A variable neighborhood descent algorithm for the open-pit mine production scheduling problem with metal uncertainty

Amina Lamghari\textsuperscript{a,*}, Roussos Dimitrakopoulos\textsuperscript{a}, Jacques A. Ferland\textsuperscript{b}

\textsuperscript{a} COSMO - Stochastic Mine Planning Laboratory, Department of Mining and Materials Engineering, McGill University, FDA Building, 3450 University Street, Montreal, Quebec, H3A 2A7, Canada

\textsuperscript{b} Department of Computer Science and Operations Research, University of Montreal, C.P. 6128, SUCC. Centre-Ville, Montreal, Quebec, H3C 3J7, Canada

Abstract

Uncertainty is an inherent aspect of the open-pit mine production scheduling problem (MPSP); however, little is reported in the literature about solution methods for the stochastic versions of the problem. In this paper, two variants of a variable neighborhood descent algorithm for solving the MPSP with metal uncertainty are proposed. The proposed methods are tested and compared on actual large-scale instances and show an average deviation of less than 3\% from optimality, while are obtained within a few minutes up to a few hours.

Keywords: Stochastic production scheduling, open-pit mining, uncertainty, heuristics.

1. Introduction

Production scheduling of open-pit mining operations is a challenging and critical issue for mining companies, a key factor in determining returns

*Corresponding author, Tel.: (514) 398-5841, Fax: (514) 398-7099
Email addresses: amina.lamghari@mail.mcgill.ca (Amina Lamghari), roussos.dimitrakopoulos@mcgill.ca (Roussos Dimitrakopoulos), ferland@iro.umontreal.ca (Jacques A. Ferland)
on investments in the order of hundreds of millions of dollars. In scheduling mine production, the mineral deposit is represented as a three-dimensional array of blocks. Each block has a weight and a metal content estimated using information obtained from drilling. To recover the metal, the block is first mined from the ground and then processed in a mill. These operations are termed mining and processing, respectively.

Blocks are classified as ore or waste according to their metal content. Ore blocks are those that have a selling revenue greater than their processing costs, while waste blocks have a total metal content whose selling revenue is less than the processing costs. Both ore and waste blocks must be mined in order to gain access to all the ore blocks. Any block that must be mined in order to reach another block is called the predecessor of the second block.

Decisions on block scheduling are subject to various types of constraints. The production schedule not only must respect the limits on extraction capacity (mining constraints) and the capacity of the processor (processing constraints) at each period of the life of the mine, but also must take into consideration the order in which blocks can be removed from the orebody to ensure that a block is not mined before any of its predecessors (slope constraints). Additionally, any block can be mined only once (reserve constraints). The problem is to determine which blocks to extract and when to extract them (mining sequence) in order to maximize the net present value (NPV) of the mine while respecting the various constraints.

A major complexity in the open-pit mine production scheduling problem is that the number of blocks is large, in general in the order of tens to hundreds of thousands, yielding a large-scale optimization problem. Another factor adding to the complexity of the mine scheduling problem is metal uncertainty, for the metal content of the blocks is not known precisely at the
time decisions are made but is inferred from limited drilling information. Up to now, most methods developed to solve the MPSP either ignore the metal uncertainty issue or are not able to solve large-scale instances in which metal uncertainty is accounted for.

Indeed, the MPSP has been frequently studied since the 1960s (Newman et al., 2010). Different methods have been applied to solve the deterministic version of the problem, which assumes that all the problem parameters are well known. These methods can be classified into three categories: exact methods (Dagdelen and Johnson, 1986; Caccetta and Hill, 2003; Ramazan, 2007; Boland et al., 2009; Bley et al., 2010), heuristic and metaheuristic methods (Gershon, 1987; Denby and Schofield, 1994; Ferland et al., 2007; Chatterjee et al., 2010), and hybrid methods (Tolwinski and Underwood, 1996; Sevim and Lei, 1998; Moreno et al., 2010). However, the uncertain nature of the problem is ignored in the deterministic version of the MPSP, resulting in misleading assessments (Ravenscroft, 1992; Dowd, 1994; Dimitrakopoulos et al., 2002; Godoy and Dimitrakopoulos, 2004). Studies that compare stochastic to deterministic approaches (Godoy and Dimitrakopoulos, 2004; Menabde et al., 2007; Albor and Dimitrakopoulos, 2009, 2010; Asad and Dimitrakopoulos, 2013)) indicate that stochastic approaches show major improvements in net present value, in the order of 20% to 30%, substantially reduce risk in meeting production forecasts, and find pit limits larger than the ones found by deterministic approaches, contributing to the sustainable utilization of mineral resources.

Stochastic versions of the problem, which are more realistic and more relevant, have been considered by an increasing number of researchers in the last decade. Usually, a scenario approach is used to handle the metal uncertainty. The scenarios are specified from estimates in the continuous space.
without any underlying scenario tree structure. Consequently, the multistage approach commonly used in stochastic programming cannot be used. Three different approaches are used in the literature. The first involves formulations maximizing the expected NPV over the scenarios describing the metal uncertainty while satisfying the production targets in an average sense (Menabde et al., 2007). The second involves formulations maximizing the expected NPV and minimizing deviations from production targets for each individual scenario (Ramazan and Dimitrakopoulos, 2007; Albor and Dimitrakopoulos, 2010; Ramazan and Dimitrakopoulos, 2012). The third approach, illustrated in Boland et al. (2008), takes into account the metal uncertainty via a multi-stage stochastic programming approach where the missing interdependency between scenarios and the decisions (i.e., the missing tree structure) is simulated using conditional non-anticipativity constraints.

While different stochastic models have been developed, solution methods have received relatively less attention. Furthermore, most of the solution methods developed have been able to deal only with instances of relatively small size (typically, up to 20,000 blocks). The stochastic models proposed in the studies by Menabde et al. (2007), Ramazan and Dimitrakopoulos (2007, 2012), and Boland et al. (2008) are solved using the mixed integer programming solver CPLEX. The method used in Albor and Dimitrakopoulos (2010) consists of generating a set of nested pits, grouping these pits into pushbacks, and then generating a schedule based on the pushback designs obtained, while the method used in Lamghari and Dimitrakopoulos (2012) is based on Tabu search.

In this paper, we propose an efficient solution method to deal with large instances of the MPSP with metal uncertainty, where the uncertainty is addressed using a two-stage stochastic programming approach. Specifically, we
introduce a metaheuristic method based on a variable neighborhood descent procedure (Hansen and Mladenovic, 2001). To generate the initial solution to be improved by this procedure, we consider two different alternatives. Both are based on a decomposition approach separating the problem into a series of sub-problems, each associated with one period. They differ in the method used to solve the sub-problems. In the first alternative, the sub-problems are solved exactly, while in the second one, the sub-problems are solved approximately with a greedy heuristic. We evaluate and compare the performance of the two variants of the proposed solution method on large-scale instances with up to 97,307 blocks. The results indicate that both variants generate very good solutions in reasonable computational times. The first variant, where the sub-problems are solved exactly, slightly outperforms the second one in terms of solution quality, while the second variant, where the sub-problems are solved with a heuristic, requires in general significantly less computational time.

The remainder of the paper is organized as follows: In Section 2, the approach used to deal with metal uncertainty is outlined, and a mathematical formulation of the problem is introduced. The following section briefly describes the methods used to generate the initial solution. Section 4 summarizes the variable neighborhood procedure used to improve the initial solution. Computational results on real-life data are reported and discussed in Section 5. Finally, some conclusions are drawn in Section 6.

2. Mathematical formulation

Referring to the description given in the previous section, the problem can be formulated as a two-stage stochastic programming model (Birge and
Louveaux, 1997). In the first stage, one determines a set of blocks to be mined at each period in order to satisfy the reserve constraints, the slope constraints, and the mining constraints. The metal content of each block, determining whether the block is ore or waste, is uncertain at this stage. In the second stage, when the blocks are mined, the metal content becomes known. In some periods, the ore blocks available, requiring processing, may have a total weight exceeding the mill capacity (i.e., the processing constraints may be violated). To deal with such a situation, some recourse actions are available, but they induce a cost. The problem is then to identify a schedule that maximizes the expected net present value of the mining operation minus the expected recourse costs incurred due to the violation of the processing constraints.

The following notation is used to formulate the mathematical model:

- \( T \): the number of periods over which blocks are being scheduled (horizon).
- \( t \): period index, \( t = 1, \ldots, T \).
- \( W^t \): maximum amount of rock (ore and waste) that can be mined during period \( t \) (mining capacity in tons).
- \( \Theta^t \): maximum amount of ore that can be processed in the mill during period \( t \) (processing capacity in tons).
- \( N \): the number of blocks considered for scheduling.
- \( i \): block index, \( i = 1, \ldots, N \).
- \( P_i \): the set of predecessors of block \( i \); i.e., blocks that have to be mined to have access to block \( i \).
• $\overline{P}_i$: the set of direct predecessors of block $i$; i.e.,

$$\overline{P}_i = \{ p \in P_i : p \text{ is on the level immediately above } i \}.$$  

• $\Gamma_i$: the set of successors of block $i$; i.e., $\gamma \in \Gamma_i$ if $i \in P_\gamma$.

Figure 1 gives a 2-dimensional illustration of the sets $P_i$, $\overline{P}_i$, and $\Gamma_i$.

Figure 1: Illustration of the sets $P_i$, $\Gamma_i$, and $\overline{P}_i$

• $w_i$: the weight of block $i$ (in tons).

• $v_i$: the economic value of block $i$. It is a random variable depending on the metal content of the block.

• $v_i^t = \frac{v_i}{(1 + d_1)^t}$: the discounted economic value of block $i$ if mined during period $t$ ($d_1$ being the discount rate per period). Note that it is assumed that ore blocks are processed during the same period when they are mined and that the profit is also generated during that period.

• $\theta_i$: a random variable indicating if block $i$ is an ore or a waste block

$$\theta_i = \begin{cases} 
1 & \text{if block } i \text{ is an ore block,} \\
0 & \text{otherwise.}
\end{cases}$$
It is a random variable because the metal content of a block determines whether the block is ore or waste.

• \( c^t = \frac{c}{(1 + d_2)^t} \): unit surplus cost incurred if the total weight of ore blocks mined during period \( t \) exceeds the processing capacity, \( \Theta^t \) (\( c \) being the undiscounted surplus cost, and \( d_2 \) represents the risk discount rate).

The variables used to formulate the problem are as follows:

• A binary variable is associated with each block \( i \) for each period \( t \):
  \[ x^t_i = \begin{cases} 1 & \text{if block } i \text{ is mined by period } t, \\ 0 & \text{otherwise.} \end{cases} \]
  This means that if block \( i \) is mined in period \( \tau \), then \( x^t_i = 0 \) for all \( t = 1, \ldots, \tau - 1 \) and \( x^t_i = 1 \) for all \( t = \tau, \ldots, T \). If \( i \) is not mined during the horizon, then \( x^t_i = 0 \) for all \( t = 1, \ldots, T \).

• A random variable \( d^t \) is associated with each period \( t \). It measures the surplus in ore mined during period \( t \).

The mathematical model can be summarized as follows:

\[
\text{max } E\left[\sum_{i=1}^{N} v^t_i x^t_i\right] + E\left[\sum_{t=2}^{T} \sum_{i=1}^{N} v^t_i (x^t_i - x^{t-1}_i)\right] - E\left[\sum_{i=1}^{T} c^t d^t\right]
\]  

\( (M) \)  

Subject to
\[ x_{i}^{t-1} \leq x_{i}^{t} \quad i = 1, \ldots, N, \ t = 2, \ldots, T \]  
\[ x_{i}^{t} \leq x_{p}^{t} \quad i = 1, \ldots, N, \ p \in \mathcal{P}_{i}, \ t = 1, \ldots, T \]  
\[ \sum_{i=1}^{N} w_{i} x_{i}^{1} \leq W^{1} \]  
\[ \sum_{i=1}^{N} w_{i}(x_{i}^{t} - x_{i}^{t-1}) \leq W^{t} \quad t = 2, \ldots, T \]  
\[ \sum_{i=1}^{N} \theta_{i} w_{i} x_{i}^{1} - d^{1} \leq \Theta^{1} \]  
\[ \sum_{i=1}^{N} \theta_{i} w_{i}(x_{i}^{t} - x_{i}^{t-1}) - d^{t} \leq \Theta^{t} \quad t = 2, \ldots, T \]  
\[ x_{i}^{t} = 0 \text{ or } 1 \quad i = 1, \ldots, N, \ t = 1, \ldots, T \]  
\[ d^{t} \geq 0 \quad t = 1, \ldots, T. \]

The objective function (1) includes two terms to maximize the expected net present value of the mining operation and to minimize the expected recourse costs incurred whenever the processing constraints are violated due to metal uncertainty. Constraints (2) guarantee that each block \( i \) is mined at most once during the horizon (reserve constraints). The mining precedence (slope constraints) is enforced by constraints (3). Constraints (4) and (5) impose an upper bound \( W^{t} \) on the amount of rock (ore and waste) mined during each period \( t \) (mining constraints). Constraints (6) and (7) are related to the requirements on the processing levels (processing constraints). The target is to have the total amount of ore mined during any period \( t \) be smaller than \( \Theta^{t} \); otherwise, the surplus penalty cost is equal to \( c^{t} d^{t} \). The model is a two-stage stochastic programming model. The variables \( x_{i}^{t} \) specifying the mining sequence are the first-stage decision variables, and the random variables \( d^{t} \) measuring the surplus in ore production are the second-stage decision variables. \( d^{t} \) depend on both the realization of the metal content.
and the first-stage decisions.

To transform this stochastic model into an equivalent deterministic one, assume that \( S \) possible scenarios are available where each scenario \( s \) specifies the metal content of each block. Furthermore, assume that the probability of occurrence of scenario \( s \) is \( \pi_s \), with \( \sum_{s=1}^{S} \pi_s = 1 \). Let \( v_{1s}^i \), \( \theta_i \), and \( d_s^t \) denote respectively a realization of the random variables \( v_1^i \), \( \theta_i \), and \( d^t \). Then, the original model (1)–(9) can be reformulated as follows:

\[
\max \sum_{i=1}^{N} \sum_{s=1}^{S} \pi_s v_{1s}^i x_1^i + \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{s=1}^{S} \pi_s v_t^i (x_t^i - x_{t-1}^i) - \sum_{t=1}^{T} \sum_{s=1}^{S} \pi_s c^t d_s^t \quad (10)
\]

\((DE)\)

Subject to

\[
\sum_{i=1}^{N} \theta_{is} w_i x_1^i - d_1^s \leq \Theta^1 \quad s = 1, \ldots, S \quad (11)
\]

\[
\sum_{i=1}^{N} \theta_{is} w_i (x_t^i - x_{t-1}^i) - d_s^t \leq \Theta^t \quad t = 2, \ldots, T, \quad s = 1, \ldots, S \quad (12)
\]

\[
d_s^t \geq 0 \quad t = 1, \ldots, T, \quad s = 1, \ldots, S. \quad (13)
\]

In practice, the proposed model utilizes a limited number of scenarios, each specifying the metal content of each block, which then is used to calculate the corresponding economic value. A natural question to address is how to choose the number of scenarios to consider. In general, this is a well-studied topic in stochastic optimization (Dupacova et al., 2003) and other fields (Scheidt and Caers, 2009). The case studies presented in this paper use 20 scenarios because past work, such as the work in Albor and Dimitrakopoulos (2009), indicates that after about 15 simulated representations
of an orebody, stochastic schedules converge to a stable final physical schedule as well as stable forecasts of production performance. This behavior is not surprising; while simulated scenarios represent a mineral deposit at the support-scale of mining blocks with volume of a few cubic meters, a mine’s production schedule represents a grouping of a few thousand of these mining blocks in just one time (mining) period under various constraints. Thus, as the support-scale of a mine’s schedule is orders of magnitude larger than that of the simulated representations of the mineral deposit being scheduled, after a relatively small number of scenarios stochastic schedule becomes insensitive to additional ones.

In this paper, the set of sufficient scenarios is provided for the objective of designing an efficient method to solve the proposed mathematical model.

If the *mining* and the *processing constraints* are eliminated, and if the scheduling horizon reduces to a single period, then the model (DE) reduces to the classical maximum closure problem. This problem is reducible to the minimum-cut problem (Picard, 1976), and thus it can be solved efficiently using any of the known polynomial maximum-flow algorithms. However, as Bienstock and Zuckerberg (2010) note on page 3, "it can be shown by reduction from max clique that adding a single cardinality constraint to a maximum closure problem is enough to make it NP-hard." Because the MPSP is more complex than a constrained maximum closure problem, and as real-world MPSP instances are very large, having typically tens to hundreds of thousands blocks, it is most likely not appropriate to solve these large-scale realistic instances using an exact method. Instead, a metaheuristic solution method is proposed herein where an initial feasible solution is first obtained and then improved with a variable neighborhood descent (VND) procedure. The methods used to generate the initial solution as well as the
VND procedure are described in the following sections.

3. Generating the initial solution

We propose to generate the initial solution using two different heuristics based on a decomposition approach where the global problem is divided into smaller sub-problems, each associated with a period \( t (t = 1, \ldots, T) \). The sub-problems are solved sequentially in increasing order of \( t \), and their solutions are combined to generate the initial solution. The heuristics differ in the method used to deal with the sub-problems. In the first one, the sub-problems are solved exactly, while in the second one, the sub-problems are solved approximately with a greedy heuristic. Note that since the sub-problems are smaller than the global problem, an exact method can be used to solve them.

3.1. Sub-problem formulation

The sub-problem associated with period \( t \) consists of determining a set of blocks \( B^t \) to be mined in period \( t \). Let us denote by \( R^t = \{ \text{block } i : x^t_i = 0 \} \) the set of blocks not mined yet (if \( t = 1 \), then \( |R^1| = N \); otherwise, \( |R^t| = N - |\bigcup_{\tau < t} B^\tau| \)). In order to satisfy the reserve constraints, the blocks to be included in \( B^t \) should be selected from \( R^t \). The sub-problem associated with period \( t \) can then be summarized as follows:

\[
\begin{align*}
\text{max} & \sum_{i \in R^t} \sum_{s=1}^S \pi_s v^t_{is} y_i - c^t \sum_{s=1}^S \pi_s d_s \\
\text{(SP^t)} & \text{Subject to}
\end{align*}
\]
\[
y_i \leq y_p \quad i \in \mathcal{R}^t, \ p \in \mathcal{P}_i \cap \mathcal{R}^t
\]
\[
\sum_{i \in \mathcal{R}^t} w_i y_i \leq W^t
\]
\[
\sum_{i \in \mathcal{R}^t} \theta_{is} w_i y_i - d_s \leq \Theta^t \quad s = 1, \ldots, S
\]
\[
y_i = 0 \text{ or } 1 \quad i \in \mathcal{R}^t
\]
\[
d_s \geq 0 \quad s = 1, \ldots, S
\]

where

\[
y_i = \begin{cases} 
1 & \text{if block } i \text{ is included in the set } \mathcal{B}^t \text{ (i.e., is mined in period } t), \\
0 & \text{otherwise.}
\end{cases}
\]

Recall that if block \( i \) is mined in period \( t \), then \( x_i^\tau = 0 \) for all \( \tau = 1, \ldots, t - 1 \) and \( x_i^t = 1 \) for all \( \tau = t, \ldots, T \).

Logical implications of the constraints are used to generate valid inequalities to strengthen the formulation above and make the sub-problems easier to solve. To be more specific, consider any block \( i \in \mathcal{R}^t \). On the one hand, the slope constraints require that to include \( i \) in \( \mathcal{B}^t \), we must also include all blocks \( j \in \mathcal{N}_i = \mathcal{P}_i \cap \mathcal{R}^t \) (the set of blocks that are predecessors of \( i \) and not mined yet). On the other hand, \( i \) should not be included in \( \mathcal{B}^t \) if \( w_i + \sum_{j \in \mathcal{N}_i} w_j > W^t \) because this would lead to violation of the mining constraints. Hence, we add the following constraints to the model \( (SP^t) \):

\[
y_i \leq e_i \quad i \in \mathcal{R}^t
\]

where \( e_i \) is a parameter equal to 1 if the extraction of block \( i \) will not lead to violation of the mining constraints and 0 otherwise; i.e.,

\[
e_i = \begin{cases} 
1 & \text{if } w_i + \sum_{j \in \mathcal{N}_i} w_j \leq W^t, \\
0 & \text{otherwise.}
\end{cases}
\]
3.2. Sub-problem solution methods

Two different methods are introduced to solve the sub-problems. In the first method, the formulation (14)–(20) is solved using the branch-and-cut algorithm (BC) implemented in the mixed integer programming solver CPLEX.

The second method is a sequential greedy heuristic procedure (GH) where at each iteration we try to include in the set \( \mathcal{B}^t \) an inverted cone formed by a "base" block in \( \mathcal{R}^t \) and all its predecessors not mined yet. Let us analyze a typical iteration.

Let \( \mathcal{V}^t = W^t - \sum_{b \in \mathcal{B}^t} w_b \) and \( \mathcal{O}_s^t = \max\{\Theta_s^t - \sum_{b \in \mathcal{B}^t} \theta_{bs} w_b, 0\} \) be the residual mining capacity and the residual processing capacity under scenario \( s \), respectively. For each block \( i \in \mathcal{R}^t \), we denote:

- \( \Omega_i = \{i\} \cup \{j : j \in \mathcal{N}_i\} \) the set formed by block \( i \) and all its unmined predecessors (i.e., the inverted cone whose base is \( i \)).
- \( \alpha_i = \sum_{k \in \Omega_i} w_k \): the total weight of blocks in \( \Omega_i \).
- \( \beta_{is} = \sum_{k \in \Omega_i} \theta_{ks} w_k \): the total weight of ore blocks in \( \Omega_i \) under scenario \( s \).
- \( \gamma_i = \sum_{k \in \Omega_i} \sum_{s=1}^S \pi_s v_{ks}^t \): the total expected discounted economic value of blocks in \( \Omega_i \).
- \( \xi_i = \gamma_i - c^t \sum_{s=1}^S \pi_s \max\{\beta_{is} - \mathcal{O}_s^t, 0\} \): the total contribution of blocks in \( \Omega_i \) to the objective function (14).

Consider the set \( \mathcal{A} = \{i \in \mathcal{R}^t : \alpha_i \leq \mathcal{V}^t\} \) of blocks \( i \in \mathcal{R}^t \) whose weight plus the weight of their predecessors not mined yet does not exceed the residual
mining capacity. Clearly, only cones having blocks in $\mathcal{A}$ as their base can be added to $\mathcal{B}^t$ while satisfying the slope constraints and the mining constraints. Select the block $i^* \in \mathcal{A}$ maximizing the value of $\xi_i$. Ties are broken up randomly. Remove all blocks $k \in \Omega_{i^*}$ from $\mathcal{R}^t$, add them to $\mathcal{B}^t$, and update $\mathcal{V}_t^s$, $\mathcal{O}_s^t$ for each $s = 1, \ldots, S$, and the set $\mathcal{A}$. This process is repeated until the mining constraints are approximately satisfied; i.e., until

$$\sum_{i \in \mathcal{B}^t} w_i \leq \delta W^t$$

where $\delta$ is a random number in the interval $[\delta_1, \delta_2]$, and $\delta_1$ and $\delta_2$ are parameters of the procedure in $[0, 1]$.

Note that choosing blocks along with their predecessors (an inverted cone at each iteration) allows a look ahead feature generating better solutions than the myopic approach of choosing blocks one by one.

4. Improving the initial solution

As mentioned before, the initial solution $x = \bigcup_{t=1}^T \mathcal{B}^t$ is improved by applying an adaptation of the Variable Neighborhood Descent method (VND) proposed by Hansen and Mladenovic (2001). The basic idea of VND is to combine different descent heuristics based on different neighborhood structures to escape from local optima. In the following, we first describe the neighborhood structures used in our adaptation of the VND method. Next, we outline the procedure used to improve the solution $x$.

4.1. Neighborhood structures

Three neighborhood structures are used in our adaptation of the VND method. The first structure tries to swap two blocks scheduled in consecutive periods. For example, a waste block scheduled to be mined in period 1 and
having no successors scheduled in period 1 could be left behind to be mined in period 2. In its place, an ore block scheduled to be mined in period 2 could be mined in period 1 provided that its predecessors are scheduled in period 1. The mining capacity should not be exceeded in either period as a result of the swap. The second and third structures try to make room for new blocks in a given period by scheduling some blocks to be mined in that period backward or forward while satisfying the slope and the mining constraints. A more formal description of the three neighborhood structures as well as an outline of the strategy used to explore them is given below.

- **N1 (Exchange or Swap):** Let i and j be two blocks mined in periods t and (t + 1), respectively. An exchange consists of replacing \( B^t \) and \( B^{t+1} \) by \( (B^t - \{i\}) + \{j\} \) and \( (B^{t+1} - \{j\}) + \{i\} \), respectively. The exchange of two blocks is feasible if the resulting solution is feasible; i.e., only if it satisfies the slope and the mining constraints. Figure 2 gives a 2-dimensional illustration of an exchange move involving two blocks, i and j, with \( T = 2 \).

![Figure 2: Exchange move between blocks i and j with T = 2. The grey area represents the set of blocks to be mined in the first period (B^1), while the white area delimited by the thick lines represents the set of blocks to be mined in period 2 (B^2).](image)
• $N^2$ (Shift-after): Let $i$ be a block mined in period $t$, and let $\mathcal{I} = \{i\} \cup \{\text{block } \gamma : \gamma \in \Gamma_i \cap B^t\}$ denote the set including $i$ and its successors mined in the same period. A shift-after consists of replacing $B^t$ and $B^{t+1}$ by $B^t - \mathcal{I}$ and $B^{t+1} + \mathcal{I}$, respectively. Clearly, the slope constraints are satisfied in the resulting solution since the blocks are moved along with their successors. However, the mining constraints must be satisfied in period $(t + 1)$ in order to allow this shift-after. Figure 3 illustrates the Shift-after move where block $i$ and its successors mined in period 1 are moved to period 2.

![Figure 3: Shift-after move of block $i$ and its successors mined in the same period. The grey area represents the set of blocks to be mined in the first period ($B^1$), while the white area delimited by the thick lines represents the set of blocks to be mined in period 2 ($B^2$)](image)

• $N^3$ (Shift-before): Let $i$ be a block mined in period $t$, and let $\mathcal{I} = \{i\} \cup \{\text{block } p : p \in \mathcal{P}_i \cap B^t\}$ denote the set including $i$ and its predecessors mined in the same period. A shift-before consists of replacing $B^t$ and $B^{t-1}$ by $B^t - \mathcal{I}$ and $B^{t-1} + \mathcal{I}$, respectively. As for the Shift-after neighborhood, the slope constraints are necessarily satisfied, but the mining constraints in period $(t - 1)$ must be satisfied in order to allow this shift-before. Figure 4 illustrates the Shift-before move where block
and its predecessors mined in period 2 are moved to period 1.

Figure 4: *Shift-before* move of block $i$ and its predecessors mined in the same period. The grey area represents the set of blocks to be mined in the first period ($\mathcal{B}^1$), while the white area delimited by the thick lines represents the set of blocks to be mined in period 2 ($\mathcal{B}^2$)

Note that the operators *Shift-after* and *Shift-before* motivate the introduction of constraints (21) when applying the greedy heuristic procedure (GH) to generate the initial solution. Indeed, if the value of $\delta$ were fixed to 1 at each iteration, then it would be more difficult to find a feasible neighbor solution when applying these operators.

The strategy to explore the first neighborhood $N^1$ is as follows: Periods $t = 1, \ldots, (T - 1)$ are considered sequentially in increasing order. Given a period $t$, all feasible exchanges involving pairs of blocks mined in $t$ and $(t + 1)$ are systematically considered. The best exchange is selected. We apply the selected exchange if it leads to a better solution or to a solution of the same value as the current solution (in order to escape from the current local optimum). The process is iterated with the new solution. When no feasible exchange exists to further improve the solution or to get a solution of equal value, the next period $(t + 1)$ is considered; i.e., exchanges involving pairs of blocks mined in periods $(t + 1)$ and $(t + 2)$ are evaluated.
A similar exploration strategy is used when considering the Shift-after neighborhood $N^2$ except that periods $t = 1, \ldots, (T - 1)$ are considered in decreasing order. For the Shift-before neighborhood $N^3$, periods $t = 2, \ldots, T$ are considered in increasing order.

### 4.2. Variable neighborhood descent procedure

The rules of a basic VND are applied. Start by exploring the Exchange neighborhood ($N^1$). When the search of $N^1$ is completed (i.e., for all $t = 1, \ldots, (T - 1)$, no feasible exchange between pairs of blocks mined in periods $t$ and $(t + 1)$ exists to further improve the solution or to get a solution of equal value), restart a new search using the Shift-after neighborhood ($N^2$). Once the search of $N^2$ is completed, if the solution has been improved, return to $N^1$; otherwise, use the Shift-before neighborhood ($N^3$). This process terminates when no move in any of the three neighborhoods improves the value of the objective function. Note that the order in which the neighborhoods are explored follows from preliminary tests.

### 5. Numerical results

Numerical tests were completed on three different problems based on actual instances from different mines. Problem $P_1$ is generated from an actual copper deposit where all blocks $i$ are $20 \times 20 \times 10$ meters in size and weigh $w_i = 10,800$ tons each. Problem $P_2$ is from an actual gold deposit where all blocks $i$ are $15 \times 15 \times 10$ meters in size and weigh $w_i = 5,625$ tons each. Finally, problem $P_3$ is also from a gold deposit but where blocks $i$ are $10 \times 10 \times 5$ meters in size and weigh $w_i = 1,250$ tons each. The economic parameters used to compute the block economic values and the recourse costs are also based on real-life data, and they are summarized in Table 1.
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Copper (P₁)</th>
<th>Gold (P₂ and P₃)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mining cost</td>
<td>$1/t</td>
<td>$1/t</td>
</tr>
<tr>
<td>Processing cost</td>
<td>$9/t</td>
<td>$15/t</td>
</tr>
<tr>
<td>Metal price</td>
<td>$2/lb</td>
<td>$900/oz</td>
</tr>
<tr>
<td>Selling cost</td>
<td>$0.3/lb</td>
<td>$7/oz</td>
</tr>
<tr>
<td>Undiscounted surplus cost</td>
<td>$15/t</td>
<td>$17/t</td>
</tr>
<tr>
<td>Discount rate (d₁)</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>Risk discount rate (d₂)</td>
<td>10%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 1: Economic parameters used to compute the objective function coefficients

Each problem is characterized by a number of blocks \( N \) and a number of periods \( T \), specified in Table 2. Note that \( N \) corresponds to the number of blocks within the pit limits obtained by solving the maximum closure problem (to maximize the expected profit from the mining operation but accounting only for the slope constraints). Each period is one year long, as this is the case in practice and in the literature related to the MPSP. The number of periods is set to \( T = \lceil \sum_{i=1}^{N} w_i / 22,300,000 \rceil \). The number of scenarios \( S \) is equal to 20. The scenarios are equiprobable (i.e., \( \pi_s = \frac{1}{S} \forall s = 1, \ldots, S \)), and they were generated from a limited amount of drilling information using the geostatistical techniques of conditional simulation (Goovaerts, 1997; Scheidt and Caers, 2009; Boucher and Dimitrakopoulos, 2012). These techniques can be seen as a complex Monte Carlo simulation framework. They reproduce all available data and information as well as spatial statistics of the data. Note that some authors use only 5 to 10 scenarios to model metal uncertainty (Menabde et al., 2007; Boland et al., 2008).

Additional parameters to specify the problems are as follows: For each problem, the production capacities are identical in all periods and emulate those in real-world problems. For each period \( t \), the mining capac-
ity $W^t$ is set to $\lceil 1.20 \frac{\sum_{i=1}^{N} w_i}{T} \rceil$ (i.e., the total amount of rock per period plus a margin of 20%). The processing capacity $\Theta^t$ is set to $\lceil 1.05 \frac{\sum_{i=1}^{N} \sum_{s=1}^{S} \pi_{is} w_i \theta_{is}}{T} \rceil$ (i.e., the total expected amount of ore per period plus a margin of 5%).

<table>
<thead>
<tr>
<th>Problem</th>
<th>Number of blocks ($N$)</th>
<th>Number of periods ($T$)</th>
<th>Number of scenarios ($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>Block size: $20 \times 20 \times 10$ m</td>
<td>26,021</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Block weight: $w_i = 10,800$ tons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_2$</td>
<td>Block size: $15 \times 15 \times 10$ m</td>
<td>40,762</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Block weight: $w_i = 5,625$ tons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_3$</td>
<td>Block size: $10 \times 10 \times 5$ m</td>
<td>97,307</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Block weight: $w_i = 1,250$ tons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Characteristics of the problems used in the numerical experiments

All the numerical experiments were performed on an Intel(R) Xeon(R) CPU X7350 computer (2.93 GHz) with 64 GB of RAM running under Linux. Version 12.2 of CPLEX was used to solve the mathematical model ($SP^t$) introduced in Section 3.1. To fix the values of the parameters of the greedy heuristic in Section 3.2, some preliminary numerical experimentation were completed with problem $P_2$ and a set of other four smaller problems. We considered 15 values for the interval $[\delta_1, \delta_2]$ ($[0.3, 0.35]$, $[0.35, 0.4]$, ..., $[0.95, 1]$ and $[1, 1]$). The best results are obtained with the interval $[0.9, 0.95]$. Hence, we complete the rest of the numerical tests using these values for $\delta_1$ and $\delta_2$.

The linear relaxation of the model ($DE$) in Section 2 was solved using...
CPLEX 12.2 to obtain an upper bound $Z_{LR}$ on the optimal value, which is used to assess the quality of the solutions produced with the two variants of the proposed solution method, denoted **BC-VND** and **GH-VND**.

Since **GH** and **VND** include random choices, each problem was solved 10 times. The results are summarized in Table 3, where

- **%Min Gap** = $\frac{Z_{LR} - Z_{best}}{Z_{LR}} \times 100$: the value of the relative gap between the value $Z_{best}$ of the best solution obtained by the variants in the 10 runs and the optimal value $Z_{LR}$ of the linear relaxation.

- **%Max Gap** = $\frac{Z_{LR} - Z_{worst}}{Z_{LR}} \times 100$: the value of the relative gap between the value $Z_{worst}$ of the worst solution obtained by the variants in the 10 runs and the optimal value $Z_{LR}$ of the linear relaxation.

- **%Ave Gap** = $\frac{Z_{LR} - Z_{average}}{Z_{LR}} \times 100$: the value of the relative gap between the average value $Z_{average}$ of the 10 solutions obtained by the variants and the optimal value $Z_{LR}$ of the linear relaxation.

- **Ave CPU**: the average solution time in minutes.

The first three criteria do not apply to **CPLEX** as indicated by "-". The last criterion indicates also the CPU time required by **CPLEX** to solve the linear relaxation of the problems.

The following observations can be derived from Table 3:

- The two variants are very efficient in the sense that for each problem the value of **%Min Gap** is less than 5% away from the upper bound provided by **CPLEX**.

- The length of the interval [**%Min Gap**, **%Max Gap**] indicates the robustness of the variants. In all cases (considering the 3 problems and
<table>
<thead>
<tr>
<th>Criterion</th>
<th>Problem</th>
<th>N</th>
<th>T (years)</th>
<th>BC-VND</th>
<th>GH-VND</th>
<th>CPLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P₁</td>
<td>26,021</td>
<td>13</td>
<td>0.74</td>
<td>3.73</td>
<td>-</td>
</tr>
<tr>
<td>%Min Gap</td>
<td>P₂</td>
<td>40,762</td>
<td>11</td>
<td>2.18</td>
<td>4.21</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>P₃</td>
<td>97,307</td>
<td>6</td>
<td>1.18</td>
<td>3.67</td>
<td>-</td>
</tr>
<tr>
<td>%Max Gap</td>
<td>P₁</td>
<td>26,021</td>
<td>13</td>
<td>0.74</td>
<td>5.12</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>P₂</td>
<td>40,762</td>
<td>11</td>
<td>2.18</td>
<td>4.98</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>P₃</td>
<td>97,307</td>
<td>6</td>
<td>1.20</td>
<td>3.97</td>
<td>-</td>
</tr>
<tr>
<td>%Ave Gap</td>
<td>P₁</td>
<td>26,021</td>
<td>13</td>
<td>0.74</td>
<td>4.42</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>P₂</td>
<td>40,762</td>
<td>11</td>
<td>2.18</td>
<td>4.64</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>P₃</td>
<td>97,307</td>
<td>6</td>
<td>1.19</td>
<td>3.79</td>
<td>-</td>
</tr>
<tr>
<td>Ave CPU (minutes)</td>
<td>P₁</td>
<td>26,021</td>
<td>13</td>
<td>21.62</td>
<td>4.83</td>
<td>9,261.95</td>
</tr>
<tr>
<td></td>
<td>P₂</td>
<td>40,762</td>
<td>11</td>
<td>215.17</td>
<td>15.99</td>
<td>23,353.00</td>
</tr>
<tr>
<td></td>
<td>P₃</td>
<td>97,307</td>
<td>6</td>
<td>278.34</td>
<td>240.75</td>
<td>12,707.03</td>
</tr>
</tbody>
</table>

Table 3: Evaluating the efficiency of the two variants of the proposed solution method
the 10 runs of each variant) the gap between the solution generated and the upper bound provided by CPLEX is smaller than 6%, and in 96.66% of cases, it is smaller than 5%.

- The computational time required by the variants is very reasonable. For instance, CPLEX may require almost 16 days to solve the linear relaxation of problem P₂, while very good feasible solutions for this problem are obtained in 16 minutes and 3.5 hours by GH-VND and BC-VND, respectively.

- When comparing the two variants, we observe that BC-VND provides the best gaps. GH-VND is, however, generally the best in terms of computational time.

Next, the ability of the VND procedure to improve the initial solutions provided by BC and GH is evaluated. Table 4 summarizes the value AveIterVND of the average number of VND major iterations performed (a major iteration is when the three neighborhoods have been explored), as well as the value %Impr. defined as follows:

\[
\%\text{Impr.} = \frac{Z_{\text{average}} - Z_{\text{init}}}{Z_{\text{init}}} \times 100
\]

where \(Z_{\text{average}}\) is as defined above (the average value of the solutions generated in the 10 runs) and \(Z_{\text{init}}\) is the average value of the initial solutions generated using either BC or GH. The last column AveCPU shows the average solution time in minutes (same values as in Table 3).

The results indicate that the BC procedure using the branch-and-cut algorithm is very efficient at generating very good solutions that VND fails to improve or improves very slightly. However, BC requires significantly longer computational times than the greedy heuristic, GH. Although the
initial solutions generated by \textbf{GH} are of low quality when compared with those generated by \textbf{BC}, \textbf{VND} improves them to the point that they are close to those generated by \textbf{BC}, and these improved solutions are obtained in significantly less computational time, indicating the benefits of \textbf{VND} when the initial solution is of low quality.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Ave $I_{\text{iter}_{\text{VND}}}$</th>
<th>%Imp.</th>
<th>Ave CPU (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\textbf{BC-VND}</td>
<td>\textbf{GH-VND}</td>
<td>\textbf{BC-VND}</td>
</tr>
<tr>
<td>$P_1$</td>
<td>1</td>
<td>15.4</td>
<td>0</td>
</tr>
<tr>
<td>$P_2$</td>
<td>1</td>
<td>14.1</td>
<td>0</td>
</tr>
<tr>
<td>$P_3$</td>
<td>4.4</td>
<td>10.0</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 4: Evaluating the ability of the VND procedure to improve the initial solutions

In summary, both variants outperform \textbf{CPLEX}. They offer an excellent compromise in terms of deviation from optimality and computational effort. On the one hand, with respect to solution quality, the results indicate that, for the tested problems, the variant \textbf{BC-VND} is slightly better. Solution times are in general significantly reduced when using \textbf{GH-VND}.

6. Conclusions

Production scheduling is a challenging and critical issue for mining companies exploiting open-pit mines. Determining the block mining sequence is a crucial step in maximizing the net present value of the mining operation. Decisions on block scheduling are subject to various types of constraints, typically slope constraints, bounds on mining, and bounds on processing. Furthermore, the open-pit mine production scheduling is even more difficult
because the number of blocks is large and because the metal content of the blocks is not known precisely at the time decisions are made. This yields a large-scale stochastic optimization problem.

We have addressed metal uncertainty using a two-stage stochastic programming approach and we have proposed a metaheuristic approach to solve the problem. An initial feasible solution is first generated using either an exact method or a greedy heuristic to solve sequentially a sub-problem for each period. A variable neighborhood descent procedure is then applied to improve the solution.

Upper bounds provided by CPLEX were used to evaluate the efficiency of the two variants of the proposed solution method, specified according to the process used for generating the initial solution. Tests were conducted on realistic actual large-scale instances, and their results indicate that the two variants generate solutions of very good quality, with an average deviation less than 3% from optimality, within a few minutes up to a few hours. When comparing the two variants, the results indicate that the variant based on an exact method slightly outperforms the one based on a greedy heuristic as far as the solution quality is concerned, but its average computational time is two times larger. The results also indicate that the VND procedure can substantially improve initial solutions of low quality.

Future research will be devoted to extending the proposed approach to account for additional operational constraints and to developing other efficient approaches for these more complex versions of the problem. One approach we are investigating consists of solving the linear relaxation of the problem using an efficient algorithm and then using an LP-rounding procedure to generate an integer solution.

Acknowledgments. The work in this paper was funded by Natural Science
and Engineering Research Council of Canada CRDPJ 411270-10, Discovery Grant ......, and the industry members of the COSMO Stochastic Mine Planning Laboratory: AngloGold Ashanti, Barrick, BHP Billiton, De Beers, Newmont, and Vale.

References


Ramazan, S., Dimitrakopoulos, R., 2012. Production scheduling with uncer-


