 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 the ultimate pit limit are removed from the set and not considered in the subsequent 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.

2.3. Exact solution

If the residual problem is small enough, it is solved using a general LP solver (we used CPLEX (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 all 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. However, if the number of blocks in the residual problem is smaller than a predetermined value (35,000 in our case) a greedy heuristic along with a local search algorithm become responsible for determining the blocks to be included in the pushback. Figure 3 illustrates how the pushback design algorithm chooses between exact and heuristic solution techniques.

2.4. Greedy heuristic

Greedy heuristic is a famous heuristic for solving the Knapsack problem (Kellerer et al., 2004). 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 seems to be 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 methods.

2.5. Local search

If the value of the solution generated with greedy heuristic is close to the upper bound calculated (1% gap) the solution is accepted. Otherwise, a local search heuristic (Aarts and Lenstra, 2003) is used to improve the quality of the solution in 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 block closest 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 forming the MILP as described in the following chapters.

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. However, we use clustering in a slightly different way. The most common objectives of using clustering techniques are to maximize intra-cluster similarity and/or to maximize the inter-cluster dissimilarity without any constraints on the number of objects in a cluster or on spatial and geographical shape of the cluster. However, we need clusters with controlled size and shape because they are being used as the units of production planning in further steps. Therefore, we evaluate the quality of our clusters by looking into their intra-cluster similarity measures (grade, rock type and destination variation) as well as their shape and size measures.

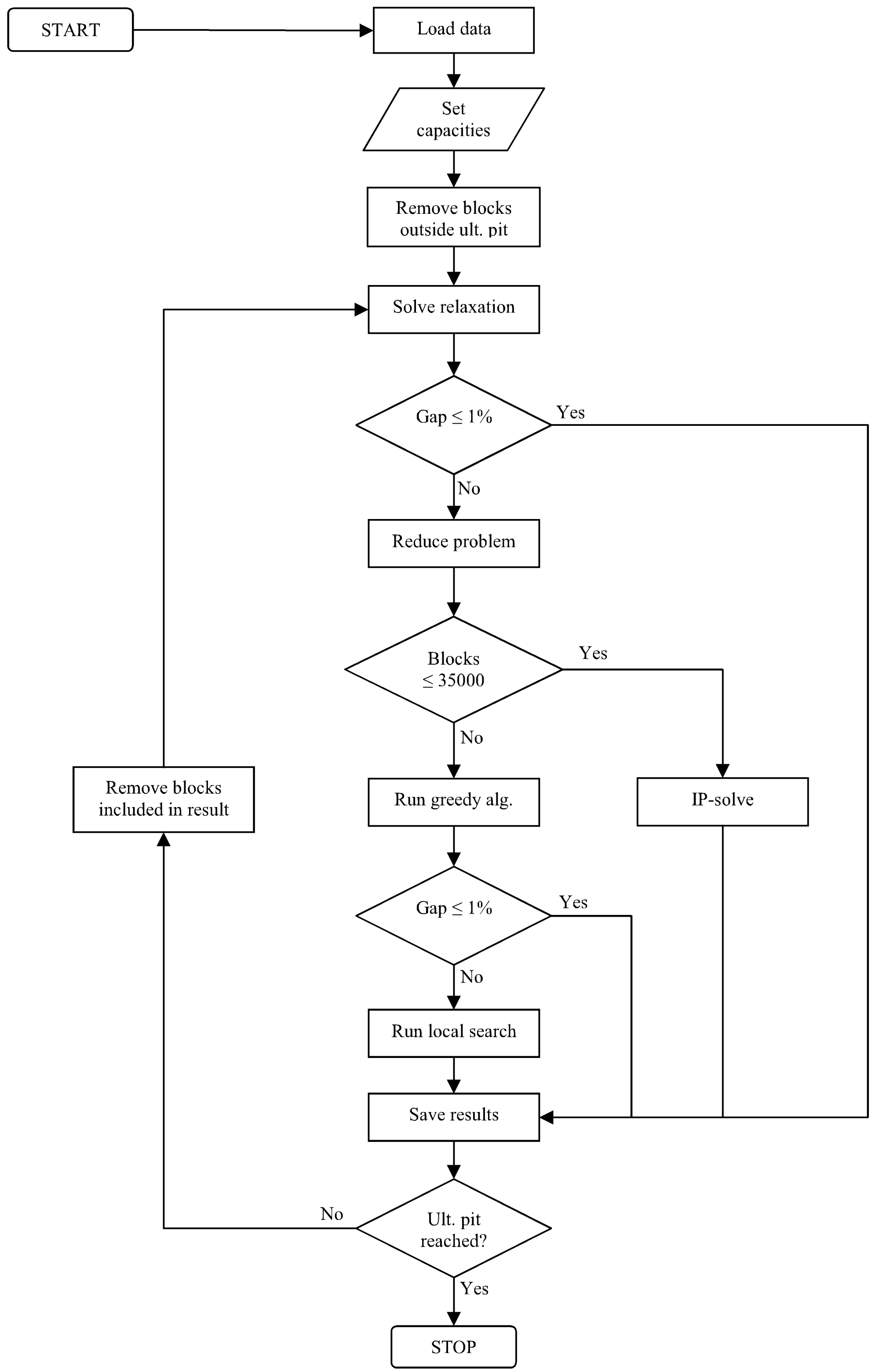

Figure 3. Pushback Program Flowchart (Mieth, 2012).

3. 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. However, we use clustering in a slightly different way. The most common objectives of using clustering techniques are to maximize intra-cluster similarity and/or to maximize the inter-cluster dissimilarity without any constraints on the number of objects in a cluster or on spatial and geographical shape of the cluster. However, we need clusters with controlled size and shape because they are being used as the units of production planning in further steps. Therefore, we evaluate the quality of our clusters by looking into their intra-cluster similarity measures (grade, rock type and destination variation) as well as their shape and size measures.

The method we use in this paper is 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 (Tabesh and Askari-Nasab, 2012). This procedure is run after the blocks are clustered in every bench and consists of disaggregating small clusters and removing sharp corners from the clusters. The procedure has two control parameters that are the number of iterations and the minimum number of blocks per cluster. Since the procedure is done iteratively, the number of iterations has to be determined by the user. The more iterations used, the smoother the clusters become. 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.

4. Production planning MILP

In order to determine the long-term production plan of a mine, an MILP is formulated 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 goal is to maximize the NPV of the mining operation while respecting mining and processing capacity constraints, slope constraints and processing plant head-grades of metal and deleterious material.

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

$$ml^t \le \sum_{p=1}^{P} (o_p + w_p) \times y_p^t \le mu^t \qquad \forall t \in \{1,...,T\} \tag{7}$$

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

$$\sum_{k \in K_p} o_k \times x_k^t \le \left(o_p + w_p\right) \times y_p^t \qquad \forall p \in \{1,...,P\}, \quad t \in \{1,...,T\} \tag{9}$$

$$0 \le \sum_{k=1}^{K} \left(g_k^e - gl^{t,e}\right) \times o_k \times x_k^t \quad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{10}$$

$$\sum_{k=1}^{K} \left(g_k^e - gu^{t,e}\right) \times o_k \times x_k^t \le 0 \qquad \forall t \in \{1,...,T\}, \quad e \in \{1,...,E\} \tag{11}$$

$$\sum_{t=1}^{T} y_p^t = 1 \qquad \forall p \in \{1,...,P\} \tag{12}$$

$$b_p^t - \sum_{i=1}^{t} y_s^i \le 0 \qquad \forall p \in \{1,...,P\}, \quad t \in \{1,...,T\}, \quad s \in C_p \tag{13}$$

$$\sum_{i=1}^{t} y_p^i - b_p^t \le 0 \qquad \forall p \in \{1,...,P\}, \quad t \in \{1,...,T\} \tag{14}$$

$$b_p^t - b_p^{t+1} \le 0 \qquad \forall p \in \{1,...,P\}, \quad t \in \{1,...,T-1\} \tag{15}$$

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 and lower bounds on ore processing capacity in period t in tonnes.
- mu^t and ml^t are the upper and the lower bounds 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. Equations (7) and (8) are the mining and processing capacity constraints. The relation between the ore tonnage extracted from the cuts and the total tonnage extracted from the corresponding panels is modeled in equation (9). Obviously, the total tonnage processed from the cuts in each period cannot exceed the tonnage extracted from their panels in the same period. Equations (10) and (11) control the head grade of the material sent to the processing plants for different elements of interest. Equation (12) ensures that all the panels are extracted during mine-life. Equations (13) to (15) are responsible for slope constraints between the panels.

4.1. Problem size reduction

4.1.1. Removing unnecessary variables

The mathematical model is based on two different sets of variables for mining and processing decisions. At the same time, clustering seeks to have cuts with homogenous rock types; i.e., most of the blocks grouped together have the same rock type. Therefore, cuts that contain only waste rocks do not need a variable to determine their processing schedule and their corresponding variables can be removed without sacrificing the optimality of the final solution.

4.1.2. Predecessor cone

Another technique used is borrowed from Bley, Boland et al. (2010) remove extraction decision variables for certain blocks based on their availability. To do so, the whole tonnage of rock in the predecessor cone of the block (each panel in our case) is calculated. This cone has to be extracted before one can get access to the panel based on slope constraints. The cone tonnage is then compared against the cumulative mining capacity from the first period up to period t. If the cone tonnage exceeds this cumulative mining capacity, the binary and consequently continuous variable for extracting that panel in that period is removed from the problem, because one can already be sure those variables cannot have values other than zero. Obviously, when a panel is not allowed to be extracted in a specific period, any of the cuts contained in that panel cannot be processed in that period and their corresponding variables are removed from the problem.

4.1.3. Successor cone

Since there is an assumption that all the material has to be extracted from the mine during mine-life (equation(12)), the same idea can be applied to the cone below each panel. This cone consists of panels that can only be extracted after that specific panel is completely extracted. In this case, if the total tonnage in the cone below the panel exceeds the cumulative mining capacity from period t to the end of mine life, the panel has to be completely extracted prior to period t. The two-cones concept is also used in a meta-heuristic approach based on simulated annealing in Kumral and Dowd (2005). The predecessor and successor cones are shown in Figure 4 with dark and light shades respectively.

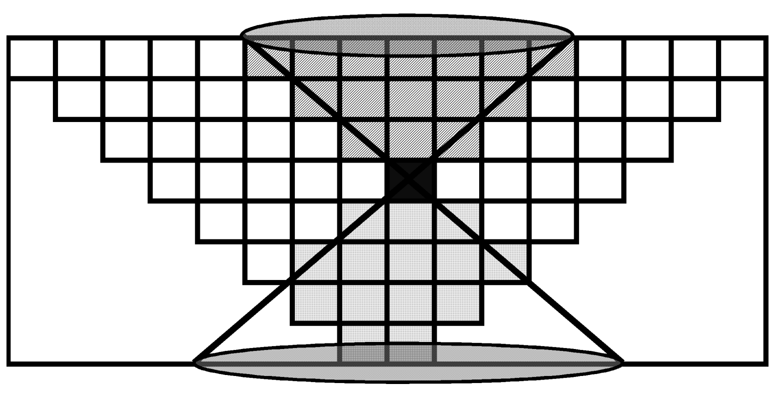

Figure 4. Predecessor and Successor Cones

4.1.4. Eliminating redundant constraints

After removing variables based on criteria mentioned earlier, we will eliminate the constraints that have become redundant. For example, not being able to extract a panel (and consequently, cuts within that panel) in a specific period, there is no need for controlling the relationship between the tonnage extracted from the panel in that period and the tonnage processed from its cuts (equation (9)).

5. Case study

In this section, we are presenting two case studies to evaluate the performance of our multi-step approach. Both case studies are from an Iron ore mine with a magnetic separation plant. The first case study is a small pushback chosen from a real-size block model that is small enough for the exact pushback design procedure. The second case study is larger and needs the heuristic procedure for determining the pushbacks.

5.1. Small iron-ore deposit

An iron ore deposit with 19,561 blocks is used as the first case study in this paper. The final pit contains 461 MT of rock with 116 MT of mineralized material. The grade of iron ore is expressed in magnetic weight recovery (MWT) with %mass unit. We first import the block model into our program in Matlab® and determine the required pushbacks using the proposed procedure. The next step is to run the clustering algorithm on the block model. The clustering is performed based on the similarity measures introduced in Tabesh and Askari-Nasab (2011) with distance weight and rock type penalty equal to 0.8 and 0 for other weights. The minimum, average and maximum blocks per cluster are set to 5, 30 and 35 respectively.

The first interesting observation is the difference between the pushbacks generated using the proposed approach and the pushbacks generated using the parameterization approach. We used the parameterization approach with revenue factors starting from 0.1 and with a step size of 0.001. We got 66 nested pits and we used Whittle auto pushback chooser to choose 3 and 5 pushbacks from the list. However, the huge gap in rock and ore tonnage contained in pits 8 and 9 makes it difficult to find pushbacks with the same tonnage of rock or ore. Comparing the pushbacks generated with the parameterization approach versus the ones generated with the proposed BIP approach in Figure 5 and Figure 6, the total rock tonnage is uniformly distributed among the pushbacks in the BIP approach. In contrast, the parameterization method creates a very large pushback and consequent smaller ones. Another important aspect of pushback design is its capability of providing pushbacks with reasonable and minable shapes. Figure 7 to Figure 9 are sample plan views comparing the resulting pushbacks from the parameterization approach and the BIP approach. We also tried to perform scheduling based on 5 and 8 pushbacks but the parameterization approach results in the same 3 pushbacks that we got earlier, as opposed to our algorithm, which can provide as many pushbacks as desired. Note that this problem, with around 20,000 blocks, is solved using the relaxation bounds and exact solver without requiring the heuristic stage. The processing times are presented in Table 1.

Table 1 – Summary of pushback design step

No. of Pushbacks	Maximum Rock Tonnage (MT) per Pushback	Maximum Ore Tonnage (MT) per Pushback	CPU Time (s)
3	145	40	1432
5	85	30	1196
8	55	18	1340

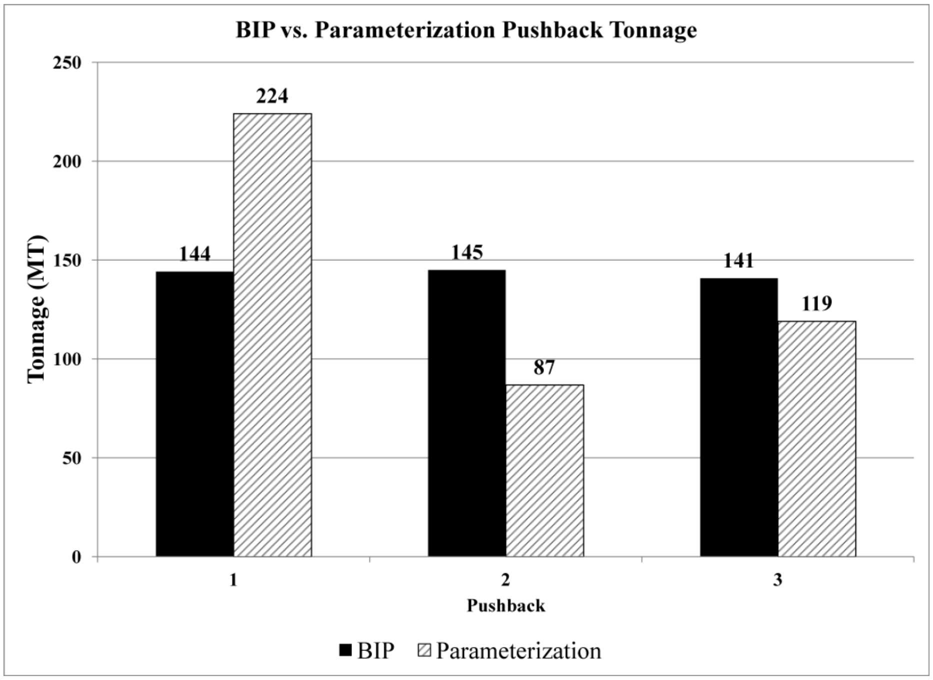

Figure 5. BIP vs. Parameterization (3 pushbacks)

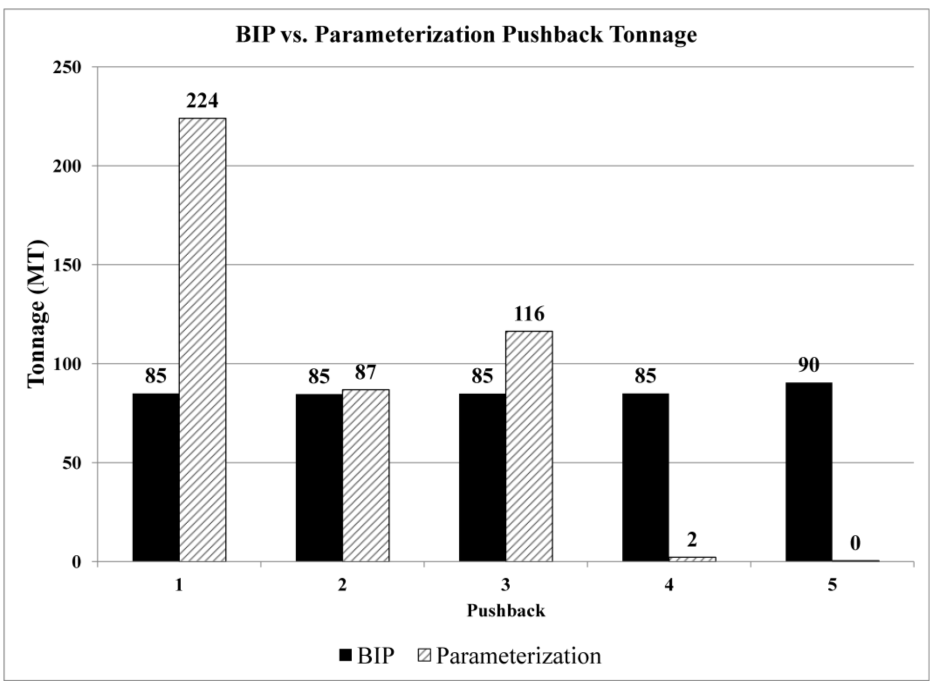

Figure 6. BIP vs. Parameterization (5 pushbacks)

The next step is to cluster blocks within the bench-phases and prepare the planning units for the MILP. The clustering approach proposed in Tabesh and Askari-Nasab (2011) forms clusters with high homogeneity in rock type and grade, but sometimes the result is not a mineable shape. A refinement procedure is added to modify the shape of the clusters and remove the sharp corners so that the resulting schedule is more practical. Figure 10(b) shows the clustering result after performing 3 shape refinement iterations which can be compared to Figure 10(a) in order to see the difference. Both figures present the clustering performed based on the 3 pushbacks we found using our BIP approach. Comparing the figures against Figure 8 we can see that the clusters are formed strictly within the pushbacks. This restriction helps when formulating the MILP since the relationship between clusters and bench-phases as the units of planning has to be modeled.

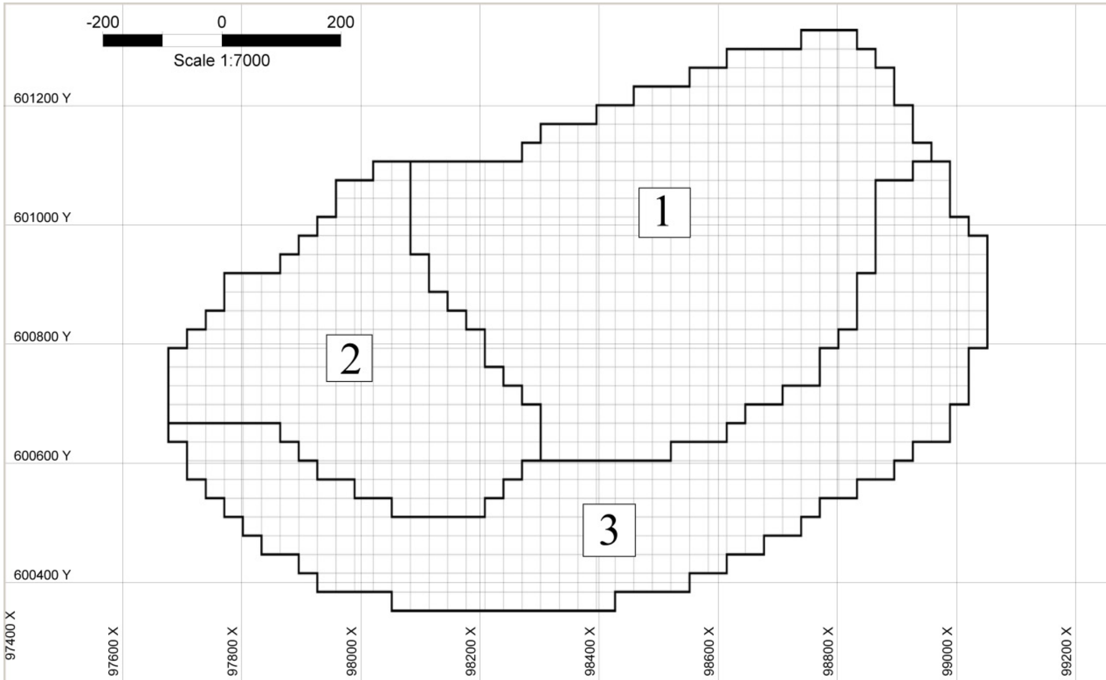

Figure 7. Parameterization Pushbacks (3)

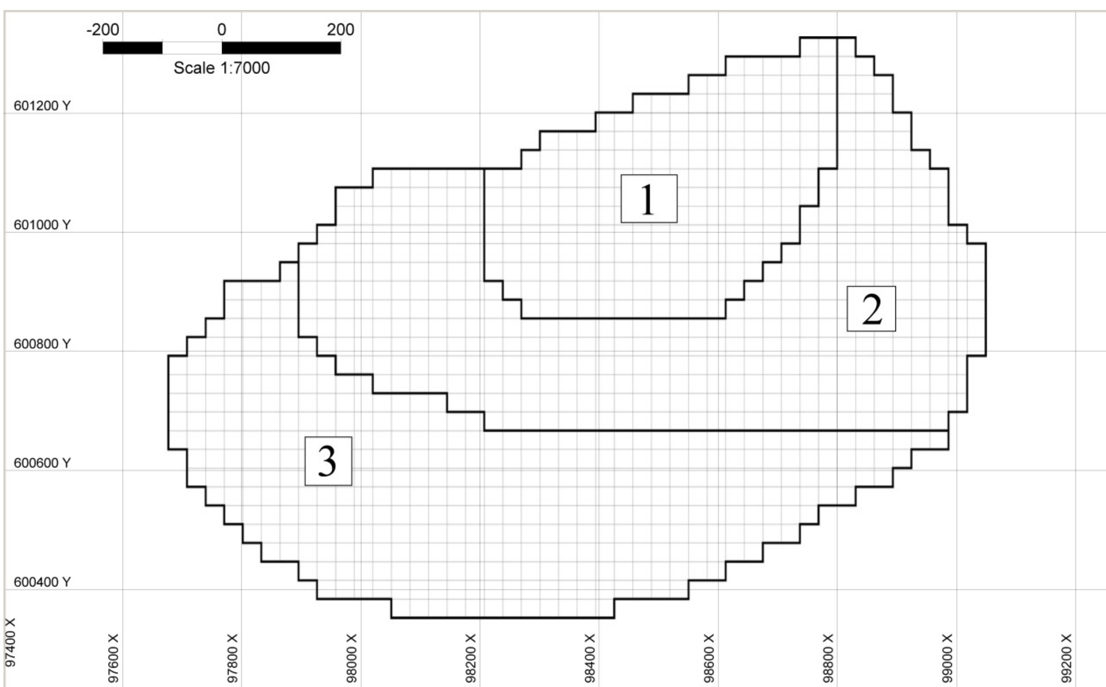

Figure 8. BIP Pushbacks (3)

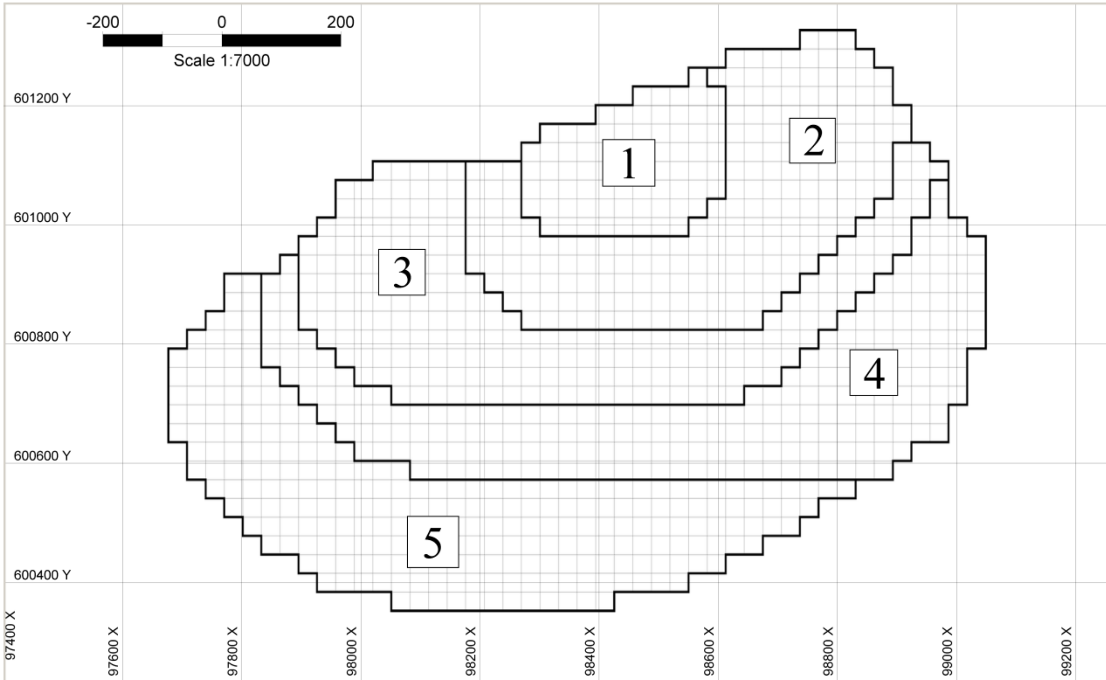

Figure 9. BIP Pushbacks (5)

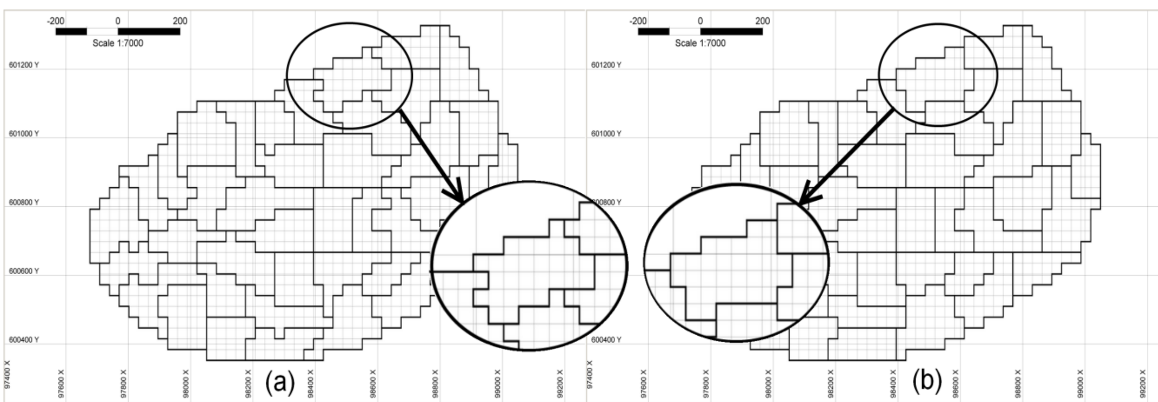

Figure 10. Clustering with Shape Refinement

Two sets of planning units are formed to be used in the MILP. The first set consists of the panels (bench-phases) determined using the BIP method and is used for making extraction decisions. The second set consists of clusters (mining-cuts) determined to divide the panels into minable units of planning with homogeneity in rock type and grade to result in a good estimate of the final revenue. Note that the tonnage processed from each cluster in each period is controlled by the tonnage extracted from its containing panel in the mathematical formulation. The MILP is formulated to maximize NPV (Net Present Value) based on 5 years of pre-stripping with 26 MT/yr mining capacity. A mining capacity of 25 MT/yr and a processing plant with 8MT/yr capacity from period 6 to the end of the planning horizon are considered. The resulting NPVs and computational processing times are presented in Table 2 It can be seen that formulating and solving the MILP with our pushbacks results in 2 to 4 percent increase in the NPV of the project.

Table 2 – NPV from MILP with different number of pushbacks

No. of Pushbacks	Parameterization pushbacks ($M)	BIP pushbacks ($M)	BIP NPV Improvement (%)
3	2163.4	2209.9	2.1%
5	2118.1	2161.2	2.0%
8	2118.1	2216.8	4.7%

Table 3 – CPU time from MILP with different number of pushbacks

Algorithm	No. of Pushbacks	No. of Panels	No. of Cuts	CPU Time (s)	No. of Integer Variables	No. of Non-Zeros in Coefficient Matrix
Parameterization	3	56	789	16.3	1120	192,250
BIP		56	804	7.9	1120	192,050
Parameterization	5	58	790	10.6	1160	194,166
BIP		100	823	21.1	2000	230,782
Parameterization	8	58	790	10.6	1160	194,166
BIP		150	947	90.3	3000	303,902

Another important factor in designing the pushbacks is their role in the production schedule. Pushbacks have to be determined such that they give the MILP more flexibility in accessing ore and waste blocks in all periods. It can be seen that the production schedules determined based on the pushbacks we designed are more favorable to mine engineers. Our pushback design (Figure 11) is capable of providing ore to the processing plant in all periods and does not suffer from shortfalls as in the parameterization approach (Figure 12).

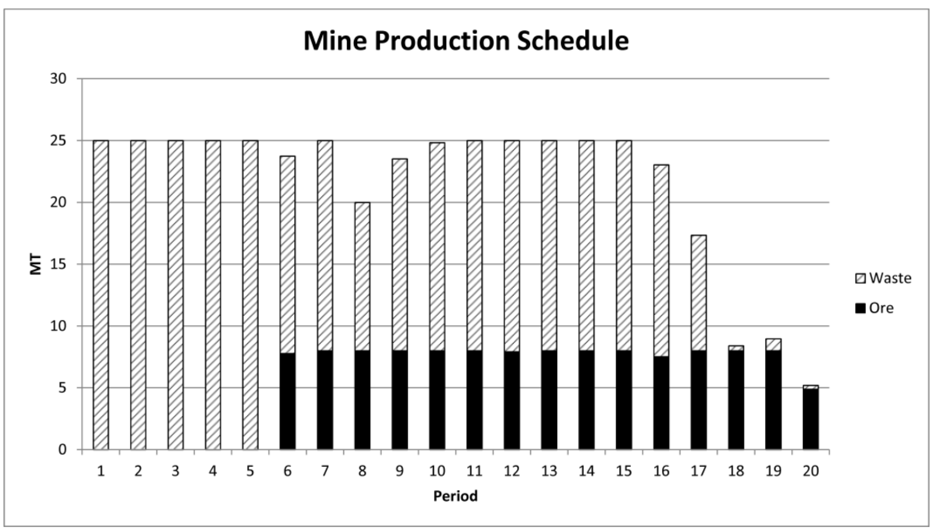

Figure 11. Production Schedule with 3 BIP Pushbacks

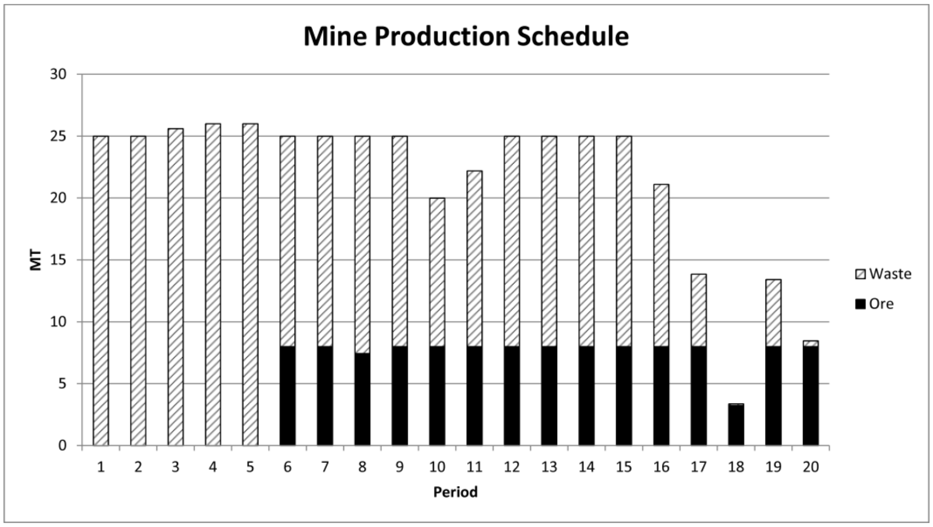

Figure 12. Production Schedule with 3 Parameterization Pushbacks

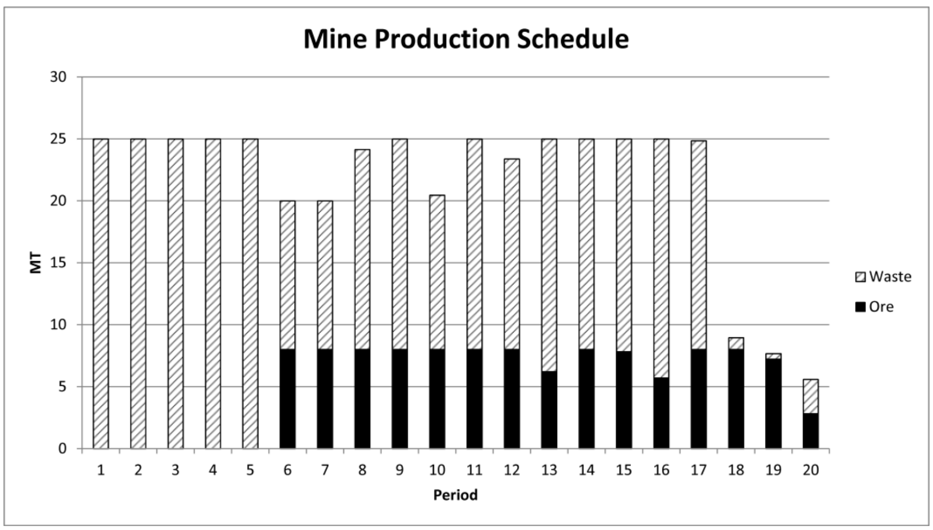

Figure 13. Production Schedule with 5 BIP Pushbacks

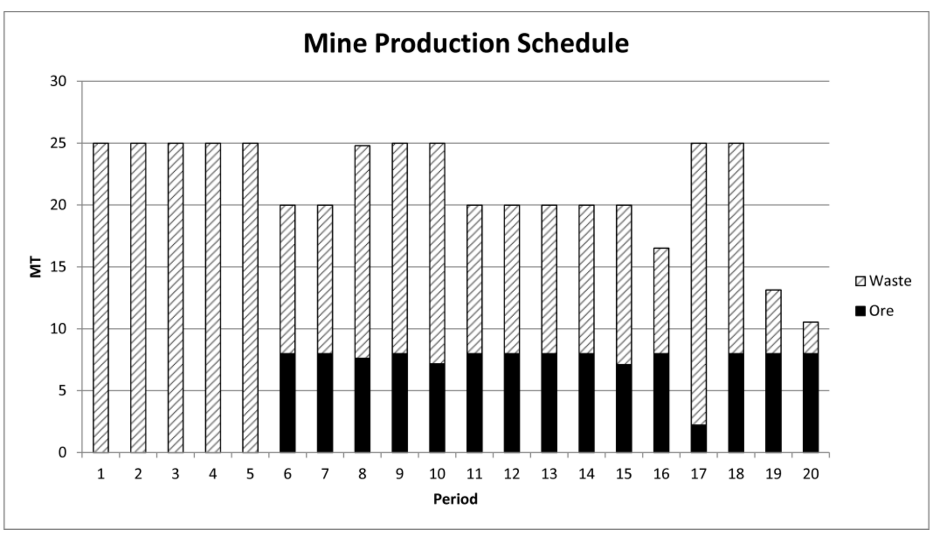

Figure 14. Production Schedule with 5 Parameterization Pushbacks

5.2. Large iron-ore deposit

Another case study is based upon the same deposit as in 5.1 but with a higher metal price that results in a larger pit with 277,000 blocks in the ultimate pit limit. The ultimate pit limit contains around 4 billion tonnes of material with around 700 thousand tonnes of ore. The ore body is tabular and goes deep (Figure 15), which makes it very hard for the parameterization approach to find pushbacks with the same tonnages.

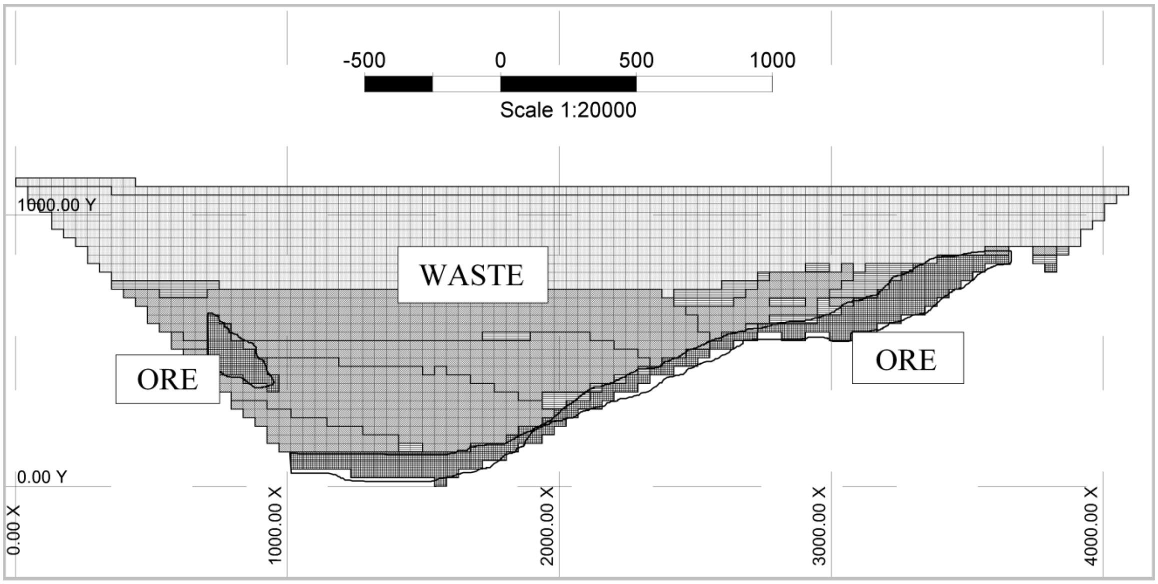

Figure 15. Sample vertical section looking West at 98100 meters Easting

Figure 16 shows the comparison of tonnages when we ask for 5 pushbacks from our algorithm and parameterization approach. Figure 17 compares the results for 8 pushbacks. We set the maximum rock tonnage in each pushback to 800 thousand and 500 thousand tonnes for the two figures respectively. The processing times for finding the pushbacks on a machine with two quad-core CPUs at 2.8 GHz are 2.7 and 3.8 hours. Pushbacks generated with the parameterization technique vary significantly in size. The largest pushback (pushback number 3) in Figure 16 is 16 times larger than the smallest one (pushback number 1). The same large pushback appears when we try to find 8 pushbacks (pushback number 2 in Figure 17). In contrast, the maximum difference between the tonnages of rock in our pushbacks is 31% for 5 pushbacks and 11% for 8 pushbacks.

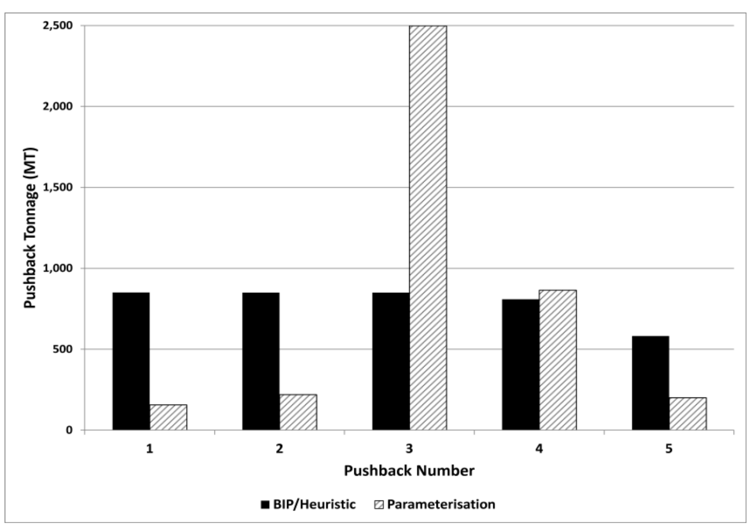

Figure 16. Comparison of 5 pushbacks

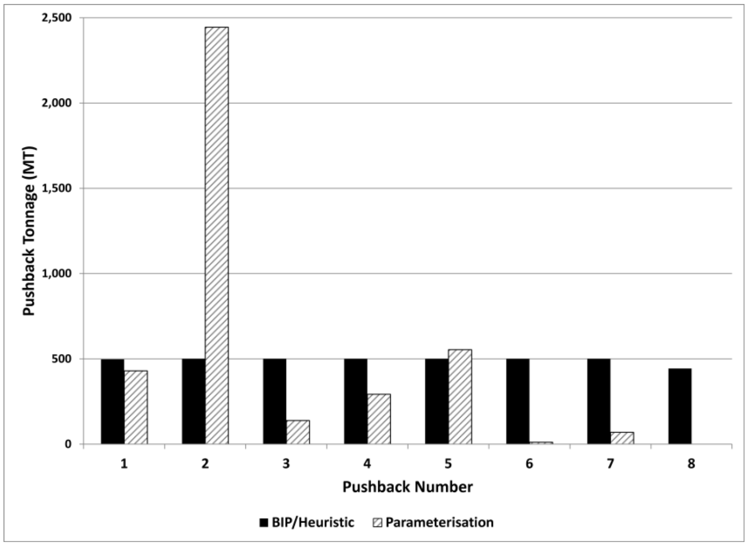

Figure 17. Comparison of 8 pushbacks

The shapes of the generated pushbacks are also important from the mining operations point of view. Very narrow pushbacks (e.g. pushback 5 in Figure 19) are not favorable since they cannot be practically mined and they do not satisfy minimum mining width requirements. On the other hand, very large pushbacks result in large panels and prevent the schedule from getting to the ore faster.

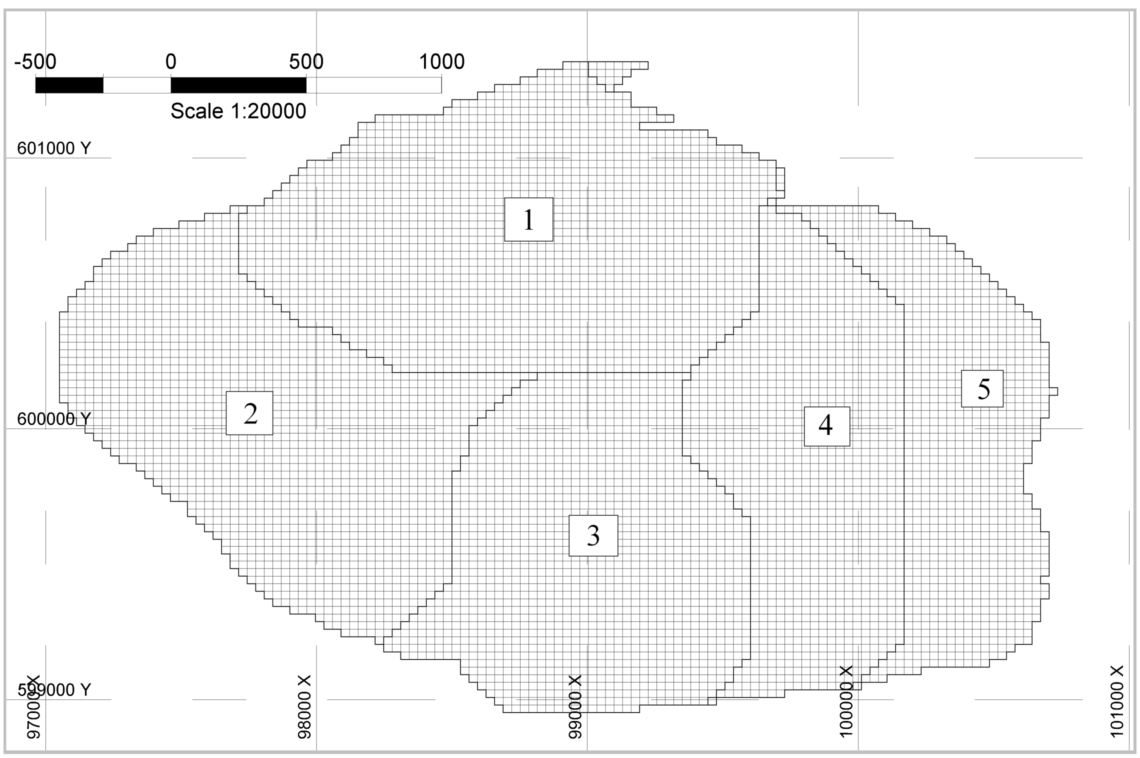

Figure 18. Sample plan view from the BIP/Heuristic pushbacks

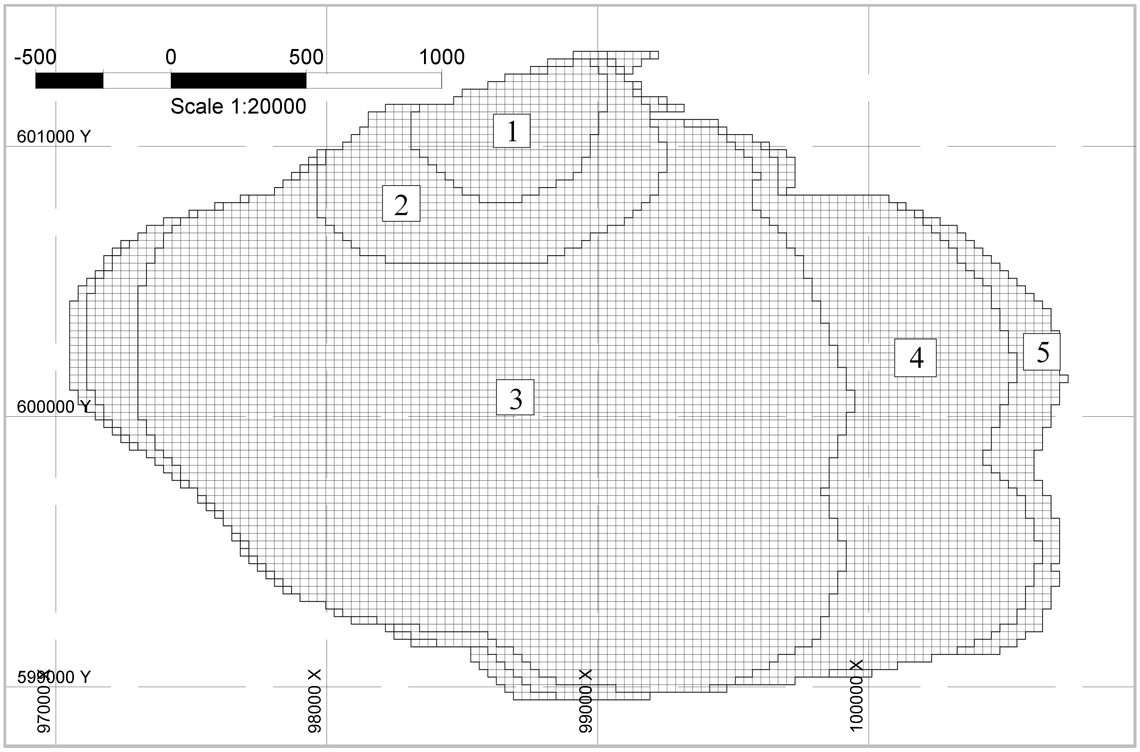

Figure 19. Sample plan view from the parameterization pushbacks

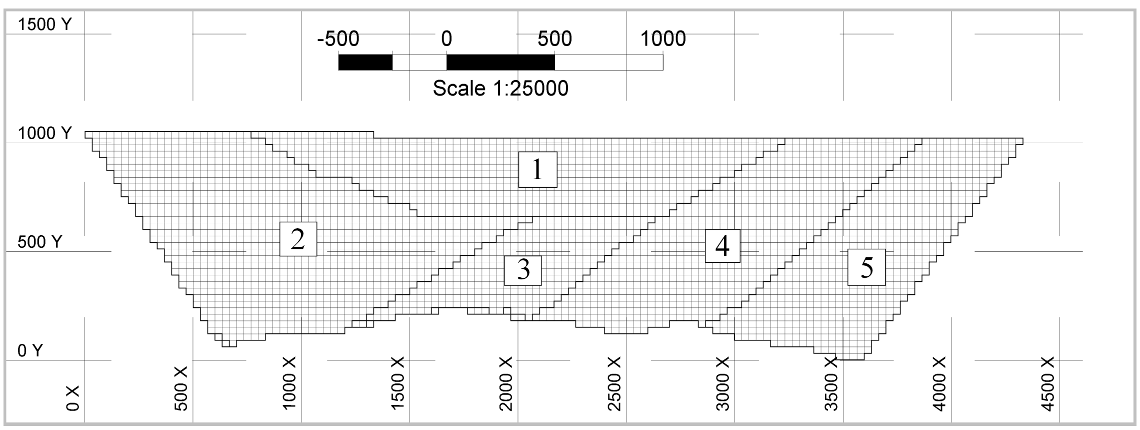

Figure 20. Sample vertical section from the BIP/Heuristic pushbacks

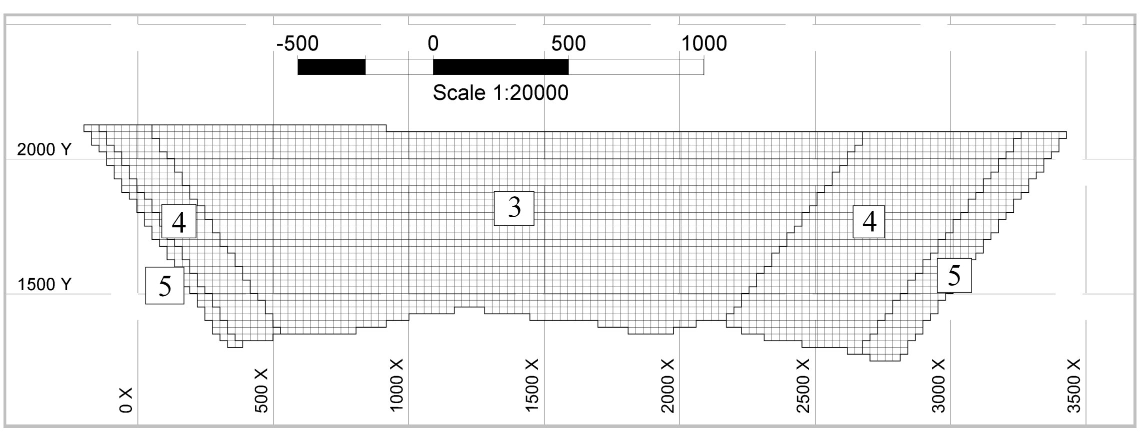

Figure 21. Sample vertical section from the parameterization pushbacks

6. Conclusion

We propose a multi-step long-term open pit mine planning optimization methodology, which includes pushback design, block aggregation into minable units, and determination of the production schedule using an MILP. The core of the pushback design algorithm is based on a hybrid methodology using binary integer programming, a greedy heuristic, and a local search routine. In contrary to the industrial practice using parameterization of revenue, the push-back design algorithm developed in this study allows the mine planner to control the mineralized material and rock tonnage in each push-back. The BIP model assigns blocks to pushbacks while respecting maximum mineralized material and rock tonnage as well as slope constraints. The results from the pushback design procedure are uniform in terms of tonnage. Moreover, the generated pushbacks are more practical in shape from the required minimum mining width operational point of view compared to pushbacks generated using parameterization approach.

A hierarchical clustering algorithm for a bench-by-bench block aggregation is presented. The algorithm aggregates blocks into selective mining units based on a similarity index, which is defined based on rock types, ore grades, and distances between blocks. The aggregated blocks are referred to as mining-cuts, which are used as the ore selection units at the production scheduling stage. A life-of-mine production scheduling MILP model is proposed that uses the bench-phases and mining-cuts as the production units and generates near-optimal practical mine production schedules in reasonable time. The MILP model inherently solves the cut-off grade optimization problem. The minimum grade of material sent to the processing plant in each period of the resultant production schedule represents the dynamic cut-off grade for that period. The performance of the proposed multi-step approach is tested on two deposits with 20 and 277 thousand blocks in the final pit limit.

The comparison shows considerable improvement against the industrial practice of revenue parameterization for pushback design. Pushbacks generated with the parameterization technique significantly vary in size. The largest pushback is 16 times larger than the smallest one. In contrast, the maximum difference between the tonnages of rock in the pushbacks generated using the BIP method is 11% for eight pushbacks generated. The shapes of the generated pushbacks are also important from the mining operations point of view. It is observed that the parameterization method generates narrow pushbacks that are not practically minable due to minimum mining width requirements.

7. References

[1] Aarts, E. H. E. H. L. and Lenstra, J. K. (2003). *Local Search in Combinatorial Optimization.* Princeton University Press,

[2] Askari-Nasab, H. and Awuah-Offei, K. (2010). *A Clustering Algorithm for Open Pit Mine Production Scheduling*, in Proceedings of Society for Mining, Metallurgy & Exploration (SME) Annual Meeting Preprint, Phoenix, Arizona, USA,

[3] Badiozamani, M. M. and Askari-Nasab, H. (2012). *Towards Integration of Oil Sands Mine Planning with Tailings and Reclamation Plans*, in Proceedings of Tailings and Mine Waste Conference, Colorado State University, Keystone, Colorado, USA,

[4] Ben-Awuah, E. and Askari-Nasab, H. (2012). *A Mathematical Programming Model for Oil Sands Production Scheduling and Waste Management*, in Proceedings of Tailings and Mine Waste Conference, Colorado State University, Keystone, Colorado, USA,

[5] Bienstock, D. and Zuckerberg, M. (2010). Solving LP Relaxations of Large-Scale Precedence Constrained Problems. in *Integer Programming and Combinatorial*

Optimization, Vol. 6080, *Lecture Notes in Computer Science*, F. Eisenbrand and F. B. Shepherd, Eds., Springer Berlin Heidelberg, pp. 1-14.

[6] Bley, A., Boland, N., Fricke, C., and Froyland, G. (2010). A Strengthened Formulation and Cutting Planes for the Open Pit Mine Production Scheduling Problem. *Computers & Operations Research*, *37* (9), 1641-1647.

[7] Boland, N., Dumitrescu, I., Froyland, G., and Gleixner, A. M. (2009). LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. *Computers & Operations Research*, *36* (4), 1064-1089.

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

[9] Caccetta, L. and Kulanoot, A. (2001). Computational aspects of hard Knapsack problems. *Nonlinear Analysis: Theory, Methods & Applications, Proceedings of the Third World Congress of Nonlinear Analysts*, *47* (8), 5547-5558.

[10] Gershon, M. E. (1983). Optimal mine production scheduling: evaluation of large scale mathematical programming approaches. *International Journal of Mining Engineering*, *1* 315-329.

[11] IBM, C. (2010). CPLEX. 12.2 ed. Armonk, New York

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

[13] Johnson, T. B. (1969). *Optimum Production Scheduling*, in Proceedings of 8th International Symposium on Computers and Operations Research, Salt Lake City, Utah, pp. 539-562.

[14] Kellerer, H., Pferschy, U., and Pisinger, D. (2004). *Knapsack Problems.* Springer,

[15] Klingman, D. and Phillips, N. (1988). Integer Programming for Optimal Phosphate-Mining Strategies. *The Journal of the Operational Research Society*, *39* (9), 805-810.

[16] Kumral, M. and Dowd, P. A. (2005). A Simulated Annealing Approach to Mine Production Scheduling. *The Journal of the Operational Research Society*, *56* (8), 922-930.

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

[18] Mieth, C. (2012). Pushback-design using mixed integer linear optimization. Master of Science Thesis, TU Bergakademie Freiberg, Freiberg, Pages 111.

[19] Newman, A. M., Rubio, E., Caro, R., Weintraub, A. s., and Eurek, K. (2010). A Review of Operations Research in Mine Planning. *Interfaces*, *40* (3), 222-245.

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

[21] Pana, M. T. (1965). *The Simulation Approach to Open-Pit Design*, in Proceedings of Transactions of the Short Course and Symposium on Computers and Computer Applications in Mining and Exploration, University of Arizona, Tucson, Arizona, pp. 1-23.

[22] Ramazan, S. (2007). The new Fundamental Tree Algorithm for production scheduling of open pit mines. *European Journal of Operational Research*, *177* (2), 1153-1166.

[23] Samanta, B., Bhattacherjee, A., and Ganguli, R. (2005). A genetic algorithms approach for grade control planning in a bauxite deposit. in *Application of Computers and Operations Research in the Mineral Industry*, doi:10.1201/9781439833407.ch44 ed, Taylor & Francis, pp. 337-342.

[24] 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*, *120* (3), 158-169.

[25] Tabesh, M. and Askari-Nasab, H. (2012). Automatic Creation of Mining Polygons using Hierarchical Clustering Techniques. *Journal of Mining Science*, *Submitted for Review in September 2012*

[26] Zhang, M. (2006). *Combining genetic algorithms and topological sort to optimise open-pit mine plans*, in Proceedings of 15th International Symposium on Mine Planning and Equipment Selection (MPES), FIORDO S.r.l., Torino, Italy, pp. 1234-1239.