



HAL
open science

Adaptive Strategies for The Open-Pit Mine Optimal Scheduling Problem

Michel de Lara, Nelson Morales, Nathanaël Beeker

► **To cite this version:**

Michel de Lara, Nelson Morales, Nathanaël Beeker. Adaptive Strategies for The Open-Pit Mine Optimal Scheduling Problem. 2017. hal-01542705

HAL Id: hal-01542705

<https://hal.science/hal-01542705>

Preprint submitted on 20 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive Strategies for The Open-Pit Mine Optimal Scheduling Problem

Michel DE LARA*, Nelson MORALES†, Nathanaël BEEKER‡

June 20, 2017

Abstract

Within the mining discipline, mine planning is the component that studies how to transform the information about the ore resources into value for the owner. For open-pit mines, an optimal block scheduling maximizes the discounted value of the extracted blocks (period by period), called the *net present value* (NPV). However, to be feasible, a mine schedule must respect the slope constraints. The optimal *open-pit block scheduling problem* (OPBSP) consists, therefore, in finding such an optimal schedule. On the one hand, we introduce the dynamical optimization approach to mine scheduling in the deterministic case, and we propose a class of (suboptimal) adaptive strategies, the so-called *index strategies*. We show that they provide upper and lower bounds for the NPV, and we provide numerical results. On the other hand, we introduce a theoretical framework for OPBSP under uncertainty and learning.

Keywords: mine planning, open-pit block scheduling problem, optimization, index strategies, uncertainty, learning.

1 Introduction

Within the mining discipline, mine planning is the component that studies how to transform the information about the ore resources into value for the owner. Among the first decisions taken in the mine planning process is the choice of an exploitation method: it can be open-pit, that is achieved by digging from the surface, or it can be underground mining, that is done by constructing shafts and tunnels to access the mineralized zones. Other relevant products of the planning process are the *production plan*, that indicates how much will be produced at each time period, and the *mine scheduling*, that backs up the production plan by

*Université Paris–Est, CERMICS, 6–8 avenue Blaise Pascal, 77455 Marne la Vallée Cedex 2, France. Corresponding author: delara@cermics.enpc.fr, fax +33164153586

†DELPHOS Mine Planning Laboratory, Advanced Mining Technology Center, University of Chile

‡Université Paris–Est, CERMICS, 6–8 avenue Blaise Pascal, 77455 Marne la Vallée Cedex 2, France.

specifying what parts of the mine will be extracted in order to reach the production. A mine scheduling is constructed by means of a *block model*, which is a partition of the terrain into a 3-dimensional array of regular blocks. For each block, geostatisticians construct estimations on the different parameters like ore content, density, etc. The block model is considered an input to the mine planning process.

The operation of a mine is constrained by the overall capacity of transportation, which is translated into a number of tons per period (for example, a number of tons per day) and therefore in the number of blocks that can be extracted from the mine. Similarly, the overall tonnage of blocks for processing is also bounded by the plant processing capacity. Notice that, in the case of open-pit mines, not all blocks qualify for processing as an important part of the blocks may not contain enough material to have revenue but must be extracted in order to access attractive blocks.

Open-pit mines are also “special” in the sense that extraction must respect slope constraints: in order to reach blocks by digging from the surface, there is a minimum set of blocks that have to be extracted before. Indeed, the shape of the pit must be such that the stability of the walls and the accessibility are possible. This translates into a set of *precedence* constraints between the blocks. Other additional constraints to the operation of the mine may include *blending* constraints, which limit the average value of processed blocks for a certain attribute (like rock hardness or pollutant contents).

Considering all these elements, a *mine scheduling* can be seen as a (non injective) mapping from the set of blocks towards the time periods. Several blocks share the same extraction time. An optimal block scheduling maximizes the discounted value of the extracted blocks (period by period), called the *net present value* (NPV). However, to be feasible, a mine schedule must respect the capacity, blending and slope constraints. The optimal *open-pit block scheduling problem* (OPBSP) consists, therefore, in finding such an optimal block scheduling.

Related to block scheduling, and central in this article, is the notion of *block sequence*. A block sequence is a total order on the set of blocks, such that a larger rank means a later extraction (due to precedence constraints). Block sequences can be easily converted into block schedules by grouping blocks so that the overall capacities and blending constraints are satisfied (or, equivalently, replacing the slope constraints by the precedences given by the sequence).

The OPBSP is mostly formulated in a *deterministic* setting, where all values are supposed to be known to the planner *before* the planning phase: block model, prices and the operation of the mine (no failure). The traditional approach to optimal OPBSP uses Binary Integer Programming (see Appendix A). A very general formulation of OPBSP is due to Johnson (Johnson, 1968, 1969), which presented the problem of block scheduling under slope, capacity and blending constraints (the last ones given by ranges of the processed ore grade) within a multi-destination setting (that is, the optimization procedure yields as an output the process to apply to a given block). Unfortunately, the computational capabilities at the time made impossible to solve the formulation of Johnson for realistic case studies. Alternatively to the work of Johnson, Lerchs and Grossman (Lerchs and Grossman, 1965) proposed a very

simplified version of OPBSP in which block destinations are fixed in advance and the only constraint considered is the slope constraint, that is, the problem reduces to select a subset of blocks such that the contained value is maximized while the precedence constraint induced by the slope angles are held. This problem is known as the *ultimate pit* or *final pit* problem. Lerchs and Grossman also presented two key results: i) an efficient algorithm for solving the ultimate pit problem; ii) reducing the economic value of any given block makes the optimal solution of the ultimate pit problem to *shrink* (that is, if the values of the blocks decrease, the new solution is a subset of the original one). These two properties allow to produce nested pits and therefore, by trial and error, to introduce time and look for block sequences that satisfy other constraints like capacity. More detailed reviews can be found in (Newman, Rubio, Caro, Weintraub, and Eurek, 2010) (for a broad survey on operations research in mining) and (Chicoisne, Espinoza, Goycoolea, Moreno, and Rubio, 2011) (for the specific case of open-pit). Finally, an approach closer to the one that will be taken in this article is due to (Goodwin, Seron, Middleton, Zhang, Hennessy, Stone, and Menabde, 2006) which abstract the mine as a set of columns and embed the problem in the context of control theory.

Regarding mine planning under *uncertainty*, since the beginning of the nineties, an increasing number of open-pit mining strategies with uncertainty have been developed, following two articles by Ravenscroft (Ravenscroft, 1992), and Denby and Schofield (Denby and Schofield, 1995). The first one presents the *conditional simulation*, which is a technique used, for a mine with a known distribution, to generate sets of equally probable profiles called *scenarios*. We shall not dwell on the issue of the design of statistical models of ore distribution with uncertainty, using geostatistical tools such as *kriging* or others (Krige, 1984; Journel, 1983; Dowd, 1989; Lajaunie, 1990; Sichel, Dohm, and Kleingeld, 1995), and their simulation. The Denby and Schofield (Denby and Schofield, 1995) paper explains how to include uncertainty in a genetic algorithm, without precisely fixing the probabilistic frame. Since almost two decades, most of the stochastic models are based on the Ravenscroft approach, and present heuristics using a predefined set of scenarios. Dimitrakopoulos has been one of the driving force behind this trend, and has developed a large number of scenario-based strategies (Godoy and Dimitrakopoulos, 2004; Dimitrakopoulos and Ramazan, 2004; Dimitrakopoulos, Martinez, and Ramazan, 2007; Dimitrakopoulos and Ramazan, 2008). The solution is generally searched as a planning, that is an open-loop strategy: we have to plan and apply the entire scheduling without modifying it in the process of extraction, even if we get more information on the profile by discovering the exact value of the blocks. Golannejad, Osanloo, and Karimi (Golannejad, Osanloo, and Karimi, 2006) and Boland, Dumitrescu and Froyland (Boland, Dumitrescu, and Froyland, 2008) have also developed scenario-based strategies with a well defined mathematical and probabilistic framework: stochastic programming on a scenario tree. This allows solutions to be defined on a tree rather than only on a line (time), which clearly is an improvement. We are interested in how the mine scheduling optimization problem is formulated and possibly solved under uncertainty. We aim at designing solutions as *adaptive* strategies.

The paper is organized as follows, where our objectives are twofold. On the one hand,

we introduce in Section 2 the dynamical optimization approach to mine scheduling in the deterministic case. In Section 3, we propose a class of (suboptimal) adaptive strategies to attack the optimal OPBSP, the so-called *index strategies*. We show that they provide upper and lower bounds for the NPV. We provide numerical results in Section 4. On the other hand, we introduce in Section 5 a theoretical framework for OPBSP under uncertainty and learning.

2 The dynamical approach to open-pit block scheduling

As in (Goodwin, Seron, Middleton, Zhang, Hennessy, Stone, and Menabde, 2006), we define the mine state as a collection of pit depths at a certain number of surface locations and we represent the evolution of this state via a dynamic model that uses mining action as control input. In this setting, an admissible profile is one that respects local angular constraints at each point, and the open-pit mine optimal scheduling problem consists in finding a sequence of blocks and admissible profiles which maximizes the intertemporal discounted extraction profit.

2.1 A state control dynamical model

To simplify the description of the algorithms in this section, we will identify the blocks by vertical position $d \in \{1, \dots, D\}$ (d for depth) and by its horizontal position $c \in \mathbb{C}$ (c for column). In the sequel, it will also be convenient to see the mine as a collection of columns \mathbb{C} of cardinal C indexed by c , each column containing D blocks. We assume that blocks are extracted sequentially under the following hypothesis:

- it takes one time unit to extract one block (thus, the time unit is different from the one in Appendix A);
- only blocks at the surface may be extracted;
- a block cannot be extracted if the slope made with its neighbors is too high, due to geotechnical constraints on mine wall slopes;
- a retirement option is available where no block is extracted.

Denote discrete time by $t = t_0, \dots, T$, where the horizon T may be finite or infinite. At time t , the *state* of the mine is a *profile*

$$x(t) = (x_c(t))_{c \in \mathbb{C}} \in \mathbb{X} = \{1, \dots, D + 1\}^{\mathbb{C}} \quad (1)$$

where $x_c(t) \in \{1, \dots, D + 1\}$ is the vertical position of the top block with horizontal position $c \in \mathbb{C}$.

An admissible profile is one that respects local angular constraints at each point, due to physical requirements. A state $x = (x_c)_{c \in \mathbb{C}}$ is said to be *admissible* if the geotechnical slope constraints are respected in the sense that

$$\|x_{c'} - x_c\| \leq 1, \quad \forall c' \in \mathcal{M}(c), \quad c \in \mathbb{C}, \quad (2)$$

where $\mathcal{M}(c)$ is the set made of columns adjacent to column c . Denote by $\mathbb{A} \subset \mathbb{X}$, the set of admissible states satisfying the above slope constraints (2). Notice that $\|x_{c'} - x_c\| \leq 1$ may be replaced by $\|x_{c'} - x_c\| \leq k$ according to slope constraints, or even by non-isotropic local slope constraints. Implicitly, all cuboids have the same dimensions, but we could deal with less regular situations.

A decision is the selection of a column in \mathbb{C} , the top block of which will be extracted. A decision may also be the retirement option, that we shall identify with an additional fictitious column denoted ∞ . Thus, a decision c is an element of the set

$$\bar{\mathbb{C}} = \mathbb{C} \cup \{\infty\}. \quad (3)$$

The relation between columns sequencing and blocks scheduling is explicated in §4.2 in the Appendix.

At time t , if a column $c(t) \in \{1, \dots, C\}$ is chosen at the surface of the open-pit mine, the corresponding block is extracted and the profile $x(t) = (x_d(t))_{d \in \mathbb{C}}$ becomes

$$x_d(t+1) = \begin{cases} x_d(t) + 1 & \text{if } d = c(t) \\ x_d(t) & \text{else.} \end{cases}$$

In case of retirement option $c(t) = \infty$, then $x(t+1) = x(t)$ and the profile does not change. In other words, the dynamics is given by $x(t+1) = F(x(t), c(t))$ where

$$F_d(x, c) = \begin{cases} x_d + 1 & \text{if } d = c \in \mathbb{C} \\ x_d & \text{if } d \neq c \text{ or } d = \infty. \end{cases} \quad (4)$$

Indeed, the top block of column d is no longer at depth $x_d(t)$ but at $x_d(t) + 1$, while all other top blocks remain. Of course, not all decisions $c(t) = d$ are possible either because there are no blocks left in column d ($x_d = D + 1$) or because of slope constraints.

When in state $x \in \mathbb{A}$, the decision $c \in \mathbb{C}$ is *admissible* if the future profile $F(x, c) \in \mathbb{A}$, namely if it satisfies the geotechnical slope constraints. This may easily be transformed into a condition $c \in \mathbb{B}(x)$, where

$$\mathbb{B}(x) := \{c \in \mathbb{C} \mid F(x, c) \in \mathbb{A}\}. \quad (5)$$

2.2 Intertemporal profit maximization

The open-pit mine optimal scheduling problem consists of finding a sequence of admissible blocks which maximizes an intertemporal discounted extraction profit. It is assumed that the value of blocks differs in depth and column because richness of the mine is not uniform

among the zones as well as costs of extraction. The profit model states that each block has an economic value $V(d, c) \in \mathbb{R}$, supposed to be known (in the deterministic case). By convention $V(d, \infty) = 0$ when the retirement option is selected. Selecting a column $c(t) \in \mathbb{C}$ at the surface of the open-pit mine, and extracting the corresponding block at depth $x_{c(t)}(t)$ yields the value $V(x_{c(t)}(t), c(t))$. When $c(t) = \infty$, there is no corresponding block and the following notation $x_{c(t)}(t) = x_\infty(t)$ is meaningless, but this is without incidence since the value $V(x_\infty(t), \infty) = 0$.

With a discounting factor function $\rho(t)$ (for instance, $\rho(t) = \rho^t$, or $\rho(t) = \rho^{y(t)}$ for a yearly discount, where $y(t) = \lfloor \frac{t}{v} \rfloor$ is the “year” of time t and v is the number of blocks extracted per year), the value of a sequence (finite or infinite)

$$c(\cdot) := (c(t_0), \dots, c(T)) \quad (6)$$

is given by the criterion

$$J(c(\cdot)) := \sum_{t=t_0}^T \rho(t) V(x_{c(t)}(t), c(t)) . \quad (7)$$

Finding the value of the mine is solving the optimization problem

$$J^* := \max \left\{ \sum_{t=t_0}^T \rho(t) V(x_{c(t)}(t), c(t)), \quad (c(\cdot), x(\cdot)), c(t) \in \mathbb{B}(x(t)) \right\}, \quad (8)$$

where the maximum is over among all sequences $(c(\cdot), x(\cdot))$ which satisfy the slope constraints (5). Any such sequence $(c^*(\cdot), x^*(\cdot))$ such that $J(c^*(\cdot)) = J^*$ is an *optimal scheduling sequence*.

2.3 Dynamic programming equation and the curse of dimensionality

Theoretically, the open-pit mine optimal scheduling problem can be solved by dynamic programming (Bellman, 1957; Whittle, 1982; Bertsekas, 2000). It is well known that the dynamic programming approach suffers from the curse of dimensionality. Indeed, to give a flavor of the numerical complexity of the problem, the set \mathbb{A} of acceptable states has a cardinal of order $2^{10} \times 3^4 = 82\,944$ for a cubic $4 \times 4 \times 4$ mine, and of order $2^{16} \times 3^8 \times 4 \approx 1.72 \times 10^9$ for a cubic mine with 5 lateral blocks ($5 \times 5 \times 5$ cuboids).

Nevertheless, usual mines can reach more than 10^6 blocks, and the dynamic programming approach will not be usable in practice, without further state reduction.

3 Index strategies

The dynamic programming equation $\mathbb{V}(t, x) = \max_{c \in \mathbb{B}(x)} \left(\rho(t) V(c, x_c) + \mathbb{V}(t+1, F(x, c)) \right)$ naturally leads to solutions as *policies* or *strategies*, where an optimal decision c at time t depends not only on t , but also on the state $x(t)$ (De Lara and Doyen, 2008).

In this section, we shall present a class of strategies called *index strategies*. Among them, the so-called *Gittins index strategy* plays a special role, in that it easily provides an upper bound to the value of the mine.

3.1 Index based policy heuristics

We introduce a technique to obtain suboptimal results, based on so-called *index strategies*. The essence of this method is to model the problem by a set of *jobs*, each job being characterized by its state of progress, and combined with an index, whose value will indicate the priority of the job. At each time period, we choose the job of higher index to work at, which has the effect of modifying its state of progress, and update its index.

In the open-pit mine scheduling problem, the jobs in question will be the vertical columns located by their surface coordinates, and the state of progress will be the depths of the columns as defined previously. We define an index which, at each column, will map a value generally linked with the worth of the blocks around and below the top block of the column, and that includes or not the slopes constraints.

Various indices can be defined, each one giving a different strategy, and therefore different results and running times. Index algorithms with slope admissibility constraints work as follows. For each column c in the block model, and for each local state x_c (attached to the column), a certain $\text{idx}_c(x_c) \in \mathbb{R}$ is calculated. Then, for each column, we check whether or not its top block is extractable (in terms of the slope constraints). Among the columns whose top blocks are extractable, we pick the column with highest index and remove its top block, recalculating the index for that column. We iterate in this way until all blocks have been extracted, therefore generating a sequence of blocks.

The index of a column can be any function of the block model. We consider the following ones (see Figure 1 for a few examples). They correspond to existing heuristics that we interpret in terms of index.

- The greedy index idx^g , that is, the one that uses as index the economic value of the top-most block in the column (that has not been extracted yet).
- The Gittins index idx^G , that calculates the maximum discounted value of blocks in the column, relative to other columns. Block values are discounted block by block.
- The best-cone index idx^{C^*} . This index is similar to the previous one, but calculates all values for the different cones truncated at different depths, selecting the one with highest value.
- Toposort idx^τ . This is the index attached to the algorithm proposed by (Chicoisne, Espinoza, Goycoolea, Moreno, and Rubio, 2011). To calculate this index, we first solve the linear relaxation of the problem and then set the following value for each block

$$T_i = T + 1 - \sum_{t=1}^T t \Delta y_{it} + (T + 1) \left[1 - \sum_{t=1}^T y_{it} \right].$$

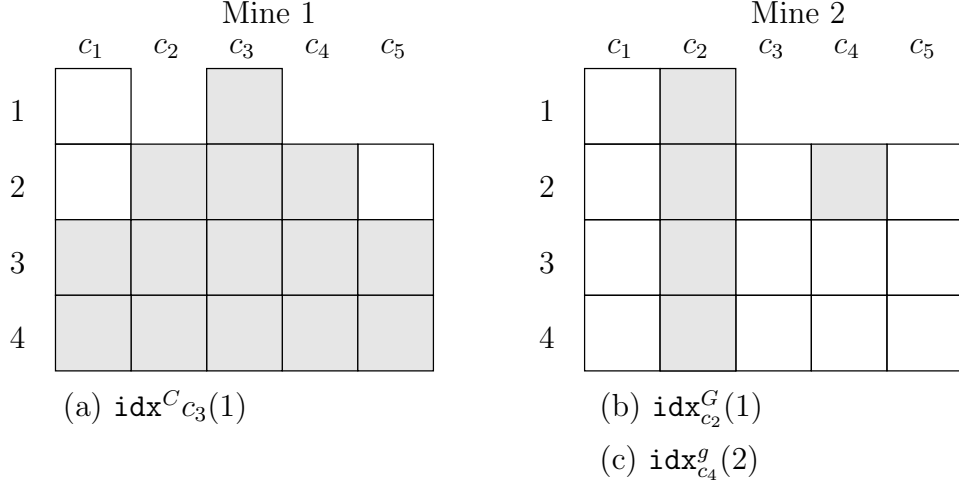


Figure 1: Example of index strategies in two small 2-D mines: (a) Cone, (b) Gittins and (c) Greedy.

Here, $\Delta \mathbf{y}_{it}$ is the binary variable associated to the decision of extracting a block i at time period t and $\mathbf{y}_{it} = \sum_{s \leq t} \Delta \mathbf{y}_{is}$ (see Appendix A for a detailed formulation). The index then corresponds to the value T_i of the top-most block in the column (that has not been extracted yet).

Famous techniques in mining can be interpreted as index strategies. For example, the Greedy index corresponds to a greedy strategy of always picking for extraction the block in the surface that: (a) is extractable (in terms of slope constraints) and (b) has the highest economic value. Furthermore, the Cone index described before is close to the *Gershon Algorithm* (Gershon, 1987) which also considers the successors' cone, but intersected with the ultimate pit.

3.2 An upper bound given by the Gittins index strategy

We shall now provide upper and lower bounds to the value (8) of the mine by means of index strategies.

To each profile $x = (x_c)_{c \in \mathbb{C}} \in \mathbb{X}$ and column $c \in \overline{\mathbb{C}}$, associate the local state $x_c \in \{1, \dots, D+1\}$, which is the vertical position of the top block with horizontal position c . For $\rho_{\#} \in]0, 1[$, define the Gittins index by

$$\text{idx}_c^G(x_c) := \sup_{\tau=t_0, \dots, +\infty} \frac{\sum_{s=t_0}^{\tau} \rho_{\#}^s \mathbf{V}(c, x_c + s)}{\sum_{s=t_0}^{\tau} \rho_{\#}^s}, \quad (9)$$

where $V(c, d) := 0$ when $d > D$ (this corresponds to fictitious blocks with zero values below the mine). With the notations of §2.1, the Gittins index strategy is defined by

$$c^G(t) \in \arg \max\{\text{idx}_c^G(x_c^G(t)), \quad c \in \overline{\mathbb{C}}\}, \quad (10a)$$

$$x^G(t+1) = F(x^G(t), c^G(t)). \quad (10b)$$

Proposition 1 *Suppose that $T = +\infty$, and that the discounting factor function $\rho(t)$ in (7) satisfies*

$$0 \leq \rho(t) \leq \rho_{\#}^t < 1. \quad (11)$$

The value (8) of the mine is bounded above as follows

$$J^* \leq \sum_{t=t_0}^{+\infty} \rho_{\#}^t V(x_{c^G(t)}(t), c^G(t)), \quad (12)$$

where the sequence $c^G(\cdot)$ is given by the Gittins index strategy (10) above. A lower bound is given by

$$J(c^i(\cdot)) \leq J^* \quad (13)$$

where the sequence $c^i(\cdot)$ is given by any index strategy respecting slope admissibility constraints

$$c^i(t) \in \arg \max\{\text{idx}_c^i(x_c^i(t)), \quad c \in \mathbb{B}(x^i)\}, \quad (14a)$$

$$x^i(t+1) = F(x^i(t), c^i(t)). \quad (14b)$$

Proof. Recall that J^* is the maximal value of (7) among all sequences $(c(\cdot), x(\cdot))$ which satisfy the slope constraints (5). Therefore, J^* is larger than any $J(c(\cdot))$, in particular for a sequence $c^i(\cdot)$ given by an index strategy respecting slope admissibility constraints. This is why (13) holds true.

On the other hand, by (11), we have that

$$J^* \leq \max\left\{\sum_{t=t_0}^T \rho_{\#}^t V(x_{c(t)}(t), c(t)), \quad (c(\cdot), x(\cdot)), \quad c(t) \in \mathbb{B}(x(t))\right\}.$$

Now, if we relax the slope admissibility constraints $c(t) \in \mathbb{B}(x(t))$, we deduce that

$$J^* \leq \max\left\{\sum_{t=t_0}^T \rho_{\#}^t V(x_{c(t)}(t), c(t)), \quad (c(\cdot), x(\cdot))\right\}.$$

Gittins theorem (Gittins, 1979) asserts that the optimum for the right hand side is achieved for the Gittins index strategy (10). Indeed, the problem is a deterministic multi-armed bandit, with independent arms since the slope admissibility constraints are relaxed, enabling thus to select any column. This is why (12) holds true. \square

Let NPV_{opt} be respectively the optimal NPV, NPV_{ind} the NPV given by any index strategy respecting the slopes constraints, and NPV_{ub} the NPV given by the Gittins index without slopes constraints, but with a discounting factor function $\rho(t)$ which satisfies (11). Then we have the following inequality:

$$\text{NPV}_{ind} \leq \text{NPV}_{opt} \leq \text{NPV}_{ub}. \quad (15)$$

4 Numerical examples

In this section, we present and discuss numerical results obtained using index heuristics over a set of synthetic data and the Marvin block model.

4.1 The Marvin dataset

The mine considered for this study is a well known mine named *Marvin*, which is available for use within the mine planning optimization Whittle from GenCom software. The overall number of blocks in Marvin is about 53,000. The block model contains the following data: block coordinates $(x, y$ and $z)$, copper and gold grades (cooper_i and gold_i respectively) and density. From these attributes we calculate: a block tonnage w_i (the product of the density by the volume of the block) and the copper content (the tonnage of the block by its copper grade). We aim to maximize overall copper production under a transportation capacity of 30,000 tons per day. Finally, we consider annual time-periods with a yearly discount rate equivalent to a 10% opportunity cost, hence a yearly discount factor $\rho = \frac{1}{1+0.1}$.

4.2 Using block sequences to obtain blocks schedulings

First, we present how to transform the output of an indexing strategy into a block scheduling and, therefore, a solution of OPBSP. We regard the output of an indexing algorithm as a *sequence* of blocks: a block sequence is a tuple of blocks $S = (i_1, i_2, \dots, i_N)$ that is compatible with the precedence constraints.

A sequence S can be converted into a solution of the open-pit block sequencing problem with capacity constraints, by creating nested pits that extract the blocks in the order given by the sequence. More precisely, let us say that $P \subset \mathcal{B}$ is capacity-feasible at time period t if for each resource r , we have that $\sum_{i \in P} a(i, r) \leq C_{r,t}^+$. We can then follow the next procedure to construct a block scheduling:

1. Set $k = 1, t = 1, P_0 = P_1 = \emptyset$.
2. While $t \leq T$:
 - (a) While $k < N$ and $(P_t \cup \{i_k\}) \setminus P_{t-1}$ is capacity-feasible at time period t : $P_t \leftarrow P_t \cup \{i_k\}, k \leftarrow k + 1$.
 - (b) $t \leftarrow t + 1$.

Notice, however, that there may exist some room for improvement on the obtained block scheduling, as it could happen that the blocks assigned to the very last time-period have a negative overall value. If this is the case, we reset these blocks as unextracted.

An alternative way to convert a block sequence into a block scheduling is the following. Given the sequence $S = (i_1, i_2, \dots, i_K)$, we set $\mathcal{B} = \{i_k : k = 1, 2, \dots, K\}$ and $\mathcal{A} = \{(i_k, i_{k+1}) : k = 1, 2, \dots, K-1\}$ and then directly solve the instance OPBSP($\mathcal{B}, \mathcal{A}, V, A, T, \rho, C^+, C^-$). This is equivalent to the procedure described above with the last “cleaning” phase.

4.3 Results and discussion

We now present the different results obtained for the heuristics and data sets, and we comment the findings of the numerical experiences.

Mine	-	Best Index	TopoSort	LP	Index UB
PM1	Time	0.43s	307s	307s	0.08s
	Value	358.42	432.14	439.30	521.53
PM2	Time	0.27s	375s	375s	0.07s
	Value	319.74	438.60	439.84	674.24
PM3	Time	0.27s	362s	362s	0.07s
	Value	139.50	149.06	198.84	318.72
Marvin	Time	1,036.00s	∞	∞	15.1s
	Value	392.9	-	-	488.5

Table 1: Numerical results by heuristic and instance. Values in million of copper tons. Running time in seconds. LP is linear programming. Index UB is index upper bound.

Numerical experiences were run with an Intel Pentium Dual Core, 2.8 Ghz processor running Linux 2.6.30-1. LP’s were solved using the GNU Linear Programming Toolkit (GLPK) using the primal simplex method.

Results in running time and economic value (NPV) are presented in Table 1. We observe that, while TopoSort obtains better results (closer to the LP upper bounds), this approach does not “scale” well, as it does not produce feasible solutions for the Marvin instances. Indeed, the main difficulty in this case is to solve the Linear Relaxation (LP), which did not end within reasonable time (12 hours). Conversely, the index strategies provide mixed results for the bounds, but the execution time is quite small, making them good candidate for fast schedulers and therefore useable with uncertainty scenarios, for example, on the grades.

We observe that there is a lot of room to improve the speed of the heuristics by optimizing the code or, for example, parallelizing some of the computations.

5 A mathematical framework for mine scheduling under uncertainty

We present here a general probabilistic framework for the OPBSP, that allows a dynamical use of information (learning), permitting to develop adaptive strategies, and which includes the planning solutions as a particular case. The approach is mostly mathematical and formal. However, in the last part, we suggest possible heuristics for future research.

5.1 Block attributes

Denote discrete time by $t = t_0, \dots, T$, where the horizon T is supposed to be finite for simplicity. Denote by \mathcal{B} the set of all blocks. Each block $b \in \mathcal{B}$, when extracted in period t , is characterized by a l -vector of *attributes* $\omega_b(t) \in \mathbb{W} = \mathbb{R}^l$. These attributes can for instance be the rock and ore volumes, price, cost, etc. In the deterministic model, these values will be simple real numbers perfectly known, but in our case it will be an uncertain vector.

This uncertain vector $\omega_b(t)$ will summarize various sources of uncertainty, and will be the basis of the construction of the worth $w_b(t)$ of block b at time t . It can for instance be of the following form, if the mine contains d different ores,

$$\omega_b(t) = (Price(t), Ore(b), Cost(b, t), \dots) \quad (16)$$

$$w_b(t) = Price(t) \cdot Ore(b) - Cost(b, t) , \quad (17)$$

where $Price(t) \in \mathbb{R}^d$ is an uncertain vector representing the selling prices per unit of the d different ores at time t , $Ore(b) \in \mathbb{R}^d$ is an uncertain vector representing the amount of each ore in the block b , and $Cost(b, t)$ is a uncertain variable representing the extraction cost of the block b at time t , each of them being coordinates of the attributes vector $\omega_b(t)$. This formulation presents the advantage to split the price distribution modeling and the distribution of the different ores in the mine; it is of course a simple instance that can be replaced by more sophisticated models including processing costs or other geotechnical data.

5.2 Scenarios

In the sequel, we will use the following notations

$$\omega(t) := (\omega_b(t))_{b \in \mathcal{B}}$$

for the collection of the attributes of the mine blocks at a time period t . while A sequence

$$\omega(\cdot) := (\omega(t_0), \dots, \omega(T))$$

is called a *scenario* and belongs to the product set

$$\Omega := \prod_{t=t_0}^T \prod_{b \in \mathcal{B}} \mathbb{W} = \mathbb{R}^{N \cdot (T-t_0+1) \cdot l} ,$$

which is the set of all possible scenarios. The situation where Ω is a singleton (a unique scenario) corresponds to the deterministic case.

5.3 A priori information data on the scenarios

Additional a priori information on the scenarios is generally given either by probabilistic or by set membership settings.

Stochastic assumptions

Notice that the vectors $\omega_b(t)$ are a priori not independent, neither with respect to b (spatially), nor with respect to t (temporally). Indeed, the price of raw materials is highly correlated in time, and a strong spatial correlation exists in the repartition of the ore. Many models of the orebody are based on the notion of variogram, which is a geostatistical tool giving an index of the spatial correlation of a certain type of ore. It gives a representation of the typology of the ore in a site, some metals as gold tending to aggregate into nuggets (with a strong short-distance correlation but a lower long-distance one), whereas other like copper will have a more long-distance dependence. It opens the way to orebody modeling such as kriging, a widespread interpolation method in geostatistics.

In the probabilistic formalism, the set Ω of all scenarios is equipped with the Borel σ -field $\mathcal{F} = \mathcal{B}_{\mathbb{R}^{N \cdot (T-t_0+1) \cdot l}}^o$. The $\omega_b(t)$ becomes random vectors, and the orebody is represented by a joint distribution law

$$\mathcal{L}(\omega_b(t), b \in \mathcal{B}, t \in [t_0, \dots, T]) , \quad (18)$$

which is a probability on (Ω, \mathcal{F}) . For instance, in the case of a unique type of ore, we can model $(Ore(b))_{b \in \mathcal{B}}$ by a Gaussian vector of size N , characterized by its mean vector $\mu = (\mathbb{E}[Ore(b)])_{b \in \mathcal{B}}$ and its covariance matrix $\Sigma = (Cov(Ore(b), Ore(b')))_{b, b' \in \mathcal{B}}$, with constant price $Price(t) = Price$ and cost $Cost(b, t) = Cost$. The set of the worths $w_b(t), b \in \mathcal{B}, t \in [t_0, \dots, T]$, is then a Gaussian vector of size $N \cdot (T - t_0 + 1)$ whose mean vector and covariance matrix can be calculated by means of μ and Σ .

Set membership

For a given block b and a given time period t , $\omega_b(t)$ can take its value in a certain set $\mathbb{S}(b, t) \subset \mathbb{R}^l$, which depends on the model. In the most general case, if we know nothing about the mine, $\mathbb{S}(b, t)$ will be \mathbb{R}^l , but it can for instance be reduced to intervals or even to a finite number of values, or to a singleton in a deterministic model.

5.4 Decisions and constraints

Each period of time (year, for instance), we can extract a certain number of blocks, and therefore we model our decision by a variable $u(t) \in \mathbb{U} = 2^{\mathcal{B}}$, corresponding to the blocks removed at time $t \in [t_0, \dots, T]$, which form a subset of \mathcal{B} . Here, $2^{\mathcal{B}}$ denotes the set of subsets of \mathcal{B} (the power set of \mathcal{B}). Since \mathbb{U} is a finite set, we equip it with the complete σ -field $\mathcal{U} = 2^{\mathbb{U}}$. We introduce the notations:

$$u^t := (u(t_0), \dots, u(t)) \quad \text{and} \quad u(\cdot) := (u(t_0), \dots, u(T)) .$$

The set $\mathbb{H} := \Omega \times \mathbb{U}^{T-t_0+1}$ is called the *history space*. Elements of the set $\mathbb{H}_t := \Omega \times \mathbb{U}^{t-t_0+1}$ represent history up to time t .

To capture slope and uncertain capacity constraints, we can restrict decisions as belonging to a subset $\mathbb{U}(t, \omega(\cdot), u^{t-1})$ of \mathbb{U} as follows:

$$u(t) \in \mathbb{U}(t, \omega(\cdot), u^{t-1}) . \quad (19)$$

5.5 On-line information

After having seen a priori information data on the scenarios, we now turn to on-line information available for the planner at time t . In essence, it is built upon the attributes $(\omega_b(t))_{b,t}$ we have discovered, and thus it a priori also depends on the past extractions u^{t-1} (i.e. the choices done on $[t_0, \dots, t-1]$). Mathematically, we shall represent information at time t as a σ -algebra \mathcal{I}_t on the history space $\Omega \times \mathbb{U}^{T-t_0+1}$.

- The blind information pattern is

$$\mathcal{I}_t = \{\Omega, \emptyset\} \otimes \{\mathbb{U}^{T-t_0+1}, \emptyset\},$$

where the decision-maker cannot distinguish elements in the history space (he cannot even recall his past decisions).

- The anticipative point of view corresponds to a stationary and constant

$$\mathcal{I}_t = \mathcal{F} \otimes \{\mathbb{U}, \emptyset\}. \quad (20)$$

The decision-maker knows the attributes of each block at each time, and knows them in advance: he is a visionary decision-maker. A visionary decision-maker having recall of his past decisions would be modeled as $\mathcal{I}_t = \mathcal{F} \otimes \bigotimes_{s=t_0}^{t-1} \mathcal{U}$.

- A *causal* information pattern is one in which the decision-maker cannot base his decision at time t upon his future decisions, and it is represented by the condition

$$\mathcal{I}_t \subset \mathcal{F} \otimes \bigotimes_{s=t_0}^{t-1} \mathcal{U}. \quad (21)$$

- In the *cumulative information pattern*, let us denote by

$$X(t, u^{t-1}) := \bigcup_{s=t_0}^t u(s) \subset \mathcal{B}$$

the set of the blocks which have been removed at time t following the sequence u^{t-1} of decisions in the periods $[t_0, \dots, t-1]$. If we assume that, each time we extract a block b at period t , we learn the exact value of the uncertainty $\omega_b(t)$, we define the information as

$$\mathcal{I}_t = \sigma\{(\omega_b(s), u^{s-1}), \quad b \in X(u^{s-1}, s), s \in [t_0, \dots, t-1]\}, \quad (22)$$

where we have abusively identified $(\omega_b(s), u^{s-1})$ with the coordinate random variable on the history space $\Omega \times \mathbb{U}^{T-t_0+1}$.

This formulation is adapted to a dynamical strategy, in which we learn step-by-step the information depending on our past choices.

5.6 Adaptive strategies

We now have the tools to define strategies adapted to on-line information. We assume that the information pattern is causal, that is, satisfies (21). A (causal) *strategy* is a sequence $\mathbf{P} = (\mathbf{P}_t)_{t=t_0, \dots, T}$ of *policies*

$$\mathbf{P}_t : \Omega \times \mathbb{U}^{t-t_0} \rightarrow \mathbb{U}$$

such that, for all $t = t_0, \dots, T$, \mathbf{P}_t is measurable with respect to \mathcal{I}_t .

Once a strategy \mathbf{P} and a scenario $\omega(\cdot)$ are given, decisions are inductively deduced by

$$u(t) = \mathbf{P}_t(\omega(\cdot), u^{t-1}). \quad (23)$$

Now, strategies will be our optimization variables.

If the family of sets $\mathbb{U}(t, \omega(\cdot), u^{t-1})$ in (19) is measurable with respect to \mathcal{I}_t , we may restrict ourselves to strategies in the admissible set \mathcal{P}^{ad} of the policies compatible with the constraints (capacity constraints, slopes constraints, etc.). For instance, a capacity constraints of k blocks per time unit will imply that, for $\mathbf{P} \in \mathcal{P}^{ad}$, the $u(t)$ generated by \mathbf{P} will not be more than k , or for a certain type of slopes constraints and precedence extraction relations, that the decisions $u(t)$ generated by \mathbf{P} will be compatible with the constraints induced by the blocks $X(t, u^{t-1})$ already removed.

A strategy $\mathbf{P} \in \mathcal{P}^{ad}$ is said to be an *open-loop* strategy if \mathbf{P}_t is a constant mapping for all t . In other words, an open-loop strategy plans the entire extraction sequence before starting it, and does not modify the sequence even if one gets information over time. In the more general case in which \mathbf{P} depends on the information, the strategy is said to be a *closed-loop* strategy. It corresponds to the adaptive case.

5.7 Decision criteria under uncertainty

For a given scenario $\omega(\cdot)$ and a given control sequence $u(\cdot)$, the sum of discounted profits (NPV) is given by

$$\mathbf{J}(\omega(\cdot), u(\cdot)) = \sum_{t=t_0}^T \rho(t) \sum_{b \in u(t)} w_b(t). \quad (24)$$

For a given scenario $\omega(\cdot)$ and a given strategy \mathbf{P} (adapted to the information pattern \mathcal{I}_t , $t = t_0, \dots, T$), let us put

$$\mathbf{J}^{\mathbf{P}}(\omega(\cdot)) := \mathbf{J}(\omega(\cdot), u(\cdot)) \quad \text{where } u(t) = \mathbf{P}_t(\omega(\cdot), u^{t-1}). \quad (25)$$

Now, contrarily to deterministic optimization, we do not know in advance the scenario $\omega(\cdot)$. How the decision-maker aggregates (25) with respect to the uncertainties, before optimizing, reflects his sensitivity to risk. The most common aggregates are the *robust* (or worst-case) and the *expected* criteria, but we also present other examples.

- **The expected criterion**

The expected optimization problem is

$$\max_{\mathbb{P} \in \mathcal{P}^{ad}} \mathbb{E}^{\mathbb{P}}[\mathbf{J}^{\mathbb{P}}(\omega(\cdot))] , \quad (26)$$

where $\mathbb{E}^{\mathbb{P}}$ denotes the mathematical expectation with respect to a probability $\mathbb{P} = \mathcal{L}(\omega_b(t), b \in \mathcal{B}, t \in [t_0, \dots, T])$ on the space Ω of scenarios. This formulation aims to maximize the mean NPV, that is the average value of all possibilities, weighted by their probability to happen. It is the best formulation you can choose in terms of average gain, but it does not penalize the possible realizations of the worst cases.

- **The robust criterion**

The robust optimization problem is

$$\max_{\mathbb{P} \in \mathcal{P}^{ad}} \min_{\omega(\cdot) \in \Omega} \mathbf{J}^{\mathbb{P}}(\omega(\cdot)) . \quad (27)$$

The strategy given by this formulation ensures to maximize the NPV if the worst case happens.

- **The multi-prior approach**

Suppose that the space Ω of scenarios is equipped with different probabilities \mathbb{P} in a set \mathfrak{P} , reflecting *ambiguity* with respect to the stochastic model. The multiprior approach is a combination of the robust and the expected criteria by taking the worst belief in term of expected NPV:

$$\max_{\mathbb{P} \in \mathcal{P}^{ad}} \min_{\mathbb{P} \in \mathfrak{P}} \mathbb{E}^{\mathbb{P}}[\mathbf{J}^{\mathbb{P}}(\omega(\cdot))] . \quad (28)$$

- **An expected criteria under probability constraint**

This last formulation is similar to the maximization of the expected NPV, but with an additional constraint to handle the risk. Given two parameters $\alpha \in \mathbb{R}$ and $p \in [0, 1]$, the expected optimization problem under probability constraint is

$$\max_{\mathbb{P} \in \mathcal{P}^{ad}} \mathbb{E}^{\mathbb{P}}[\mathbf{J}^{\mathbb{P}}(\omega(\cdot))] \quad (29)$$

under the restriction that

$$\mathbb{P}[\mathbf{J}^{\mathbb{P}}(\omega(\cdot)) \leq \alpha] \leq p . \quad (30)$$

The meaning of this formulation is to maximize the expected profit, with the condition that the chosen strategy will give, with high probability $1 - p$, at least a certain gain α .

Risk measures (Value-at-Risk, Conditional Value-at-Risk, etc.) could also be taken for aggregation (Föllmer and Schied, 2002).

5.8 From planning towards adaptive solutions

As we have seen, since the nineties, a certain number of “scenario-based strategies” have been proposed in the literature. The common denominator of these approaches is the use of conditional simulation (or any other simulation method), using the distribution law of the orebody, to generate a set of representative scenarios of the mine. Then, the solution is generally searched as a planning, that is an open-loop strategy.

A schematic way to represent the elaboration of a scenario-based strategy is the following

$$\mathcal{L} \xrightarrow{\text{sample}} (\omega_j(\cdot))_{j \in \mathcal{J}} \xrightarrow{\text{compute}} u(\cdot) , \quad (31)$$

that is, we sample the distribution law to obtain a set \mathcal{J} of scenarios. Then, with one or another method, we use these scenarios to elaborate an open-loop decision sequence $u(\cdot)$.

We suggest that this approach may be extended in the spirit of the *open-loop with feedback control* (OLFC) (Bertsekas, 2000). We do not detail the mathematics, but simply sketch the method. In the probabilistic setting, we assume that the arrival of an observation at time t allows us to update the conditional distribution \mathcal{L}^t on the space Ω of scenarios, knowing past observations. Then, the sketch is

$$\begin{aligned} & \mathcal{L}^0 \xrightarrow{\text{sample}} (\omega_j(\cdot))_{j \in \mathcal{J}^1} \xrightarrow{\text{compute}} (u^1(1), \dots, u^1(T)) \xrightarrow{\text{select}} u^1(1) \\ \hookrightarrow & \mathcal{L}^1 \xrightarrow{\text{sample}} (\omega_j(\cdot))_{j \in \mathcal{J}^2} \xrightarrow{\text{compute}} (u^1(1), u^2(2), \dots, u^2(T)) \xrightarrow{\text{select}} u^2(2) \\ & \dots \\ \hookrightarrow & \mathcal{L}^{T-1} \xrightarrow{\text{sample}} (\omega_j(\cdot))_{j \in \mathcal{J}^T} \xrightarrow{\text{compute}} (u^1(1), \dots, u^T(T)) \xrightarrow{\text{select}} u^T(T) \end{aligned}$$

returning a closed-loop strategy $u(\cdot)$.

To end this section, let us stress the fact that index methods are well adapted to the uncertain case, where the index may be a function of the conditional distribution \mathcal{L}^t .

6 Conclusions

We have presented the dynamic optimization approach to the open-pit block scheduling problem, a relevant problem in the mining industry that remains elusive to be solved due to its size. We have proposed heuristics based on so-called index strategies, together with upper and lower bounds for the NPV. Some of the results are promising, and index strategies are very fast and scale well for large instances of mines. This encourages their use when one generates a large number of scenarios, for which case a fast planning simulation and NPV calculation is crucial. In the future, we expect to do more experimentation on larger case studies and other (more realistic) data sets, and to compare the results with others found in the literature.

We have also introduced a general framework to deal with uncertainty and dynamical learning. We expect to implement this framework and to test it against real data.

Acknowledgments. The authors thank the STIC-AmSud OVIMINE project for the financial support. This paper was exposed at several OVIMINE meetings — 2011, March 17-18, Lima, Perú, 2011, September 2-19, Paris, France, 2011, October 6-8, Lima, Perú, 2012, January 9, Valparaiso, Chile, 2012, November 8, Santiago, Chile — and we thank the participants for their comments.

References

- Felipe Alvarez, Jorge Amaya, Andreas Griewank, and Nikolai Strogies. A continuous framework for open pit mine planning. *Mathematical Methods of Operations Research*, 73(1): 29–54, 2011.
- R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, N.J., 1957.
- D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts, second edition, 2000. Volumes 1 and 2.
- Daniel Bienstock and Mark Zuckerberg. Solving LP relaxations of large-scale precedence constrained problems. In *IPCO*, pages 1–14, 2010.
- N. Boland, I. Dumitrescu, and G. Froyland. A multistage stochastic programming approach to open pit mining production scheduling with uncertain geology. *Draft paper*, 2008.
- Renaud Chicoisne, Daniel Espinoza, Marcos Goycoolea, Eduardo Moreno, and Enrique Rubio. A new algorithm for the open-pit mine scheduling problem. October 2011. Draft article.
- M. De Lara and L. Doyen. *Sustainable Management of Natural Resources. Mathematical Models and Methods*. Springer-Verlag, Berlin, 2008.
- B. Denby and D. Schofield. Inclusion of risk assessment in open pit design and scheduling. *Transactions of the IMM, Section A: Mining Industry*, 1995.
- R. Dimitrakopoulos and S. Ramazan. Uncertainty-based production scheduling in open-pit mining. *Transactions of the SME*, 2004.
- R. Dimitrakopoulos and S. Ramazan. Stochastic integer programming for optimizing long-term production schedules of open pit mines: Methods, application and value of stochastic solutions. *Transactions of the IMM, Section A: Mining Industry*, 2008.
- R. Dimitrakopoulos, L. Martinez, and S. Ramazan. A maximum upside/minimum downside approach to the traditional optimization of open pit mine design. *Journal of Mining Science*, 2007.
- P.A. Dowd. Some observations on confidence intervals and kriging errors. *Geostatistics, Kluwer Academic Publishers*, 1989.

- H. Föllmer and A. Schied. *Stochastic Finance. An Introduction in Discrete Time*. Walter de Gruyter, Berlin, 2002.
- M. Gershon. Heuristic approaches for mine planning and production scheduling *Geotechnical and Geological Engineering*, 5: 1–13, 1987.
- J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B*, 41(2):148–177, 1979.
- M. Godoy and R. Dimitrakopoulos. Managing risk and waste mining in long-term production scheduling of open-pit mine. *Transactions of the SME*, 2004.
- J. Golamnejad, M. Osanloo, and B. Karimi. A chance-constrained programming approach for open pit long-term production scheduling in stochastic environments. *The Journal of the South African Institute of Mining and Metallurgy*, 2006.
- G.C. Goodwin, M.M. Seron, R.H. Middleton, M. Zhang, B.F. Hennessy, P.M. Stone, and M. Menabde. Receding horizon control applied to optimal mine planning. *Automatica*, 42(8):1337–1342, 2006.
- Dorit Hochbaum and A. Chen. Performance analysis and best implementation of old and new algorithms for the open-pit mining problem. *Operations Reserach*, 48:894–914, 2000.
- T. B. Johnson. *Optimum open-pit mine production scheduling*. PhD thesis, 1968.
- T. B. Johnson. *Optimum open-pit mine production scheduling*. 1969.
- A.G. Journel. Non-parametric estimation of spatial distributions. *Journal of the International Association of Mathematical Geology*, 1983.
- D.G. Krige. Geostatistics and the definition of uncertainty. *Transactions of the IMM, Section A: Mining Industry*, 1984.
- C. Lajaunie. Comparing some approximate methods for building local confidence intervals for predicting regionalised variables. *Mathematical Geology*, 1990.
- H. Lerchs and H. C. Grossman. Optimal design of open-pit mines. *Transactions C.I.M.*, 58: 47–54, 1965.
- Georges Matheron. Paramétrage de contours optimaux. *Note Geostatistics*, 128, 1975.
- Alexandra M. Newman, Enrique Rubio, Rodrigo Caro, Andres Weintraub, and Kelly Eurek. A review of operations research in mine planning. *Interfaces*, 40(3):222–245, 2010.
- P.J. Ravenscroft. Risk analysis for mine scheduling by stochastic simulation. *Transactions of the IMM, Section A: Mining Industry*, 1992.

H.S. Sichel, C.E. Dohm, and W.J. Kleingeld. New generalized model of observed ore value distributions. *Transactions of the IMM, Section A: Mining Industry*, 1995.

P. Whittle. *Optimization over Time: Dynamic Programming and Stochastic Control*, volume 1. John Wiley & Sons, New York, 1982.

A Integer linear programming formulation of the open-pit block scheduling problem

In this section we introduce the relevant notation and formulation for the deterministic case of the open-pit block scheduling problem using binary linear programming.

A.1 Modeling and notation

We consider \mathcal{B} the set of all blocks and $N = |\mathcal{B}|$. We denote the elements of \mathcal{B} (the blocks) with indices i, j , unless otherwise stated. Similarly, we consider $T \in \mathbb{N}$ time-periods and denote individual time-periods with $s, t = 1, 2, \dots, T$. T is called the *time horizon*. We also use the notation $\mathbf{T} = \{1, 2, \dots, T\}$ for the set of time-periods.

Slope constraints are modeled as precedence constraints and encoded as a set of *arcs* $\mathcal{A} \subset \mathcal{B} \times \mathcal{B}$, so $(i, j) \in \mathcal{A}$ means that Block j has to be extracted before Block i . We say, in this case, that Block j is a predecessor of Block i , which in turn is a successor of j . Notice that arc (i, j) goes from the successor to the predecessor.

In this work we address a simplified version of the problem in which the decision of the destination of the block is done beforehand. This allows us to

1. consider that the net profit (which can be negative) of processing Block i is already known and noted as $v_i \in \mathbb{R}$, and
2. define a set of resources \mathcal{R} , and for Block $i \in \mathcal{B}$ and Resource $r \in \mathcal{R}$ the quantity $a(i, r)$ of resource r that is used when i is processed.

For each time period t , upper and lower bounds on the consumption of resource r are given by the quantities $C_{rt}^- \in \{-\infty\} \cup \mathbb{R}$ and $C_{rt}^+ \in \{+\infty\} \cup \mathbb{R}$, respectively.

We also assume that the block is processed in the same time period in which it is extracted from the mine (that is, we do not allow to stock material for future processing). We also assume, as is usual in these models, that all block extraction, handling and processing is done within a time-period length.

While the modeling can be easily extended to the general case, the heuristics presented in this article do not always work to the case in which blending constraints apply, therefore, we assume there are not such constraints.

Finally, Table 2 summarizes the notation introduced in this Appendix.

Symbol	Meaning
\mathcal{B}	The set of blocks
i, j	Blocks (elements of \mathcal{B})
s, t	Time-periods
T	Time horizon (number of periods)
\mathbf{T}	Set of time-periods
\mathcal{A}	Set of precedence arcs
\mathcal{R}	Set of resources
v_i	Economic value (net profit) of Block i
$a(i, r)$	Consumption of Resource r by Block i
C_{rt}^-, C_{rt}^+	Lower and upper bounds on resource r

Table 2: Main notations in the Appendix

A *block scheduling* is a function $\tau : \mathcal{B} \rightarrow \{1, 2, \dots, T, \infty\}$ where $\tau(i)$ is the time-period in which block i is extracted, hence, a block scheduling must satisfy the precedence constraints, that is if $(i, j) \in \mathcal{A}$ then $\tau(i) \geq \tau(j)$.

If τ is a block scheduling then the preimage sets $P_1 = \tau^{-1}(1)$ and $P_t = P_{t-1} \cup \tau^{-1}(t)$ for $t > 1$ are called *pits*. We observe that $P_t \subset P_{t+1}$ hence we say that the pits are *nested*.

A *block sequence* is a tuple $s = (s_1, s_2, \dots, s_K) \in \mathcal{B}^K$ such that $k \neq \ell \Rightarrow s_k \neq s_\ell$ (all blocks in the tuple are different) and that is compatible with the precedence constraints, that is if $(s_k, s_\ell) = (i, j) \in \mathcal{A}$ then $\ell > k$ (predecessors appear before in the sequence).

A.2 The binary programming formulation

The open-pit block scheduling problem is defined on the following variables. For each $i \in \mathcal{B}, t = 1, 2, \dots, T$:

$$y_{it} = \begin{cases} 1 & \text{block } i \text{ is extracted by time-period } t, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that the interpretation of variable y_{it} is *by* time-period, that is $y_{it} = 1$ if and only if block i has been extracted (and processed) at some period s with $1 \leq s \leq t$. For this reason, it is also useful to introduce the following auxiliary variables for any $i \in \mathcal{B}$: $\Delta y_{i1} = y_{i1}$, and $\Delta y_{it} = y_{it} - y_{i,t-1}$ for $t = 2, 3, \dots, T$. We have that $y_{it} = \sum_{s=1}^t \Delta y_{is}$ and $\Delta y_{it} = 1$ if and only if block i is extracted exactly at time period t .

The optimization program is the following:

$$(\text{OPBSP}) \max \sum_{t=1}^T \rho^t \sum_{i=1}^N v_i \Delta y_{it} \quad (33)$$

$$y_{it} \leq y_{jt} \quad (\forall (i, j) \in \mathcal{A})(\forall t \in \mathbf{T}) \quad (34)$$

$$y_{i,t-1} \leq y_{it} \quad (\forall i \in \mathcal{B})(\forall t = 2, \dots, T) \quad (35)$$

$$\sum_i a(i, r) \Delta y_{it} \leq C_{rt}^+ \quad (\forall r \in \mathcal{R})(\forall t \in \mathbf{T}) \quad (36)$$

$$\sum_i a(i, r) \Delta y_{it} \geq C_{rt}^- \quad (\forall r \in \mathcal{R})(\forall t \in \mathbf{T}) \quad (37)$$

$$y_{it} \in \{0, 1\} \quad (\forall i \in \mathcal{B})(\forall t \in \mathbf{T}) .$$

Equation (33) presents the goal function, which is the discounted value of extracted blocks over the time horizon T . Equation (34) corresponds to the precedence constraints given by the slope angle. Equation (35) states that blocks can be extracted only once. Finally, Equations (36) and (37) fix the resource consumption limits.

For a block model \mathcal{B} , precedence arcs \mathcal{A} , block values $V = (v_i)_{i \in \mathcal{B}}$ and attribute matrix $A = (a(i, r))_{i \in \mathcal{B}, r \in \mathcal{R}}$ we will use the notation $\text{OPBSP}(\mathcal{B}, \mathcal{A}, v, A, T, \rho, C^+, C^-)$ to denote an instance of the open-pit block scheduling problem for a certain time horizon T , discount rate ρ , and resource limit matrices $C^- = (C_{r,t}^-)_{r,t}$ and $C^+ = (C_{r,t}^+)_{r,t}$. We will omit some of the parameters if the context allows it.