A new model for automated pushback selection

Enrique Jélvez[a,b], Nelson Morales[a,b], Hooman Askari-Nasab[c]

[a] *Advanced Mining Technology Center, University of Chile, Santiago, Chile*
[b] *Delphos Mine Planning Laboratory, Department of Mining Engineering, University of Chile, Santiago, Chile*
[c] *Mining Optimization Laboratory, Department of Civil & Environmental Engineering, School of Mining and Petroleum Engineering, University of Alberta, Edmonton, Canada*

Abstract

The design of pushbacks is essential to long-term open pit mine scheduling because it partitions the pit space into individual units, controlling ore and waste production. In this paper, a new model is proposed for the pushback selection procedure, which consists of characterizing the potential pushbacks based on the comprehensive family of nested pits and selecting those ones that meet a set of criteria, for instance, bounded ore and waste. An advantage of this method is the possibility to automate the pushback selection methodology, applying well-defined criteria for the selection and reducing the time employed in the planning task.

Keywords: long-term open-pit mine planning, nested pits, pushback selection, mixed integer programming

1. Introduction

Long-term open pit mine production planning plays a key role in assessment of mining projects. One of the main output is known as production plan, which is about how and when the mining reserves will be extracted, generating a promise that commits the mine production over time. Usually, due to the complexity of the problem, the planning process is divided into stages, generating three related problems that are sequentially solved in order to obtain a tentative production plan, that is: (i) determination of the final pit, which consists of delimiting the subregion of the mine where the extraction will be carried out; (ii) pushback selection,

*Corresponding author. E-mail address: ejelvez@ug.uchile.cl (E. Jélvez).

that corresponds to a partition of the final pit that allows to guide the sequence of extraction and to control the design; and finally, (iii) temporary production scheduling, which defines in every pushback when the different zones will be extracted and which of them will be processed. This paper is focused on the second stage, about pushback selection.

The design of pushbacks is a key component for the long-term open pit mine scheduling process because it is critical for the final design of the mine and the profit obtained. The pushbacks are used as a guide for the subsequent temporary production scheduling stage, defining where the extraction process begins and where it stops. In addition, pushbacks ensure safe pit walls, assist in meeting the ore production requirement and provide a minimum operational width to accommodate mining equipment and different access to the mine, among other activities.

Traditionally, the pushback selection is often carried out manually by expert mine planning engineers using a number of empirical rules on the nested pits obtained using the methodology developed by Lerchs and Grossmann (1965). From the total number of generated nested pits, a selected group is used to define pushbacks, based upon some criteria, for instance, minimum operational width that must be maintained. However, this procedure has important limitations: (i) it does not guarantee that the ore and waste tonnages are uniformly distributed between the pushbacks, which could affect the quality of the scheduling stage (also known as gap problem); (ii) the in-situ grade uncertainty is not taken into account; and (iii) generally, the selection of pushbacks is a subjective decision of a mine planning engineer, among others.

In this paper, a new model is proposed for the pushback selection, which consists of characterizing the potential pushbacks based on the comprehensive family of nested pits and selecting those pushbacks that meet certain specified conditions, for instance, bounded ore and waste, using mixed integer linear programming. Details of the proposed model followed by the illustration of its performance using a number of deposits as case studies are provided.

The paper is organized as follows: in Section 2 it provides a summary of the most relevant (and best-known) approaches found in the literature. Section 3 contains all the details concerning the modeling, notation and problem statement. All the implementation details, and the numerical results obtained are given and commented on Section 4. Finally, Section 5 contains some concluding remarks and perspectives.

2. Related work

The most common approach for producing pushbacks is the pit limit parameterization, applied on an algorithm that produces an ultimate pit limit. This is done by scaling the block economic value when the algorithm is used to define the pit limit, partitioning the ultimate final pit into a set of nested pits from which pushbacks are manually selected. From a point of view, it is possible to classify the algorithms used to find pit limit as optimal or heuristics, depending on whether optimality can be ensured. A complete reference for this topic can be found in Meagher et al. (2014a), where a review of methods are examined in order to produce pushback designs, particularly, how they can tackle the gap problem.

2.1. Parameterization and optimal approach

Lerchs and Grossmann (1965) proposed the earliest algorithm (LG) for the ultimate pit limit, which is formulated as maximal closure problem using graph theory, where nodes represent blocks and arcs represent precedences between blocks. The authors noted that it is possible to generate nested pits by repeatedly parameterizing the block revenues and applying the LG algorithm. Vallet (1976) developed a variant of the LG algorithm that produces a series of nested pits by searching, at each stage, for the pit with the highest revenue/volume ratio among all the feasible pits in the graph. Another variant of this algorithm was developed by Zhao and Kim (1992), where the blocks are aggregated after it has been noted that a block in a profitable group lies under a block in an unprofitable group. Seymour (1995) modified the LG algorithm to incorporate pit volume as a parameter, following the approach of the LG method with the addition of the parametrized variables and the added ability to notice when a subtree can be regarded as a small pit, producing then a series of nested pits. Wang and Sevim (1995) proposed a pushback design algorithm imposing an upper bound on the size of the incremental pushbacks in order to overcome the gap problem. Ramazan and Dagdelen (1998) developed the minimum stripping ratio pushback design algorithm, where among all possible pushbacks with the same size, the algorithm finds the one with minimum stripping ratio.

Picard (1976) showed that the ultimate pit problem is equivalent to the *maximum closure problem*, which in turn, can be reduced to the *min cut* problem: given a directed graph $G = (V, A)$ with weight function w defined over the nodes, one looks for a subset of nodes $U \subset V$ such that $\sum_{u \in U} w(u)$ is maximal but $u \in U, (u, v) \in A \Rightarrow v \in U$. This allows one to use known efficient algorithms for maximum flow to find the ultimate pit.

Using this fact, Hochbaum and Chen (2000); Hochbaum (2008); Chandran and Hochbaum (2009) proposed to tackle the pit limit problem by means of existing efficient algorithms for the min cut problem, showing that the LG algorithm can be used as a *network flow algorithm*, best known as *pseudoflow algorithm* as well. From the series of normalized trees they showed how one could obtain an optimal network flow. They also analyzed the runtime of the LG algorithm and improved it by scaling techniques (different from those used to generate pushback designs) to show that LG can be implemented to run in $O(mn \log n)$ time, where m and n are the number of arcs and nodes, respectively.

2.2. Parameterization and heuristic approach

A number of heuristic procedures have been developed to generate pits. Given a block b and a slope angle requirement, the set of blocks that must be extracted before block b form a *cone* with b at its base. Based on this, Floating Cone method was described by Pana (1965) and Carlson et al. (1966). For each positive (ore) block, this method involves constructing a cone with sides oriented parallel to the pit slope walls, and then determining the value of the cone by summing the values of blocks enclosed within it. If the value of the cone is positive, all blocks within the cone are included into the pit. This process starting from the top level and moves downward searching for positive blocks. Searching process continues until no positive cones can be found. The basic assumption is that every cone in the optimal pit is profitable, whereas in fact an optimal pit may consist of a collection of cones none of which alone is of positive value, but together the cones share negative value blocks and have total weight which is positive. Mixing this procedure with parameterization, either to finding the best pit at different cutoff grades or changing the metal price, floating cone method should be used in pushback design. There exist some variations of it, for example, Wright (1999) and Kakaie et al. (2012).

A different approach was proposed by Dagdelen and François-Bongarçon (1982) and Francois-Bongarcon and Guibal (1984) in replacing the economic parameters (metal price, mining cost, cutoff grades, etc) by ore content and recoverable metal quantity. They presented an algorithm that generates a series of nested pits, but parameterizing the total metal content and volume of incremental pit. A pit belongs to the parameterization if it contains the highest quantity of recoverable metal.

Dagdelen and Johnson (1986) used Lagrangian relaxation in the context of linear programming (LP). This technique removes a troublesome constraint from the LP and place it in the objective function, and can be applied to the problem of finding a pit of a bounded tonnage. Modeling this problem as a LP with a capacity constraint, it is straightforward to see that relaxing the complicated constraint, the problem being solved is the ultimate pit limit problem, where the economic value of the orebody model is scaled down by a constant factor λ. Choosing λ to be zero this is equivalent to finding the ultimate pit limit. As λ gets larger one can expect to get smaller and smaller pits. One can therefore view the procedure of finding nested pits by Dagdelen and Johnson's Lagrangian parametrization as an equivalent procedure to that of scaling the orebody model value and running the LG algorithm to get a series of nested pits. It therefore suffers from the same gap problems as those discussed in the review of existing methods in an earlier section. Choosing appropriate values of λ is not always straight forward either, it may take quite a bit of time to try and find the value of λ to produce pits close to the desired tonnage, and it might not even be possible to produce a pit of the desired size with this technique. Somrit and Dagdelen (2013) presented a max flow-Lagrangian based phase design algorithm, including time value of money and blending requirements in its formulation.

2.3. Other approaches

Some efforts to incorporate the mine production scheduling decision into the pushback selection are: Elkington and Durham (2011) outline a method for simultaneously optimizing intermediate and ultimate pushback selection with the determination of production scheduling, cutoff grades and stockpiling to maximize net present value. This work is done at bench-phase resolution (aggregation of blocks). Meagher et al. (2014b) presented a method based on *pipage rounding* that generates near maximal expected profit and dynamically defines the optimal cutoff grade, and which aims to produce a set of pushbacks in a way that the total discounted profit to be generated through production scheduling is maximized. These efforts are made so that pushback definition does not depend on the judgment of decision-maker. Tabesh et al. (2014) developed a pushback design algorithm based on mathematical programming and a two-step heuristic based on greedy and local search approaches, where the algorithm assigns blocks to pushbacks such that the ore and rock tonnages in each pushback do not exceed maximum values.

In the last years, the incorporation of several sources of uncertainty in the planning process has attracted great interest of the researchers. For example, those that have included geological uncertainty: Gholamnejad and Osanloo (2007), Consuegra and Dimitrakopoulos (2010), Goodfellow and

Dimitrakopoulos (2013); and market uncertainty as well: Meagher et al. (2009).

3. Modeling, notation and problem statement

In this section the main notation and formulation for the automated pushback selection will be presented.

3.1. Nested pit generation

The methodology is based on the following assumptions in order to obtain a series of nested pits.

1. A set of blocks denoted as $\mathcal{B}$.
2. Set $\mathcal{B}$ represents the final pit of the deposit and the blocks are denoted with letters b and b'.
3. The generation of nested pits has a great impact on pushback selection. They are generated using a metal price parameterization and an economic block model.

   3.1. Parameterization: it corresponds to a sequence of N revenue factors $0 < \lambda_1 < \lambda_2 < \ldots < \lambda_N \leq 1$, which are used to scale the metal price. Each revenue factor produces a pit. These parameters normally are obtained defining an initial, step and final values, but this choice is dependent on the subjective criteria from the mine planning evaluator.

   3.2. Economic block model: if for all $i \in [1, N]$, v_b^i represents the value of block b when the metal price is scaled by the revenue factor λ_i, then

$$v_b^i = \begin{cases} [(P \cdot \lambda_i - SC) \cdot R \cdot y_b - MC - PC] \cdot ton_b & \text{if } y_b \geq \frac{MC+PC}{(P\lambda_i - SC)\cdot R} \\ -MC \cdot ton_b & \text{otherwise} \end{cases} \quad (1)$$

   where P is the metal price, SC is the selling cost, y_b is the ore grade of block b, R represents the metallurgical recovery, PC and MC the processing and mining cost, respectively, and ton_b is the total tonnage of block b. All factors must be in the correct units.
4. It solves a series of final pit problems (UPIT) by considering each economic block model according to (1) in order to obtain a family of nested pits.

$$\begin{aligned} \text{(UPIT)} \quad \max \; & \sum_{b \in \mathcal{B}} v_b^i x_b \\ \text{s.t.} \; & x_b \leq x_{b'} && \forall\, b \in \mathcal{B}, b' \in \texttt{PREC}(b) \\ & x_b \in \{0, 1\} && \forall\, b \in \mathcal{B} \end{aligned}$$

In this model, for all $i \in [1, N]$, x_b is a binary variable for intermediate pit selection, where $x_b = 1$ means that block b belongs to pit i and $x_b = 0$ otherwise. $\texttt{PREC}(b)$ represents the subset of blocks with a precedence arc from block b: this set is completely defined by indicating the slope angle and the number of levels (benches) upper from block b.

In the parameterization step, the best choice of revenue factors in terms of controlled increments of tonnage between two successive pits is not guaranteed (gap problem). A simple form to potentially refine these increments may be done by means of Procedure 1.

Procedure 1. Steps in order to refine pits in terms of volume and tonnage.

1. *Identify where it is possible to refine*: suppose that two consecutive revenue factors λ_k and λ_{k+1} are identified, where the respective pits present a huge difference in terms of tonnage. The objective is to split, if possible, the intermediate pit into M parts.
2. *Calculate new intermediate revenue factors*: for this, the following expression is used

$$\lambda(j) = \frac{\lambda_k}{N} + j\frac{\lambda_{k+1} - \lambda_k}{MN} \quad \forall\ j \in [1, M-1]$$

   where N is the number of revenue factors and M is the potential number of parts resulting in the refinement. Although there is no general rule for choosing M, a value of 10 is recommended in order to balance the number of potential intermediate pits and computation time when solving the additional pit problems.

It is very important to highlight that this procedure does not guarantee to reduce the gap between successive pits, because this strongly depends on the distribution of ore in the deposit. For example, in deposits with diseminate ore and huge tonnage of waste material, it is not possible to ensure such refinement.

3.2. The model for automatic pushback selection

Once the set of nested pits is defined, the next step is to use these pits in order to select pushbacks that help to control the production planning. Under the traditional methodology, one of the difficulties and limitations this task has is related to the non-application of well-defined criteria to select pits that represent pushbacks, and its success strongly depends on the expertise of mine planner in order to control key indicators, such as gapping problem, stripping ratio between waste and ore, minimum operational spaces, among others. In this paper a new model is proposed: characterizing the pushbacks from the set of nested pits, the model allows to find the best combination of pushbacks by satisfying a number of different criteria, which are related to the key indicators said before.

Let N be the number of nested pits generated by metal price parameterization and let P_i be the i^{th} nested pit, with $i \in [1, N]$. Define a dummy pit as one without blocks (empty set), denoted as P_o. The property of nested pits (Lerchs and Grossmann, 1965) allows to write

$$P_o \subseteq P_1 \subseteq \ldots \subseteq P_N$$

Based on the above set of assumptions, a pushback F_{ij} is defined as the set of blocks within the difference between two pits, P_i and P_j, that is

$$F_{ij} = P_i \setminus P_j \quad \forall\ i \in [1, N],\ j \in [0, i-1].$$

Some considerations:

(i) The set of pushbacks will be denoted by $\mathcal{F}$.

(ii) Given N different nested pits, the total number of theoretically possible pushbacks is $\frac{N(N+1)}{2}$.

(iii) Also, there exist 2^{N-1} different ways to select a partition of the final pit based on pushbacks.

It is impossible to test all combinations (enumerative schema), considering the exponential behavior of (iii). For example, if $N = 30$ the number of possible combinations is more than 500 millions.

For each pushback, let $\texttt{rton}_{ij}$ be the total tonnage (rock tonnage) of pushback F_{ij}. Similarly, $\texttt{oton}_{ij}$ represents the ore tonnage of pushback F_{ij}. The desired rock and ore tonnages in each pushback are given by upper and lower limits and they are denoted respectively by: (i) rock tonnage in pushback F_{ij}, RT^+_{ij} and RT^-_{ij}; and by (ii) ore tonnage in pushback F_{ij}, OT^+_{ij} and OT^-_{ij}.

To automate the pushback selection, it is necessary to characterize the family of potential pushbacks. The set of *predecessor pushbacks* defined at a given one F_{ij} is denoted by

$$\texttt{PREC}_{ij} = \{F_{i_o j_o} \in \mathcal{F}\ :\ i_o = j, j_o \in [0, j-1]\} \quad \forall\ i \in [2, N],\ j \in [1, i-1]$$

Similarly, the set of successor pushbacks at F_{ij} is defined by

$$\texttt{SUC}_{ij} = \{F_{i_o j_o} \in \mathcal{F}\ :\ i_o \in [i+1, N], j_o = i, \} \quad \forall\ i \in [1, N-1],\ j \in [0, i-1]$$

Now, the model for automated pushback selection is presented from a general point of view, showing how to define variables, possible objectives to select from and what constraints must be satisfied.

3.2.1. Variables

The variables related to the decision of whether to select or not a given pushback are

$$x_{ij} = \begin{cases} 1 & \text{if pushback } F_{ij} \text{ is selected,} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

3.2.2. Objective function

There exist a number of alternatives in order to define the objective function. The particular choice will depend on the mine planner objectives and the particular case study. In general, the recommendations in order to realize a good choice are to ensure the extraction of mineral as soon as possible, which is related to nested pits definition, based on parameterization over metal price, but also to ensure equilibriums of waste and ore movements across life of mine, postponing the waste extraction as much as possible, while at the same time, the design of pushbacks keeps minimum operational spaces. Some examples of objective functions that should be chosen are:

1. Minimization of the tonnage differences among selected pushbacks (gap problem).
2. Minimization of the stripping ratio (waste/ore) into the first pushbacks, delaying as much as possible the waste extraction.
3. Maximization of the ore tonnage within each pushback.
4. Optimization of the number of pushbacks, for instance, what is the minimum number of pushbacks that satisfy the requirements given by the constraints.
5. Definition of an operational design measure in order to maximize the pushback width along a preferred direction.

3.2.3. Constraints

The constraints are classified as *structurals* (a.1)-(a.3), because they are always applied and keep the model well-defined; *specials* (b.1) and (b.2), because they impose conditions on the pushbacks and are applied only if these are not in conflict with the objective function; and *simplifiers* (c.1) and (c.2), which want to set variables and reduce the search space.

(a) Structurals: pushback selection must be a partition of the final pit. Therefore, it must be imposed by the following constraints:

(a.1) *Initial and final pushbacks*: the pushback selection must consider an initial pushback F_{i0} and a final pushback F_{Nj} into the partition, then

$$\sum_{i=1}^{N} x_{\mathrm{i0}} = 1 \tag{3}$$

$$\sum_{j=0}^{N-1} x_{\mathrm{Nj}} = 1 \tag{4}$$

(a.2) *Relationship with predecessor pushbacks*: in order to select a pushback F_{ij}, it must have selected one and only one of their predecessor pushbacks $F_{i_o j_o} \in \texttt{PREC}_{ij}$ as well

$$x_{ij} \leq \sum_{F_{i_o j_o} \in \texttt{PREC}_{ij}} x_{i_o j_o} \leq 1 \quad \forall\, i \in [2, N], j \in [1, i-1] \tag{5}$$

(a.3) *Relationship with successor pushbacks*: in order to select a pushback F_{ij}, it must have selected one and only one of their successor pushbacks $F_{i_o j_o} \in \texttt{SUC}_{ij}$ as well

$$x_{ij} \leq \sum_{F_{i_o j_o} \in \texttt{SUC}_{ij}} x_{i_o j_o} \leq 1 \quad \forall\, i \in [1, N-1], j \in [0, i-1] \tag{6}$$

(b) Specials: they are optional constraints that affect some attribute such as:

(b.1) *Predefined number of pushbacks*: instead of optimizing the number of pushbacks, as given in point 4 of Section 3.2.2, this is a pre-set parameter in the model equal to n_o, where $1 \leq n_o \leq N$, then the respective constraint is

$$\sum_{i=1}^{N} \sum_{j=0}^{i-1} x_{ij} = n_o \tag{7}$$

Of course, it is possible to be less strict with this condition, imposing upper and/or lower bounds to the number of required pushbacks.

(b.2) *Pushback tonnages*: the quantity of ore and rock tonnages should be bounded (gapping problem). For this, it makes

$$OT^{-}_{ij}\ x_{ij} \leq \texttt{oton}_{ij}\ x_{ij} \leq OT^{+}_{ij} \quad \forall\, i \in [1, N], j \in [0, i-1] \tag{8}$$

$$RT^{-}_{ij}\ x_{ij} \leq \texttt{rton}_{ij}\ x_{ij} \leq RT^{+}_{ij} \quad \forall\, i \in [1, N], j \in [0, i-1] \tag{9}$$

(c) Simplifiers: these constraints aim to set variables in order to reduce the search space of feasible solutions. Basically, they take advantage that it is not possible to ensure that all nested pits are different, that is, $P_o \subsetneq P_1 \subsetneq \ldots \subsetneq P_N$ when considering *revenue factors* $0 < \lambda_1 < \ldots < \lambda_N$.

(c.1) *Setting variables I*: if $K < N$ first *revenue factors* give empty pits $P_1 = \ldots = P_K = P_o$, then it is possible to set the pushbacks associated with these pits, that is

$$x_{ij} = 0 \quad \forall\ i \in [1, K], j \in [0, i-1] \tag{10}$$

(c.2) *Setting variables II*: if K first pits are empty, the pushbacks F_{ij} associated with $i = K+1, \ldots, N$ will be equal, with $j = 0$, therefore, it is possible to set the remain variables

$$x_{ij} = 0 \quad \forall\ i \in [K+1, N], j \in [1, i-1] \tag{11}$$

Figure 1 shows a simplified procedure: a set of nested pit is calculated and, based on specific criteria, for instance, bounded ore and rock tonnages, the best combination of pushbacks is selected using the proposed model.

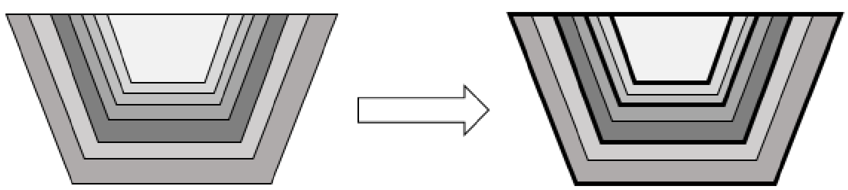


Figure 1: Nested pits generated by a parameterization of the metal price (left). Some of these pits are selected as pushbacks (right), using the model to automate the selection.

3.3. An example of automated pushback selection model

A very interesting particular case is about controlling the differences among pushback tonnages. If the growth rate of tonnages for the nested pits is relatively low and constant, the gap problem in the pushback selection should be avoided. In this case, in order to represent the desired objective function, a minimization of total deviations regarding to a reference value shall be used. The reference value corresponds to the tonnage of each pushback when it has a equipartition of the ultimate final pit into n_o pushbacks, and this value is given by $\frac{\texttt{rton}_{\texttt{N0}}}{n_o}$. An alternative is by means of quadratic functions such as variance, but another one is the mean absolute deviation (MAD), which in this case measures the mean of the deviations of all values (tonnages) from the reference value, disregarding their sign. More precisely

$$\text{MAD} \; = \; \frac{1}{n_o}\sum_{i=1}^{N}\sum_{j=0}^{i-1}\left|\texttt{rton}_{ij}\, x_{ij} - \frac{\texttt{rton}_{\text{N0}}}{n_o}\right| \tag{12}$$

Then, the mean absolute deviation works as variability measure that would be minimized to select a set of pushbacks from nested pits. While the absolute value function (12) is non-linear, in order to transform this problem to a linear equivalent formulation (see Mansini et al. (2015)), by using the fact that for $z > 0$, $|ax - b| \leq z \Leftrightarrow -z \leq ax - b \leq z$. Therefore, including $\frac{N(N+1)}{2}$ auxiliary variables $z_k \geq 0$, the following problems are equivalents

$$\min_{\mathbf{x}} \left\{ \sum_k |a\, x_k - b| : \mathbf{x} \in X \right\} \iff \min_{(\mathbf{x},\mathbf{z})} \left\{ \sum_k z_k : \mathbf{x} \in X,\, -z_k \leq a\, x_k - b \leq z_k,\, \mathbf{z} \geq 0 \right\} \tag{13}$$

The important thing from (13) is that it allows pass from a non-linear program with MAD objective function to an equivalent linear programming, which can be solved using mixed integer linear programming tools. Then, the problem to automate the pushback selection task (PS) consists of solving:

$$(PS) \;\; \min \sum_{i=1}^{N}\sum_{j=0}^{i-1} z_{ij} \tag{14}$$

$$s.t. \sum_{i=1}^{N} x_{\text{i0}} = 1 \tag{15}$$

$$\sum_{j=0}^{N-1} x_{\text{Nj}} = 1 \tag{16}$$

$$x_{ij} \leq \sum_{F_{i_o j_o} \in \texttt{PREC}_{ij}} x_{i_o j_o} \leq 1 \qquad \forall\, i \in [2, N],\, j \in [1, i-1] \tag{17}$$

$$x_{ij} \leq \sum_{F_{i_o j_o} \in \texttt{SUC}_{ij}} x_{i_o j_o} \leq 1 \qquad \forall\, i \in [1, N-1],\, j \in [0, i-1] \tag{18}$$

$$\sum_{i=1}^{N}\sum_{j=0}^{i-1} x_{ij} = n_o \tag{19}$$

$$x_{ij} = 0 \qquad \forall\, i \in [1, K],\, j \in [0, i-1] \tag{20}$$

$$x_{ij} = 0 \qquad \forall\, i \in [K+1, N],\, j \in [0, i-1] \tag{21}$$

$$\left(\texttt{rton}_{ij} - \frac{\texttt{rton}_{\text{N0}}}{n_o}\right) x_{ij} \leq n_o\, z_{ij} \qquad \forall\, i \in [1, N],\, j \in [0, i-1] \tag{22}$$

$$\left(\frac{\texttt{rton}_{\text{N0}}}{n_o} - \texttt{rton}_{ij}\right) x_{ij} \leq n_o\, z_{ij} \quad \forall\, i \in [1, N],\, j \in [0, i-1] \quad (23)$$

$$x_{ij} \in \{0, 1\} \quad \forall\, i \in [1, N],\, j \in [0, i-1] \quad (24)$$

$$z_{ij} \geq 0 \quad \forall\, i \in [1, N],\, j \in [0, i-1] \quad (25)$$

Expression (14) represents the objective function, where the sum of auxiliary variables that are upper bound of the differences among pushback tonnages is minimized; constraints (15) to (18) guarantee the pushback selection is a partition of the final pit; (19) ensures a selection of n_o pushbacks; (20) and (21) reduce the number of variables, setting their values to zero; (22) and (23) relate the auxiliary variables with MAD function; and finally (24) and (25) denote the nature of variables.

4. Implementation and results

In this section the methodology above presented was applied on two case studies: one from a gold deposit and the other from a copper deposit. The aim of these experiments is essentially to evaluate the performance of the proposed model on two different kind of orebodies. In both instances the model implemented was (PS), that is, throughout equations (14) to (25). The experiments consist of two stages: (i) a number of nested pits are generated by scaling the metal price following the steps of Section 3.1, and (ii) the proposed model for automatic pushback generation is applied.

In order to implement the model (PS), the `Minelink` library developed at Delphos Mine Planning Laboratory at University of Chile (Delphos, 2015) is used, which implements data structures to store the block model, precedence arcs and then to generate nested pits. The library is written in C++, but there also exists a wrapper to use it from `Python`, a general-purpose free available scripting language, version 2.7 (Python, 2015). (PS) is implemented inside `PuLP` library (Mitchell et al., 2011). `GUROBI` is used as solver, version 5.6.3 for mixed integer linear optimization (Gurobi Optimization, 2015). Execution of the code was done on an Intel Core i7-4510U machine with 8Gb ram, running Windows version 8.1. This machine has 4 processors that run at 3.10 GHz. Before reporting the main results, the instances and the parameters used in the experiments will be introduced.

4.1. Instances

The first case study is known as McLaughlin gold deposit, which was located in the northern Coast Ranges of California. This deposit was mined from 1985 until 1996, but gold processing continued through 2002. It was a world-class gold orebody and one of the world's finest examples of a hot spring-type epithermal precious metals system. The dataset (blocks coordinates, tonnage and ore grade) is obtained from the Minelib library (Espinoza et al., 2013) and this case will be called MCL, for short.

The second case study corresponds to a part of a porphyry copper deposit, but due to confidentiality considerations no more details can be given. This deposit will be called BMT.

The parameters used for the generation of nested pits for both cases are shown in Table 1, where slope precedence relationships are given by technical parameters (slope angle and number of levels) and an economic block model is defined from Equation (1) by using values of metal price, recovery, mining, processing and selling costs. Remember that ore grades and tonnages from each block are given as attributes in the block model.

The MCL block model contains more than 2 millions blocks but 247,793 blocks are in the ultimate final pit, which has 250.8 (Mton) of rock and 194.1 (Mton) of mineralized material. On the other hand, the BMT block model has near to 400,000 blocks, but inside the final pit there are 106,781 blocks, having 250.6 (Mton) of total tonnage and 172.1 (Mton) of mineral. In both cases, all blocks not included in the ultimate final pit are removed from the set and not considered in the subsequent steps of the procedure.

Table 1: Parameters to generate nested pits from MCL and BMT.

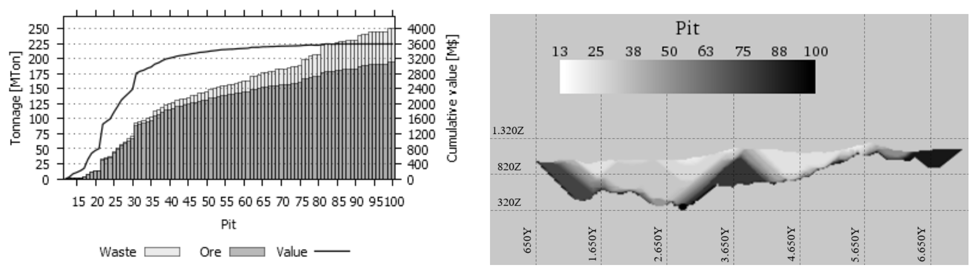
Technical & economic parameters	Symbol	Value (MCL)	Value (BMT)
Slope angle ; # levels	PREC(b)	45° ; 5	45° ; 5
Metallurgical recovery	R	0.76	0.90
# revenue factors	N	100	90
Metal price	P	1,100 ($/ozt)	2.5 ($/lb)
Selling cost	SC	100 ($/ozt)	0.4 ($/lb)
Mining cost	MC	1.5 ($/ton)	3.2 ($/ton)
Processing cost	PC	8.2 ($/ton)	9.0 ($/ton)

Now, considering a family of N revenue factors as $\lambda_i = i/N$, for $i \in [1, N]$, and solving repeatedly (P_{λ_i}), a set of N potential pits would be generated.

4.2. Results

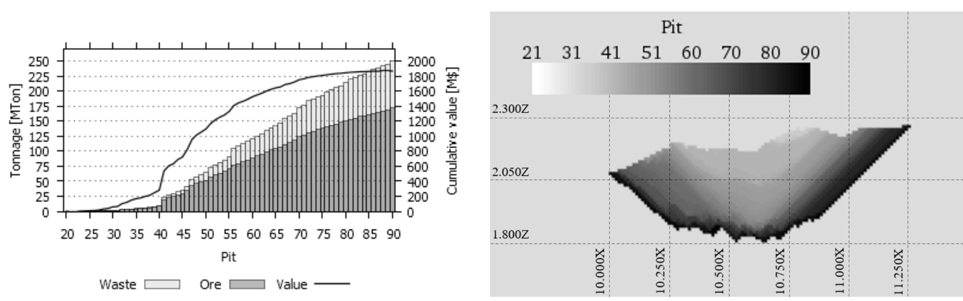
In MCL case, time employed to compute the family of nested pits was 50 seconds. The first non-empty pit was found using a revenue factor

$\lambda = 0.13$, therefore $K = 12$. From BMT block model, the set of nested pits was found in 42 seconds and the first non-empty pit corresponds to a revenue factor $0.2\overline{3}$, so that in this case $K = 20$. In Figures 2 and 3, the pit by pit graphs show ore and waste tonnages for each pit and cumulative undiscounted value, besides long section views from obtaining nested pits: it is worth highlighting that, traditionally, these aspects are used to identify candidates for pushbacks. Since the tonnage increments between successive pits do not present significant volume increases, it is not necessary in both cases to apply the Procedure 1 in order to generate intermediate pits.



(a) (b)

Figure 2: (a) Pit by pit graph, and (b) E1750 long section view showing nested pits from MCL model.



(a) (b)

Figure 3: (a) Pit by pit graph, and (b) N4900 long section view showing nested pits from BMT model.

Now, different instances for automatic pushback generation are set: for these, the number of required pushbacks are imposed from 3 to 6, resulting in four instances for each case. Table 2 shows a summary of the parameters. Table 3 shows for each case study and for each instance: average total tonnage (ATT) and objective value, that is, minimization of the mean absolute deviation (MAD) among pushbacks, presenting when it is possible

Table 2: Parameters to generate an automated pushback selection from MCL and BMT with MAD objective function.

PS model parameters	Symbol	Value (MCL)	Value (BMT)
# pits	N	100	90
Total tonnage (Mton)	$\mathtt{rton}_{\mathrm{N0}}$	250.8	250.6
# empty pits	K	12	20
# pushbacks	n_o	3, 4, 5, 6	3, 4, 5, 6

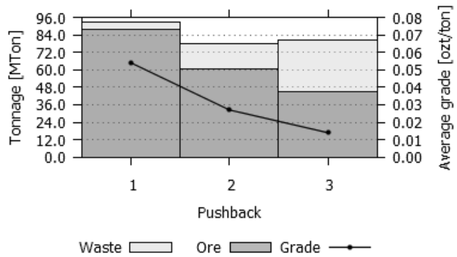
a solution for the gap problem in the pushback selection in a minimum required time, which is shown as well in column *Time*. Additionally, column *Ratio* shows the ratio between mean absolute deviation and average total tonnage, expressed as percentage. For each case, four different results were obtained, one per required number of pushbacks, being denoted as Push-n_o. Default optimality gap of the solver was set to 1%, but all obtained results were optimal (0% gap).

Table 3: Main numerical results. For each case and respective instance are shown: average tonnage, MAD and ratio as deviation measures, and required time to compute the solution.

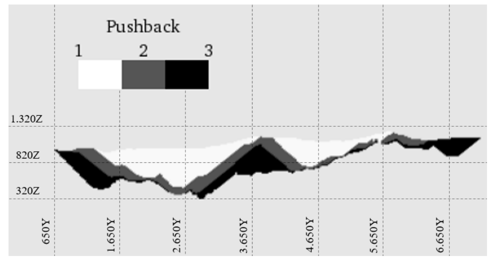
Instance	MCL				BMT			
	ATT (Mton)	MAD (Mton)	Ratio (%)	Time (s)	ATT (Mton)	MAD (Mton)	Ratio (%)	Time (s)
Push-3	83.6	6.1	7.3	2.8	83.5	2.8	3.3	1.3
Push-4	82.7	0.6	0.7	3.6	62.7	1.5	2.4	0.9
Push-5	50.2	1.1	2.2	3.5	50.1	1.7	3.4	1.4
Push-6	41.8	4.1	9.8	5.1	41.8	1.4	3.5	1.4

An important aspect of this methodology is the time required for arriving at the final solution: the nested pit generation was computed in both cases for all instances in less than 60 seconds, while the automated pushback selection was computed in 5 seconds in the most time-consuming instance. This indicates one of the main advantages of this model, which is so fast to compute a pushback selection applying well-defined criteria.

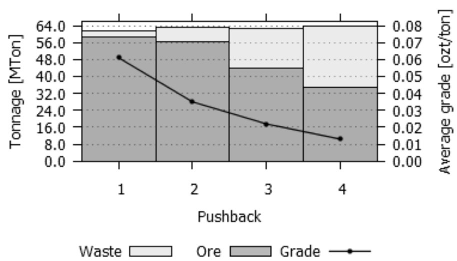
The results for each instance and each case are shown in Figures 4 and 5, presenting ore and waste tonnages, average grade and long section views from each pushback selection. All instances show decreasing both ore tonnage and average grade.



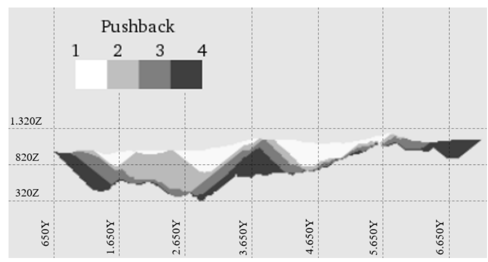
(a) Push-3: Tonnages/grade



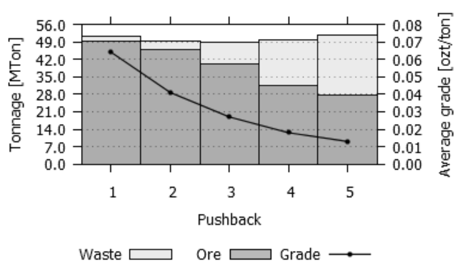
(b) Push-3: E1750 long section view



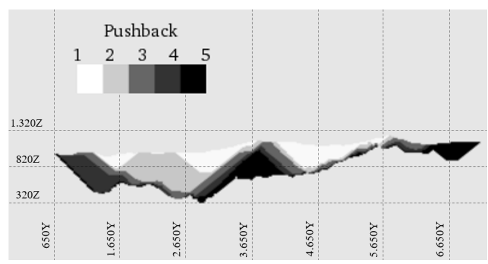
(c) Push-4: Tonnages/grade



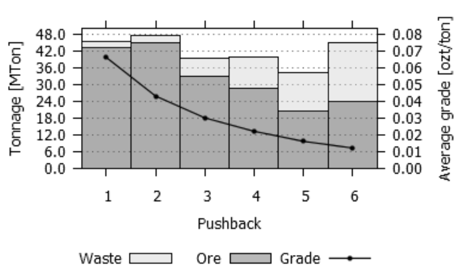
(d) Push-4: E1750 long section view



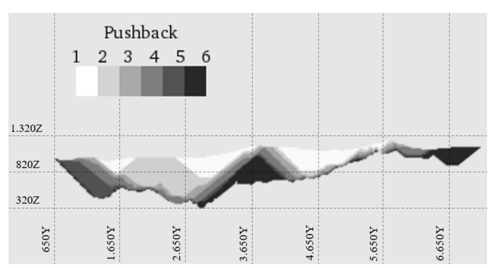
(e) Push-5: Tonnages/grade



(f) Push-5: E1750 long section view

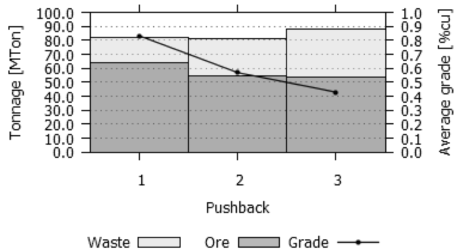


(g) Push-6: Tonnages/grade

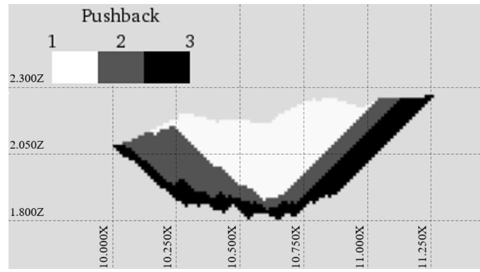


(h) Push-6: E1750 long section view

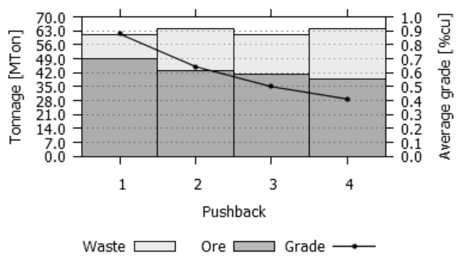
Figure 4: For each instance, both ore-waste tonnages and average grade graph is shown (left side), along with corresponding long section view (right side) for `MCL` case.



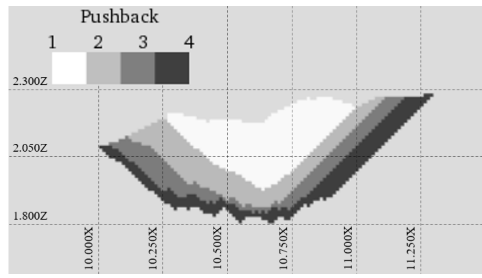
(a) Push-3: Tonnages/grade



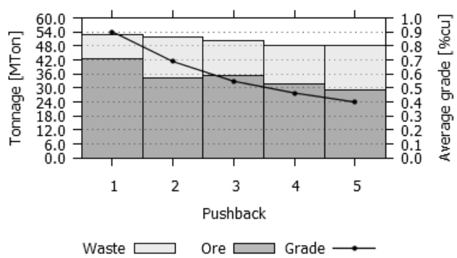
(b) Push-3: E1750 long section view



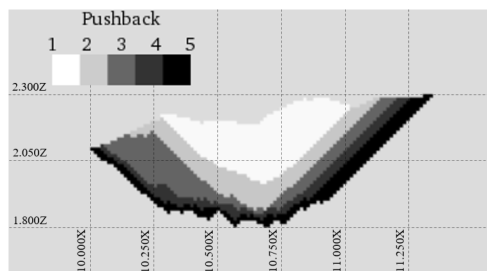
(c) Push-4: Tonnages/grade



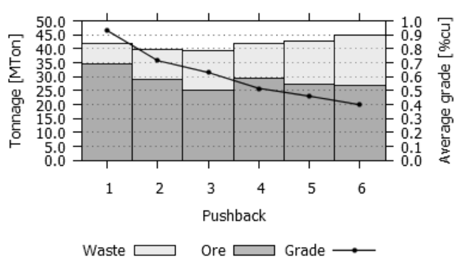
(d) Push-4: E1750 long section view



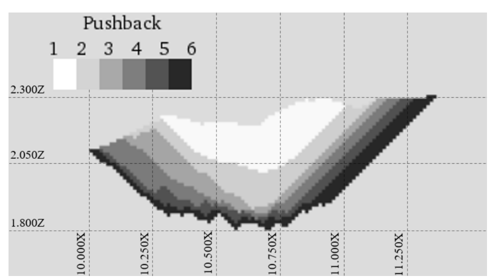
(e) Push-5: Tonnages/grade



(f) Push-5: E1750 long section view



(g) Push-6: Tonnages/grade



(h) Push-6: E1750 long section view

Figure 5: For each instance, both ore-waste tonnages and average grade graph is shown (left side), along with corresponding long section view (right side) for `BMT` case.

An important aspect in pushback design is their role in the production scheduling. Pushbacks strongly influence access to the ore and waste blocks in all periods in order to maximize the net present value (NPV). It is clear that NPV will be higher if ore blocks can be mined during early periods. By construction of nested pits approach as revenue factor increases, so does the amount of waste. Based on these facts, if the gap problem is controlled, an additional result is the desired behavior of stripping ratio: the pushback selection moves from low stripping ratio sectors of the mine towards high stripping ratio sectors, helping to obtain a high NPV in scheduling stage. Tables 4 and 5 show the stripping ratio for the pushback selection obtained from `MCL` and `BMT` cases, respectively. In general, for both cases and most of the instances, the stripping ratio is increasing, from first to last pushback, which indicates the delayed extraction of waste material.

Table 4: Stripping ratio for each pushback and instance from the selection in `MCL` case.

`MCL` instance	Pushback number					
	1	2	3	4	5	6
Push-3	0.05	0.29	0.77	-	-	-
Push-4	0.05	0.11	0.43	0.82	-	-
Push-5	0.05	0.08	0.22	0.58	0.87	-
Push-6	0.04	0.06	0.19	0.39	0.66	0.89

Table 5: Stripping ratio for each pushback and instance from the selection in `BMT` case.

`BMT` instance	Pushback number					
	1	2	3	4	5	6
Push-3	0.28	0.49	0.62	-	-	-
Push-4	0.25	0.49	0.49	0.64	-	-
Push-5	0.24	0.52	0.43	0.52	0.66	-
Push-6	0.21	0.37	0.57	0.43	0.58	0.67

5. Conclusions and future work

In this paper, a new model for automatic selection of pushbacks from a set of nested pits as support and applying a set of well-defined criteria is presented, in the context of long-term open pit mine planning. This model works on a mixed integer programming setting, characterizing properly the family of pushbacks from pits that can be obtained using, but not limited to, standard methods such as parameterization of metal price or similar.

Some important considerations about the nested pits support: for one side, the generation of nested pits has a key impact into automated pushback selection, therefore it is important to be careful in the selection of the *revenue factors* that define the set of pits. For the other side, since the model works on pit support instead of block support, the size of deposit is not a limitation to generate pits and then pushbacks to select from, because the associated programming model has at most $\frac{N(N+1)}{2}$ variables, where N is the number of pits. Note that, when it is possible, the simplifiers constraints allow to reduce the amount of variables.

The proposed approach was applied on two different case studies, showing that it was able to produce quickly alternative sets of pushbacks that minimize the total tonnage differences among them, helping to avoid the so called *gap problem* in pushback design. The main advantages of this new approach are: (i) it gives several and well-defined criteria to select pushbacks from nested pits; therefore, it allows the mine planner to automate this task. (ii) It is very fast and low computational resource consuming to compute an optimal solution, having the choice to assess other interesting criteria defined by the mine planner. (iii) Easy implementation, in the sense that it can be integrated to existing software based on parametrization approaches.

As future work, the characterization of the set of pushbacks from nested pits support should be used as the basis in order to develop a family of models that enable to move forward in new algorithms for pushback generation in a more efficient way. Additionally, to research about alternative supports to nested pits to guarantee good feasible solutions. Another research line should be to incorporate time dimension into pushback selection to ensure a design that reach high NPV in production scheduling, similar to Elkington and Durham (2011) or Jelvez and Morales (2017). Finally, in order to model operational constraints such as minimum pit bottom width, the ideas from Saavedra-Rosas et al. (2016) can be incorporated into nested pit generation.

Acknowledgments

E. Jélvez and N. Morales were supported by CONICYT basal grant FB0809 (Advanced Mining Technology Center), University of Chile. E. Jélvez was also partially supported by Canada-Chile Leadership Exchange Scholarship from Government of Canada.

References

Carlson, T. R., Erickson, J. D., OBrain, D., and Pana, M. T. (1966). Computer techniques in mine planning. *Mining Engineering*, 18(5):53–56.

Chandran, B. G. and Hochbaum, D. S. (2009). A computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem. *Operations research*, 57(2):358–376.

Consuegra, F. A. and Dimitrakopoulos, R. (2010). Algorithmic approach to pushback design based on stochastic programming: method, application and comparisons. *Mining Technology*, 119(2):88–101.

Dagdelen, K. and François-Bongarçon, D. (1982). Towards the complete double parameterization of recovered reserves in open pit mining. In *Proceedings of 17th international APCOM symposium*, pages 288–296.

Dagdelen, K. and Johnson, T. B. (1986). Optimum open pit mine production scheduling by lagrangian parameterization. In *Proceedings of the 19th APCOM Symposium of the Society of Mining Engineers, AIME, New York*, pages 127–141.

Delphos, M. P. L. (2015). Minelink library. http://delphos.dmi.uchile.cl.

Elkington, T. and Durham, R. (2011). Integrated open pit pushback selection and production capacity optimization. *Journal of Mining Science*, 47(2):177–190.

Espinoza, D., Goycoolea, M., Moreno, E., and Newman, A. (2013). Minelib: a library of open pit mining problems. *Annals of Operations Research*, 206(1):93–114.

Francois-Bongarcon, D. and Guibal, D. (1984). Parameterization of optimal designs of an open pitbeginning of a new phase of research. *Trans. SME, AIME*, 274:1801–1805.

Gholamnejad, J. and Osanloo, M. (2007). Incorporation of ore grade uncertainty into the push back design process. *Journal of the South African Institute of Mining & Metallurgy*, 107(3):177.

Goodfellow, R. and Dimitrakopoulos, R. (2013). Algorithmic integration of geological uncertainty in pushback designs for complex multiprocess open pit mines. *Mining Technology*, 122(2):67–77.

Gurobi Optimization, I. (2015). Gurobi optimizer reference manual. http://www.gurobi.com.

Hochbaum, D. S. (2008). The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations research*, 56(4):992–1009.

Hochbaum, D. S. and Chen, A. (2000). Performance analysis and best implementations of old and new algorithms for the open-pit mining problem. *Operations Research*, 48(6):894–914.

Jelvez, E. and Morales, N. (2017). Mip-based procedure to pushback selection. In *Proceedings of the 38th International Symposium on Application of Computers and Operations Research in the Mineral Industry (APCOM). Golden, Colorado, USA*, pages 7.1–7.8.

Kakaie, R. et al. (2012). A new algorithm for optimum open pit design: Floating cone method iii. *Journal of Mining and Environment*, 2(2):118–125.

Lerchs, H. and Grossmann, I. (1965). Optimum design of open-pit mines. *Transactions, C.I.M.*, 68:17–24.

Mansini, R., Ogryczak, W., and Speranza, M. G. (2015). *Linear and Mixed Integer Programming for Portfolio Optimization*. Springer.

Meagher, C., Abdel Sabour, S., and Dimitrakopoulos, R. (2009). Pushback design of open pit mines under geological and market uncertainties. In *Proceedings of the International Symposium on Orebody Modeling and Strategic Mine Planning: Old and New Dimensions in Changing World*, pages 297–304.

Meagher, C., Dimitrakopoulos, R., and Avis, D. (2014a). Optimized open pit mine design, pushbacks and the gap problem: A review. *Journal of Mining Science*, 50(3):508–526.

Meagher, C., Dimitrakopoulos, R., and Vidal, V. (2014b). A new approach to constrained open pit pushback design using dynamic cut-off grades. *Journal of Mining Science*, 50(4):733–744.

Mitchell, S., OSullivan, M., and Dunning, I. (2011). Pulp: a linear programming toolkit for python. *The University of Auckland, Auckland, New Zealand, http://www. optimization-online. org/DB_FILE/2011/09/3178. pdf*.

Pana, M. T. (1965). The simulation approach to open pit design. In *5th APCOM, Symposium on Computers and Computer Applications in Mining and Exploration*, volume 5, pages 1–24. University of Arizona, Tucson, Arizona, USA.

Picard, J.-C. (1976). Maximal closure of a graph and applications to combinatorial problems. *Management science*, 22(11):1268–1272.

Python, S. F. (2015). Python™ Programming language. http://www.python.org.

Ramazan, S. and Dagdelen, K. (1998). A new push back design algorithm in open pit mining. In *Proceedings of 17th MPES conference, Calgary, Canada*, pages 119–124.

Saavedra-Rosas, J., Jelvez, E., Amaya, J., and Morales, N. (2016). Optimizing open-pit block scheduling with exposed ore reserve. *Journal of the Southern African Institute of Mining and Metallurgy*, 116(7):655–662.

Seymour, F. (1995). Pit limit parameterization from modified 3d lerchs-grossmann algorithm. *SME, Preprint Number:95-96.*

Somrit, C. and Dagdelen, K. (2013). Max flow lagrangian based phase design algorithm for open pit production scheduling optimization. In *Proceedings of the 36th International Symposium on Applications of Computers and Operations Research in the Mineral Industry (APCOM). Porto Alegre, Brazil*, pages 401–414.

Tabesh, M., Mieth, C., and Askari-Nasab, H. (2014). A multi–step approach to long–term open–pit production planning. *International Journal of Mining and Mineral Engineering*, 5(4):273–298.

Vallet, R. (1976). Optimisation mathematique de l'exploitation d'une mine a ciel ouvert ou le problem de l'enveloppe. *Annales des Mine de Belgique*, pages 113–135.

Wang, Q. and Sevim, H. (1995). Alternative to parameterization in finding a series of maximum-metal pits for production planning. *Mining Engineering*, 47(2):178–182.

Wright, E. (1999). Moving cone ii-a simple algorithm for optimum pit limits design. In *Proceedings of the 28th Symposium on the application of computers and operations research in the mineral industries (APCOM),(Colorado USA)*, pages 367–374.

Zhao, Y. and Kim, Y. (1992). A new graph theory algorithm for optimal ultimate design. In *Proceedings of the 23rd International Symposium on the Application of Computers and Operations Research in The Mineral Industries (APCOM)*, pages 423–434. AIME Littleton, Co.