Les Cahiers du GERAD ISSN: 0711–2440

Long-term mine production scheduling with multiple processing destinations based on multi-neighbourhood Tabu search

R. Senécal
R. Dimitrakopoulos

G–2015–87

September 2015

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Long-term mine production scheduling with multiple processing destinations based on multi-neighbourhood Tabu search

Renaud Senécal
Roussos Dimitrakopoulos

GERAD & COSMO – Stochastic Mine Planning Laboratory, Department of Mining and Materials Engineering, McGill University, Montreal (Quebec) Canada, H3A 2A7

renaud.senecal2@mail.mcgill.ca
roussos.dimitrakopoulos@mcgill.ca

September 2015

Les Cahiers du GERAD
G–2015–87

Abstract: This paper presents a new mathematical formulation to address mine production scheduling with multiple processing streams, under mineral supply uncertainty, and where the destination is formulated as a variable for each block. The proposed mathematical model maximizes discounted cash flows and penalizes deviations from production targets. A parallel multi-neighbourhood Tabu Search metaheuristic is developed to optimize the proposed model. An application at a gold deposit shows the practical aspects and computational advantages as well as the ability of providing a schedule that meets production targets and provides a stable destination feed in term of tonnage.

1 Introduction

The purpose of the open pit mine production scheduling problem (OMPSP) is to generate a feasible extraction sequence that maximizes the net present value (NPV), while meeting processing requirements, such as processing and blending constraints. Over the past several decades, research has focused on the mathematical formulation and exact solution of the OPMPSP (Gershon 1983; Dagdelen and Johnson 1986; Tolwinski and Underwood 1996; Tachefine and Soumis 1997; Bley et al. 2010). Ramazan (2001) and Cacetta and Hill (2003) consider blending constraints in their formulation, which makes the formulation hard to solve. Bley et al. (2012) add a stockpile to the OPMPSP (OPMPSP+S), leading to a new quadratic non-convex formulation. However, the aforementioned models do not account for geological uncertainty, which is one of the key reasons why the schedules generated fail to meet production targets (Baker and Giacomo 1996; Vallee 2000). Over the past several decades, geostatistical simulation methods (Journel 1974; David 1988; Goovaerts 1997) have been developed to provide equally probable representations of the mineral deposit. Using an existing schedule, it is possible to assess the risk in terms of metal content and ore tonnages for the design using these simulations (Ravenscroft1992; Dowd 1997; Dimitrakopoulos et al., 2002).

Stochastic integer programming (SIP) (Birge and Louveaux 2011), along with different orebody simulations, are used to model the OPMPSP under geological uncertainty. Ramazan and Dimitrakopoulos (2007, 2013) present a two-stage SIP for OPMPSP where the first stage variables correspond to the year of extraction and the second stage variables are the deviations from production targets. The SIP formulation proposes to maximize the NPV of the mining project, while minimizing deviations from production targets, and uses a geological risk discount factor (GRD) (Dimitrakopoulos and Ramazan 2007) to penalize more deviations in early periods to delay risk. Ramazan and Dimitrakopoulos (2013) propose a formulation of the OPMPSP+S where the average grade of the stockpile is fixed. However, the assumption of fixed grade can result in misleading results, since blocks with a lower grade than the fixed grade will be more profitable, in terms of NPV, to be sent and reclaimed from the stockpile and be processed than by processing the block directly. The scheduling problems are solved using a heuristic that divides the problem into smaller sub-problems. The first sub-problem consists of solving the complete SIP for only a small given number of periods while considering only blocks that can be extracted during these periods. The second step consists of solving the later periods until the life of mine and includes in the optimization the blocks that have been extracted during the last period of the previous step. The solution approach takes about forty hours in total for a sub-optimal solution and uses only 22,000 blocks, which makes the method inapplicable for a large scale deposit.

Lamghari and Dimitrakopoulos (2013) present an approach to solve the OPMPSP efficiently using a Tabu Search algorithm (TS) (Glover and Laguna 1997) where the neighborhood to be searched at each iteration consists of all feasible schedules that differ by only a block schedule in a different period. Lamghari et al. (2014) present a variable neighbourhood search metaheuristic (VNS) with three different types of searches. The first one consists of exchanging the year of extraction of two blocks, such that the new solution does not violate slope constraints after the exchange. The second and third searches consist of advancing and delaying the year of extraction of blocks with respect to the slope constraints. The application of those metaheuristic approaches for the open pit mining problem shows the ability to generate a close to optimal solution in a practical amount of time. Thus, in this paper a metaheuristic approach is developed to account for a more complex case, where a stockpile is considered, as well as multiple processing streams.

As an extension of the TS procedure, the multi-neighbourhood Tabu Search algorithm (MNTS), has been shown to be efficient in other fields of combinatorial optimization. Examples include Jin et al. (2010), where the authors define four neighbourhood structures for the vehicle routing problem (VRP). A Granular Tabu Search (Toth and Vigo 2003) is used to restrict some unpromising moves to be computed. The algorithm simultaneously launches multiple Tabu Search Threads (TST), where each thread runs independently for a specified amount of time with a specific neighbourhood, and a synchronization step is added to share the best solution found so far. Wu et al. (2012) define three different neighbourhoods for the maximum weight clique problem (MWCP) and the union of those neighbourhoods is searched at each iteration. The best solution of the three neighbourhoods is selected at each iteration. The results show that the algorithms can provide very good quality solution and, in some instances, outperform existing algorithms.

In this paper, a two-stage stochastic formulation using stockpiles, and considering multiple processing destinations is presented. The block destination is added into the formulation as first stage variables, leading to a scenario-independent block destination. To solve this formulation, a parallel multi-neighbourhood Tabu Search approach is implemented (P-MNTS), where the neighborhood search is parallelized to gain more efficiency. The algorithm presented in this paper follows the general idea presented by Wu et al. (2012), but also uses a Granular Tabu Search procedure to restrict the search to only promising regions The remainder of the paper is organized as follows: first, a mathematical description of the OPMPSP+S with multiple processing destinations is given. In Section 3, the description of the method to solve the model is given. The parallelization procedure is then given in Section 4. Numerical results for the different application of the method are given in Section 5. Finally, conclusions and improvements are discussed in Section 6.

2 SIP formulation of stochastic mine scheduling

2.1 Notation

The following notation is used

- N is the number of blocks considered in the mine and i represents a block index, $i \in \{1...N\}$.
- P_i is the set of blocks that must be extracted in order to extract block i,
- S is the number of scenarios used for grade uncertainty and s is a scenario index, $s \in \{1...S\}$.
- w_{is} is the weight of block i under scenario s.
- T is the life time of the mine and t is a period index, $t \in \{1...T\}$.
- P is the number of destinations in the mining complex and p is the index of a destination.
- W^t is the maximum total weight (mining capacity) that can be extracted at period t.
- g_{is} is the grade of block i in scenario s.
- d is the financial discount rate.
- r is the geological risk discount rate.
- C^p_{su} is the undiscounted cost per unit of surplus material going to destination p and $C^{pt}_{su} = \frac{C^p_{su}}{(1+r)^t}$ is the discounted cost per unit of surplus material going to destination p.
- C^p_{sh} is the undiscounted cost per unit of shortage material going to destination p and $C^{pt}_{sh} = \frac{C^p_{sh}}{(1+r)^t}$ is the discounted cost per unit of shortage material going to destination p.
- δ_p is the undiscounted cost per unit for sending material to the stockpile associated with destination p and $\delta^t_p = \frac{\delta_p}{(1+r)^t}$ is the cost per unit for sending material to the mill stockpile at period t.
- η_p is the undiscounted unit cost per unit for taking material from the stockpile associate with destination p and $\eta^t_p = \frac{\eta_p}{(1+r)^t}$ is the unit cost per unit for taking material from the stockpile associated with destination p at period t.
- α^{pt}_{is} is a discounted profit generated if block i is mined during period t in scenario s.
- The expected profit made if block i is extracted in period t and sent to destination p is given by

$$E\{NPV\}^{pt}_i = \sum_{s=1}^{S} \frac{\alpha^{pt}_{is}}{S}.$$

- SG^p_s is the fixed grade of the material in the stockpile of destination p under scenario$s$. The fixed grade is done using the formula

$$SG^p_s = \frac{\sum\limits_{i \in B_s} g_{is} \cdot w_i}{\sum\limits_{i \in B_s} w_i},$$

  where B^p_s is the set of blocks for which destination p is the most profitable destination under scenario s. SG^p_s is, therefore, the average over all blocks that are more profitable to be processed in destination p under scenario s.

- SV_s^{pt} is the unit discounted profit for material in the stockpile associate with destination p at period t under scenario s.

$$SV_s^{pt} = \frac{SG_s^p \cdot Recovery_p \cdot (MetalPrice - SellingCost) - ProcessingCost_p}{(1+d)^t},$$

where $Recovery_p$, $ProcessingCost_p$ are the recovery and processing cost (\$/unit) of material processed in destination p and $MetalPrice - SellingCost$ are in (\$/$unit_{metal}$).
- Q_s^p is the metal recovered in the stockpile of destination p under scenario s for a unit of ore.
- O^{pt} is the ore target associated with destination p in period t.
- Q^{pt} is the metal quantity target associated with destination p in period t.
- S^{pt} is the maximum capacity that can be in the stockpile in destination p in period t.

The following variables are used to formulate the problem

- $x_i^t = \begin{cases} 1 & \text{if block } i \text{ is mined during period } t \\ 0 & \text{otherwise.} \end{cases}$
- $y_i^{pt} = \begin{cases} 1 & \text{if block } i \text{ is sent to process } p \text{ during period } t \\ 0 & \text{otherwise.} \end{cases}$
- d_{os-}^{pt},d_{os+}^{pt} represent the shortage/surplus of a unit of material for destination p in period t under scenario s.
- d_{qs-}^{pt},d_{qs+}^{pt} represent the shortage/surplus of a unit metal recovered for destination p in period t under scenario s.
- k_{s-}^{pt},k_{s+}^{pt} represent the amount of material taken from or sent to destination p's stockpile at period t under scenario s.
- u_s^t represents the amount of material in the destination p's stockpile at the end of period t under scenario s.

2.2 Model

$$\max \sum_{t=1}^{T} \left\{ \underbrace{\sum_{i=1}^{N}\sum_{p=1}^{P} E\{NPV\}_i^{pt}\, y_i^{pt}}_{Part\,1} + \underbrace{\sum_{s=1}^{S}\sum_{p=1}^{P} \frac{SV_s^{pt}}{S} k_{s-}^{pt}}_{Part\,2} - \underbrace{\sum_{p=1}^{P}\sum_{s=1}^{S} \frac{\left((SV_s^{pt} + \delta^{pt})\, k_{s+}^{pt} + \eta^t k_{s-}^{pt}\right)}{S}}_{Part\,3} - \underbrace{\sum_{p=1}^{P}\sum_{s=1}^{S} d_{os-}^{pt} + d_{os+}^{pt} + d_{qs-}^{pt} + d_{qs+}^{pt}}_{Part\,4} \right\} \quad (1)$$

The objective Function (1) consists of four parts. The first part represents the expected economic value obtained by processing the extracted blocks at the chosen destination. In this specific case, since a fixed grade is used, the value of all blocks extracted is taken into account, but the economic value associated with the amount of tonnage sent to the stockpile is removed in Part (3). Part (2) computes the value obtained by processing resources from the stockpile using the fixed stockpile grade. Part (3) discounts the cost incurred by sending material to the stockpile and a rehandling cost is added for both taking and sending material to the stockpile. Thus, sending and removing from the stockpile during the same period will incur a loss that will not happen in an optimal solution. Part (4) is the GRD added to control the deviations from ore and quantity of metal targets for each destination, period and each scenario (Ramazan and Dimitrakopoulos 2013).

The following constraints are considered for the present model

$$\sum_{p=1}^{P} y_i^{pt} = x_i^t \qquad \forall i, t \tag{2}$$

$$\sum_{t=1}^{T} x_i^t \leq 1 \qquad \forall i \tag{3}$$

$$x_i^t - \sum_{\tau=1}^{t} x_j^{\tau} \leq 0 \qquad \forall i, j \in P_i, t \tag{4}$$

$$\sum_{i=1}^{N} w_{is} x_i^t \leq W^t \qquad \forall t \tag{5}$$

$$\sum_{i=1}^{n} w_{is} y_i^{pt} + d_{os-}^{pt} - d_{os+}^{pt} + k_{s-}^{pt} - k_{s+}^{pt} = O^{pt} \tag{6}$$

$$\sum_{i=1}^{n} q_{is} y_i^{pt} + d_{qs-}^{pt} - d_{qs+}^{pt} + Q_s^p k_{s-}^{pt} - Q_s^p k_{s+}^{pt} = Q^{pt} \tag{7}$$

$$u_s^{pt} = u_s^{p(t-1)} - k_{s-}^{pt} + k_{s+}^{pt} u_s^{pt} \leq S^{pt} \qquad \forall t, s, p \tag{8}$$

$$x_i^t \in \{0, 1\} \qquad \forall i, t \tag{9}$$

$$y_i^{pt} \in \{0, 1\} \qquad \forall i, t, p \tag{10}$$

$$k_{s+}^{pt}, k_{s-}^{pt}, u_s^t \geq 0 \qquad \forall s, t, p \tag{11}$$

$$d_{os+}^{pt}, d_{os-}^{pt}, d_{qs-}^{pt}, d_{qs+}^{pt} \geq 0 \qquad \forall t, p, s \tag{12}$$

Constraint (2) is introduced to link the extraction variable to the destination variable and represents the fact that a block needs to be extracted if it is sent to a destination and that if it is extracted then it must be sent to one of the destinations. Constraints (3) and (4) ensure a block is not extracted more than once, and that all blocks covering it are also extracted by the same period. Constraint (5) ensures that during a period, the extraction capacity of the mining equipment is not exceeded. Constraints (6) and (7) define the capacity of the different processes in terms of ore and metal quantities. The ore and metal quantities are formulated as targets to be met and the deviations from those targets are penalized. These targets allow a control over the ratio $\frac{Q^{pt}}{O^{pt}}$, which is the average grade sent to the process p at period t. Constraint (8) is associated with the stockpiles and balance the material stockpiled at the beginning and the end of each period for each process. Constraints (9), (10), (11) and (12) define the variables as binary or continuous.

Figure 1 shows an example of a deposit that can be modeled using this formulation.

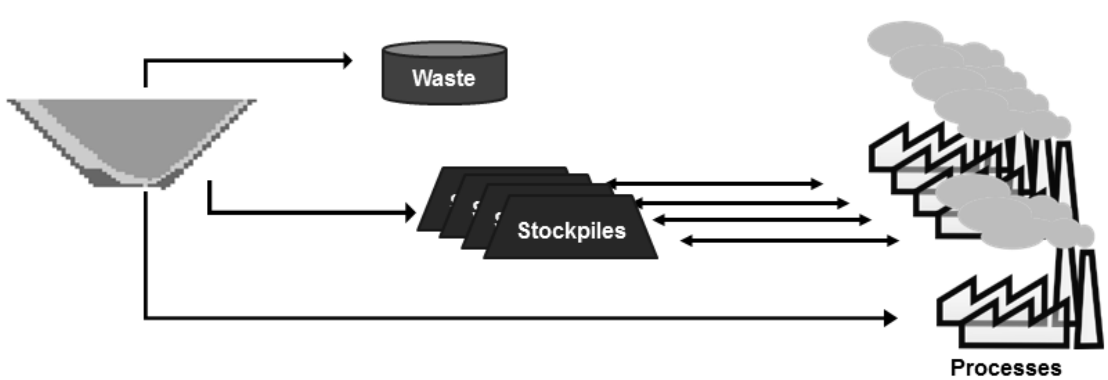

Figure 1: Example of a generic deposit.

Note that the constraints in the model are generic. Thus, to simulate processes without income and stockpile, such as a waste dump, the stockpiles variables and the coefficient for deviations must be set to 0 for the specific destination p.

3 Multi-neighbourhoods Tabu Search method

The proposed SIP model is solved using a MNTS approach, where basic move operators over the period and destination variables are used to generate neighbour solutions. Those neighbours are grouped into neighbourhoods that are searched simultaneously by the Tabu Search procedure. The Tabu Search procedure uses a short term data structure referred to as a Tabu List (TL) to track recently applied moves, and allow solution with lower value to be visited when it reaches a local optima. This method works as follows:

1. Define an initial feasible solution.
2. Search the multiple neighbourhoods and apply the best admissible move over the initial solution. Loop until a stopping criterion is met.
3. Apply a diversification strategy to generate a new starting solution, once a particular criterion is met.

In order to restrict the search to only promising solutions and to avoid the search of the whole neighbourhoods, a mechanism to remove solutions from the search space is used. The mechanism restricts solutions that are far from meeting processing capacity constraints (6) to be considered during the search process. To search the solution space extensively, a mechanism that considers infeasible solutions in terms of mining capacity constraints (5) in the neighbourhoods is also implemented.

3.1 Solution definition

A 'solution' given by the algorithm corresponds to (i) the period of extraction and (ii) the destination for each block. In the remainder of this paper, a solution x is an array containing the two attributes for all blocks; $x_i.destination$ is the current destination for block i and $x_i.period$ corresponds to its period of extraction. In the following, blocks that are not extracted will be denoted by $x_i.period = T + 1$.

3.2 Basic move operators and neighbourhoods

In this MNTS procedure, two different types of move operators have been implemented to generate the new neighbours to be searched. Based on those move operators four different neighbourhoods have been defined in total. The first basic move operator, $\oplus_{pr}$, corresponds to a change in a block's extraction period. Applying this move operator to the current solution, x, for a specific block and period, will result in a new solution denoted by $x \oplus_{pr} (i, t)$ and where $\oplus_{pr} (i, t)$ is called a move. A Tabu List (TL) is associated with this move operator (TL-P) that keeps track of the recently applied moves of the first type during the progression of the algorithm. The second move operator, $\oplus_d$, corresponds to a change of destination for a block and applied to a solution, x, will lead to the solution denoted by $x \oplus_d (i, p)$. A Tabu List is also associated with this move operator (TL-D).

To be admissible, a move associated with solution x must respect the slope constraints (4), it must not be in any TL (TL-D or TL-P), or if it is, it must lead to a global maxima. The first neighbourhood, N_1, is defined by the collection of admissible moves of the first type for the current solution, x. The second neighbourhood, N_2, is defined by the collection of admissible moves of the destination operator. It is worth noting that all the extracted blocks can be swapped from one destination to another without violating any constraints, which makes the second neighbourhood larger than the first. The third neighbourhood, N_3, is defined by a combination of the two previous move operators that will be applied at the same time. The neighbors are the schedules that differ by one block where the period and the destination are different than the actual solution, x. The fourth neighbourhood, N_4, considers swapping the destination of two blocks extracted in the same period, and is defined by a combination of two different move operators for the destination. The union of those neighbourhoods, N_{TS}, is considered to be the global neighbourhood for the search step of the TS algorithm.

3.3 Granular Tabu Search

Granular TS (Toth and Vigo, 2003) is derived from the TS algorithm but allows admissible moves to be discarded from the neighbourhoods during the search step. To reduce the number of moves to compute inside N_{TS}, N_4 is discretized into smaller sub-neighbourhoods to be computed or skipped, depending on how far the current solution is from satisfying constraint (6) for a specific period and destination. Let N_4^{pt} represent the subset of moves from N_4 where the current destination or the new destination is p at period t. If constraint (6), for a specific p,t, is far from equality, that is $d_{os-}^{pt} + d_{os+}^{pt} \gg 0$, then reducing the deviations will be more profitable than swapping blocks, since those variables will only decrease or increase by the difference of tonnage of the two blocks. Thus, the sub-neighbourhood N_4^{pt} will be searched only if $d_{os-}^{pt} + d_{os+}^{pt} \leq \varepsilon_o^{pt}$ for a predefined ε_o^{pt}. The modified neighbourhood N_4', at each iteration, will be the union of the sub-neighbourhoods N_4^{pt} for periods t and processes p that satisfy the above criteria. N_{G-TS} is the entire neighbourhood for the MNTS with granularityand consists of the union of neighbourhood $N_1, N_2, N_3, N_4^{'}$.

3.4 Exploring infeasible solutions

In order to extensively explore the solution space, a mechanism allows the algorithm to explore solutions that do not fully satisfy mining capacity constraints (5). This mechanism is similar to Lamghari and Dimitrakopoulos (2013), where a penalty term is used to quantify how far the current infeasible solution is from feasibility.

$$\max \sum_{t=1}^{T} \left\{ \underbrace{\sum_{i=1}^{N}\sum_{p=1}^{P} E\left\{NPV\right\}_i^{pt} y_i^{pt}}_{Part\,1} + \underbrace{\sum_{s=1}^{S}\sum_{p=1}^{P} \frac{SV_s^{pt}}{S} k_{s-}^{pt}}_{Part\,2} - \underbrace{\sum_{p=1}^{P}\sum_{s=1}^{S} \frac{\left(\left(SV_s^{pt} + \delta^{pt}\right) k_{s+}^{pt} + \eta^t k_{s-}^{pt}\right)}{S}}_{Part\,3} \right.$$
$$\left. - \underbrace{\sum_{p=1}^{P}\sum_{s=1}^{S} d_{os-}^{pt} + d_{os+}^{pt} + d_{qs-}^{pt} + d_{qs+}^{pt}}_{Part\,4} - \underbrace{P^{+}\left(\frac{1}{S}\sum_{s=1}^{S} \max\left\{\left(\sum_{i=1}^{N} w_{is} x_i^t - W^t\right), 0\right\}\right)^2}_{Part\,5} \right\} \quad (13)$$

subject to the constraints (3) to (12).

The algorithm can choose an admissible infeasible solution if the value of the objective function (13) is higher than any admissible feasible solution. The penalty coefficient, P^{+}, varies as a function the number of iterations the algorithm has spent in the infeasible space. As the algorithm select moves that lead to feasible solutions, P^{+} decreases. At some point, those decreases allow moves that do not satisfy constraints (5) to have a higher value in terms of objective value (13) than any move that leads to feasible solution, thus allowing the algorithm to explore the infeasible space. During the search in the infeasible space, the coefficient P^{+} increases, helping the algorithm to return to a feasible solution. The adjustment algorithm starts with $P^{+} = 1$ and, at every h iteration, checks whether all previous solutions have been satisfy constraints (5). If yes, then update $P^{+} = P^{+}/2$, otherwise, set $P^{+} = 2P^{+}$.

3.5 Algorithm

3.5.1 Initial solution

In order to use the MNTS, a feasible solution must be provided. To test the robustness of the method, two different strategies to generate the initial solution have been implemented. The first method uses exact methods to generate a sub-optimal solution. The second method uses a random heuristic.

3.5.2 Initial solution using exact methods

This method is similar to the method to generate initial solution using the exact methods of Lamghari and Dimitrakopoulos (2013), but is adapted for the proposed formulation to include the block destination decisions. The formulation presented in Section 2 is solved sequentially using a MILP solver for each period $t = \{1...T\}$ and considers only the non-extracted blocks that can be mined during this period with respect to constraints (4). As opposed to the other implemented method based on randomness, this algorithm always generates the same initial solution given that it is based on the exact methods and greedy heuristics. The procedure is summarized in Figure 2.

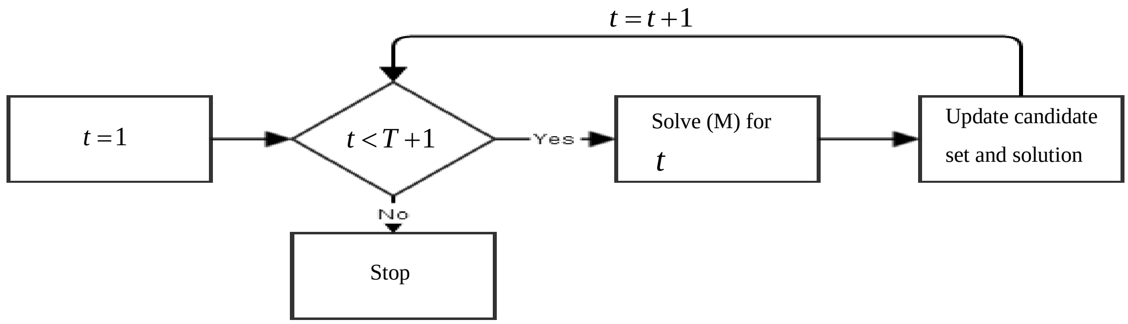

Figure 2: Example of a generic deposit.

3.5.3 Initial solution using a random heuristic

This method produces a feasible schedule that satisfies slope requirements and processing capacities. An iterative process is used to build a feasible schedule, block-by-block, as described in Figure 3. The algorithm starts with the first period and chooses a block randomly from a set of admissible candidates that can be scheduled without violating constraints (3). The block is sent to the process where its NPV is maximized and the algorithm updates the admissible candidate set. If the process has already reached its capacity, then it is sent to the second most profitable, and continues testing all processes until it reaches one with availability. The period is increased when all processes have reached their capacities or the mining capacity is exceeded. The algorithm stops when all the blocks have been scheduled or the period t reaches $T + 1$.

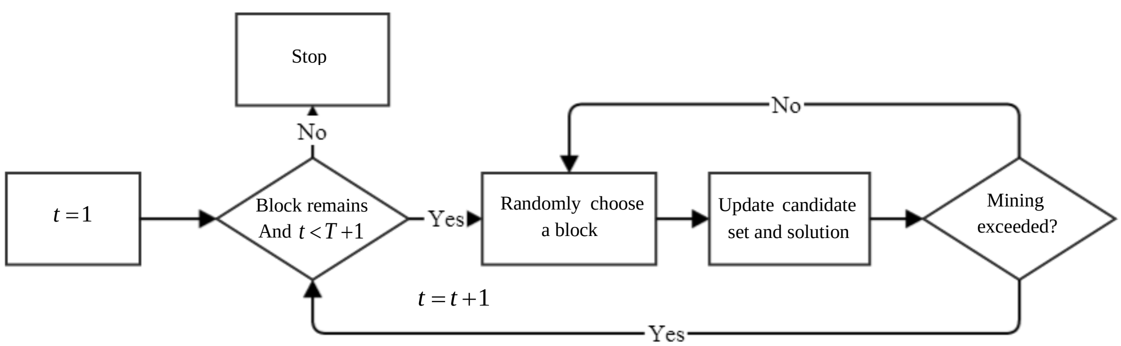

Figure 3: Generating initial solution using heuristics.

3.5.4 Multi-neighbourhood Tabu Search algorithm

The conventional TS algorithm starts with an initial solution and improves it by applying a sequence of admissible moves from a neighbourhood defined by a move operator. It chooses the admissible move that leads to a better improvement over the modified objective function. Once applied, the reverse of the move is added to its respective TL to be stored for a random number of iterations. The TS iteration stops when a successive number of moves without improvement are made (n_{iter}) or every neighbourhood is empty (criteria 2). When one of these criteria is satisfied, a diversification strategy is applied to generate a new initial solution, and the TS iteration is restarted. When the maximum allocated time for the algorithm is reached (criteria 1), the algorithm terminates. The MNTS works exactly as the usual TS, but instead of searching for only one neighbourhood, all of the defined neighbourhoods are searched simultaneously. The algorithm is summarized in Figure 4.

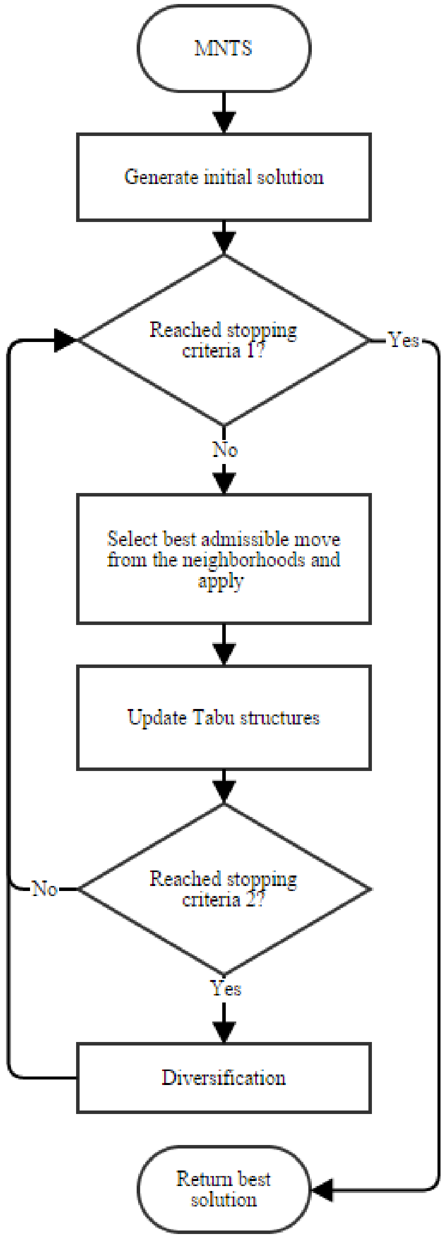

Figure 4: MNTS algorithm.

4 Parallelization of multi-neighbourhoods Tabu Search

The most computationally intensive task during the proposed TS algorithm is searching for the multiple neighbourhoods (MNS) at each iteration. A parallelization strategy is implemented to reduce the time spent searching the neighbourhoods. The search step is embarrassingly parallel, since the computation of each move can be made independently of the others. The parallelization of the MNS is done by the creation of a "pool" of grouped moves called "nodes". A node is a group of moves that belong to the same neighbourhood and share common characteristics. The grouping is made in order to avoid explicit enumeration of all possible moves during the parallel search, which will increase by a large amount the memory needed, and will fasten the updates of the neighborhoods once a move is applied.

Each TST, that runs TS independently, is initialized with the same structures and states. Then each one requests a node, computes the impact on the objective for each move in the node, and subsequently removes the computed node from the pool. The general framework for the proposed parallel algorithm is shown in Figure 5.

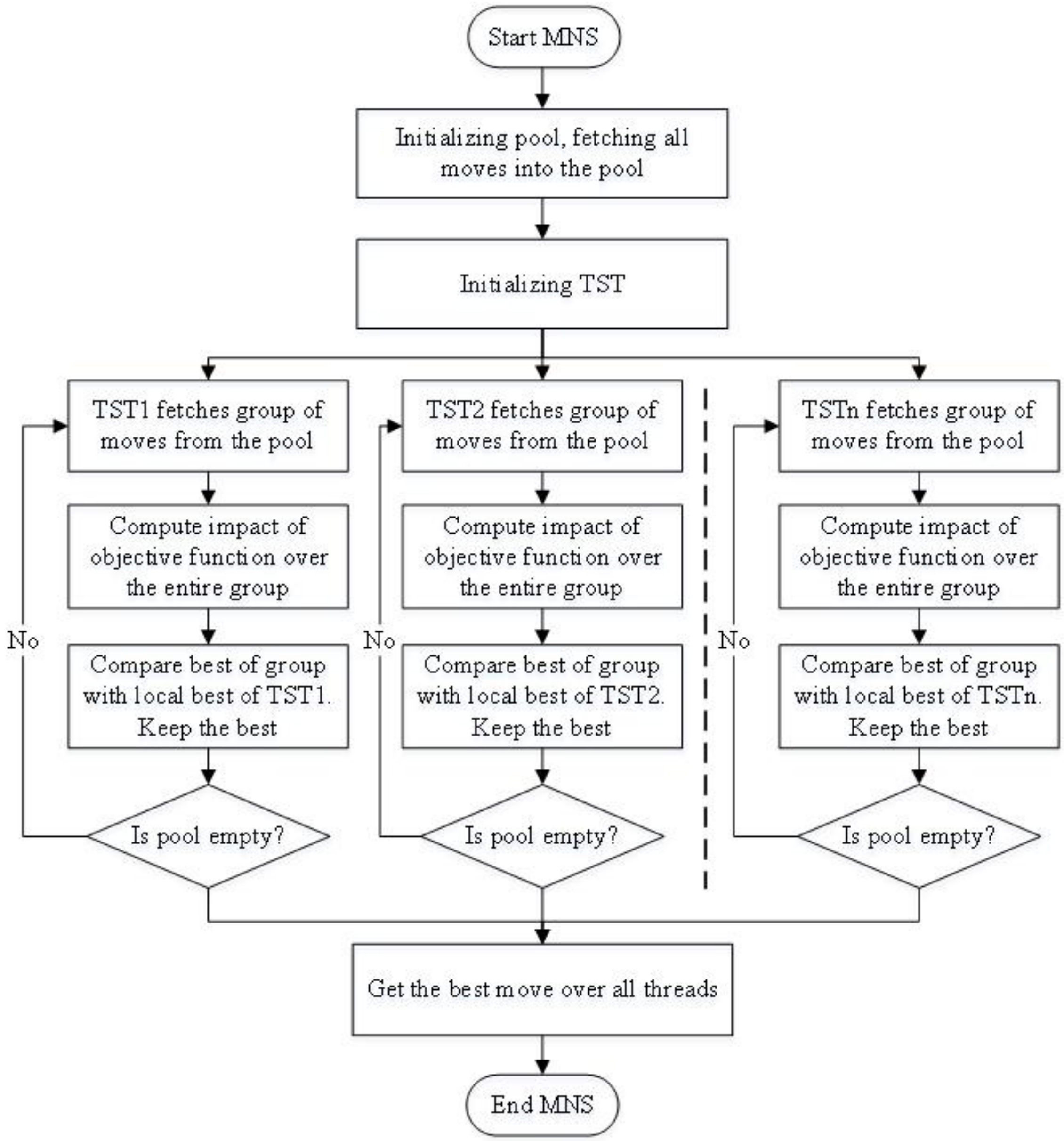

Figure 5: Parallel computation of the multiple neighbourhoods.

A load balancing strategy is designed to define the node to be used by neighborhood, in order to avoid that a thread is working in a very large portion of the neighbourhood while the other threads are waiting because the pool is empty. For the first neighbourhood, N_1, a node consists of one block and all possible periods where it can be swapped. For the second neighbourhood, N_2, the node consists of one block and all its possible destination where it can be swapped to. The node of the third neighbourhood, N_3, is composed by one block and its admissible periods for a fixed destination to be swapped. The node of the fourth neighbourhood, N_4, contains P different moves associates with a same block where the process will be swapped.

5 Numerical results

In this case study, the efficiency and the quality (in terms of optimality) for the proposed algorithm is presented. Following this, the robustness of the method is demonstrated by comparing the solution generated using two different initial solution methods. In this case study, three of the ten instances (small and medium) used in Lamghari and Dimitrakopoulos (2013) are used to assess the quality of the solution generated and the efficiency of the algorithm against a commercial solver, in this case CPLEX v12.5.1. Finally, an application at a large deposit is used to assess the ability of the proposed algorithm to generate solutions for deposit of realistic size. The parameters are given in Table 1.

Table 1: Model parameters.

Set	Problem	Number of blocks (N)	Number of periods (T)	Number of scenarios (S)	Number of processes (P)	Number of integer variable
P1 (Copper)	C1	4,273	3	20	3	51276
	C3	12,627	7	20	3	353556
	C4	20,626	10	20	3	825040
P2 (Gold)	G1	119,805	25	14	3	11980500

In order to compare the different instances, a time linearly proportional to the number of blocks, processes and number of periods is used to define criteria 1, $T_{Z_{best}}$, i.e. $T_{Z_{best}} = 0.015 \cdot P \cdot N \cdot T$. Table 2 shows the economic parameters for both data sets.

Table 2: Economic parameters and processing.

Parameters		P1	P2
Block Size (m)		20x20x10	10x10x5
Block weight (t)		10,800	-
Metal Price		3747.85 $/t	25.75 $/g
Selling Price		881.85 $/t	0.176 $/g
Mining Cost		1.00 $/t	1.50$/t
Discount rate (d)		10%	8%
Geological discount rate (r)		20%	20%
Processing Cost	Mill	9.00 $/t	9.50 $/t
	Leaching	2.25 $/t	5.00 $/t
	Waste Dump	Included in mining cost	Included in mining cost
Mill - Stockpile	Rehandling Cost	0.70 $/t	1.35 $/t
Recovery	Mill	90%	90%
	Leaching	55%	50%
	Waste Dump	0 %	0%
Capacities:	Mill	5 Mt	15 Mt
	Leaching	2.5 Mt	5 Mt
	Waste Dump	$+\infty$	$+\infty$

5.1 Quality of the solutions

The program was compiled using VC++ 2013 and run on a Xeon processors 5500 with 24 GB of DDR3 RAM running on Windows 7 x64, using OpenMP as the API for parallelization. To solve the relaxation of (1) subject to (2) to (13) to obtain an upperbound, Z^*_{LR}, the commercial solver CPLEX is used. In the remainder of this paper, a –E will be appended to the name of a solution that was generated using "exact" method described in Section 3.5.1.1 for the initial solution, and a –H will be appended to the one using the starting solution as described in Section 3.5.1.2. Thus, for example, C1-H refers to the first instance when starting with an initial solution generated by the method described in Section 3.5.2.2. Z_{best} refers to the best solution obtained by the developed method. In order to test the quality of the solution, the value obtained by the algorithm for different instances is compared against the linear relaxation value for the same instance. The gap is a metric used to show how far the solution is obtained in terms of the linear relaxation value, and is computed using

$$GAP\,(\%) = \frac{Z^*_{LR} - Z_{best}}{Z^*_{LR}} \times 100. \tag{14}$$

Table 3 summarizes the results for the different starting solutions and the final objective function values. For small instances, the MNTS provides a near-optimal solution; however, as the size of the instance increases, the gap also increases. Table 3 also shows that for the small instance, it can produce a solution with less than a 1% gap in less time than CPLEX can solve the linear relaxation. The method seems to work better when it is started with a good initial solution (Initial Gap), but this initial solution method is computationally more expensive. Since the total time allowed by the P-MNTS is limited, it is possible that in some cases, the total time is spent generating the initial solution with the exact methods, rather than improving it. The table also shows that starting with any of the two initial solution methods, the algorithm is able to improve the quality of the solution to a near-optimal solution.

Table 3: Robustness of the method.

Problem	Initial solution	T_{init} (s)	Z_{best} (M$)	$T_{Z_{best}}$ (min)	Initial gap (%)	$T_{Z^*_{LR}}$ (min)	Gap (%)
C1	E	6.59	112.65	6.00	4.96	0.25	0.55
	H	0.011	112.02	6.00	117.05	0.25	1.12
C2	E	469.94	175.88	31.7	37.98	139.05	0.98
	H	0.056	166.07	31.7	235.85	139.05	6.52
C3	E	1175.85	213.15	61.7	38.86	965.37	6.76
	H	0.11	209.69	61.7	304.77	965.37	8.27

5.2 Quality of the parallel algorithm

The gains made by adding the parallelization to the proposed algorithm is summarized in Figure 6.

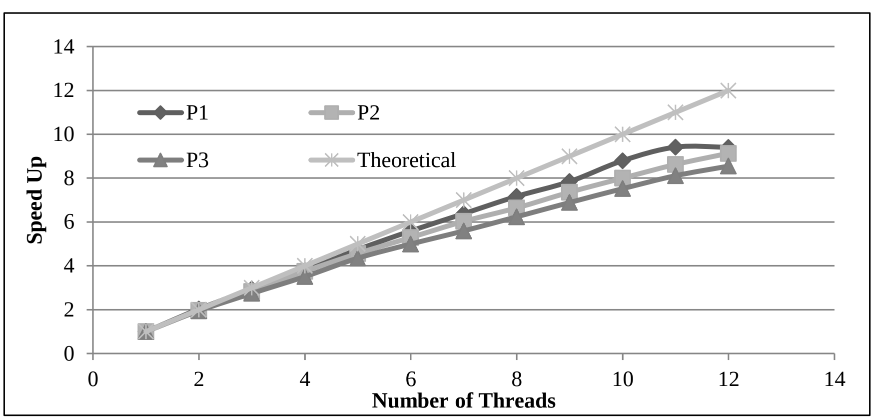

Figure 6: Efficiency of the parallelization.

Since the search step, where the algorithm computes the impact over the objective function for each move, is embarrassingly parallelizable since each move is independent, the theoretical speed up should be linear with respect to the number of threads, given that each node needs the same time to be computed, i.e. $SpeedUp = (n_{nodes}/(n_{nodes}/N_{Threads})) = N_{Threads}$, where n_{nodes} is the number of total nodes to be computed and $N_{Threads}$ is the number of threads. For a small number of threads, the gain is almost identical as the theoretical that can be achieved to parallelize the multi-neighbourhood search; however, as the number of threads increases, the gain looks to be lower than the theoretical. Overall for the proposed implementation, the search is executed nine times faster using twelve threads than using a single threading on all the different instances.

Figure 7 highlights the impact of using the parallelization approach over the final NPV as compared to using a single thread for the MNTS. The parallelization addition have a higher impact when starting with an initial solution generated using random heuristic, since the initial gap, using this method, is bigger than the one using exact method. Since the algorithm needs more computational time to go from the initial solution to a good-quality solution in the first case, the multi-threading brings a higher improvement over the final NPV than when starting with a near-optimal solution already, where the overall improvement is less. Figure 7 also shows that for any case, the multi-threading provide an improvement over the project value, meaning that the objective of searching more extensively the search space is met. The improvement gained from a multi-threaded approach permits the algorithm to perform more iterations before reaching criteria 1, thus improving the final solution in any case.

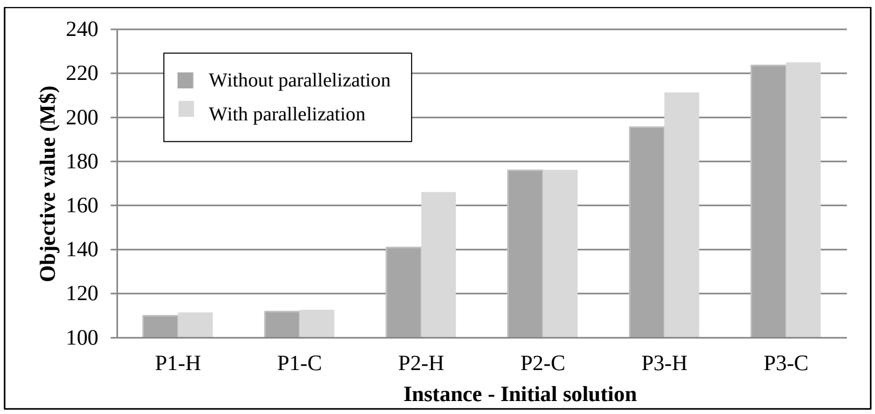

Figure 7: Impact of parallelization over the objective value.

5.3 Application at a gold deposit

The effectiveness of the proposed approach is also demonstrated through an example for a gold deposit (P2). As seen in Table 1, the resulting SIP has almost 12 million variables, thus the relaxed SIP is too large to be solved in a reasonable amount of time and memory. The gap over the final solution can therefore not be obtained, thus, a risk profile is generated to assess the quality of the solution in practical terms. In this case study, only the ore target constraints (6) and mining capacity (5) are considered, and the penalties associated with the metal production targets (7) are set to 0. For the leaching process, only quantities that exceed an upper-bound are penalized, and there is no target associated with the waste dump. Figure 8 shows the ore tonnage associated with each process. There is no risk of not meeting this tonnage since the tonnage is assumed equal for every scenario, however, since the destination of a block is scenario independent, the optimizer can send waste to process to fulfill the target. Thus, variability can be seen if looking at the tonnage above/below marginal cut-off grade. The graph shows that the ore target is met and constant until the last year for the mill and that the capacity of the leach plant is not exceeded. The waste tonnage is almost constant over the life of mine, but a big drop in tonnage occurs in year 23. This is justified by the use of the stockpile, as seen in Figure 10 where at year 23 the stockpile content goes to 0.

Figure 9 shows the average grade of the material sent to each process over the life-of-mine. The more valuable material is sent at the beginning of the mine life and decreases with time. Figure 9 shows a low variability for the quantity of metal for each process, meaning that the expected NPV will have a low risk of not being met, as seen in Figure 1. Figure 9 also shows that the expected average grade in the leach pad is

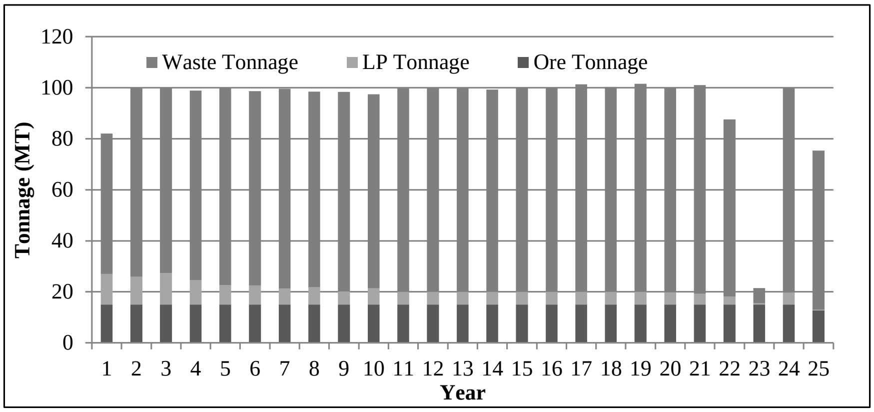

Figure 8: Ore tonnage graph.

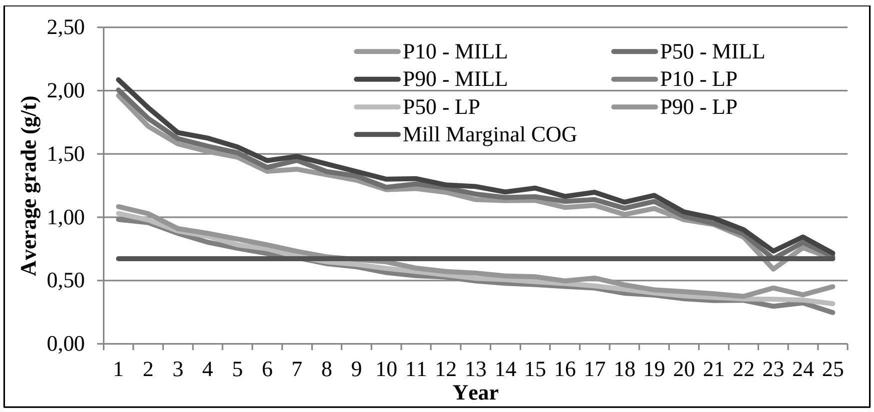

Figure 9: Average grade.

higher than the cut-off for the first six periods, where it is actually more profitable to send material to the mill instead of the leach pad. This has a direct impact on the NPV because the valuable material is recovered sooner, thus increases the NPV in the initial years.

The effect of the average grade assumption for the stockpile made in the proposed model can be directly seen in Figures 8, 9, and 10. The decrease of waste material by period 23 can be explained by fact that the proposed optimizer is not able to extract material with the quality as high as the one of the fixed stockpile grade. It can also be explained by the fact that the optimizer do not need to pull out the waste to access the ore. Thus, it is worth to empty the stockpile earlier than at the last period since the cost of mining waste and the profit made by processing lowest quality material later are decreasing the NPV for this year. There is no risk in tonnage of the stockpile because the mining and processing decisions are first stage decisions and blocks have the same tonnage in all scenarios. The tonnage mined, processed, and stockpiled are then the same in every scenario.

Figure 11 shows the cumulative NPV over the life of mine, which has been scaled for confidentiality. The figure shows that more than 80% of the NPV is generated before the first 10 years of the life of mine. It also shows that the variability in term of NPV is very low at the beginning, and with the low risk of not meeting production target, the NPV have a low risk of not being met.

6 Conclusions

In this paper, a mathematical formulation that uses profit to define the value of a block in each process, where the destination of the block is decided, regardless of the scenario, is presented. The method proposed

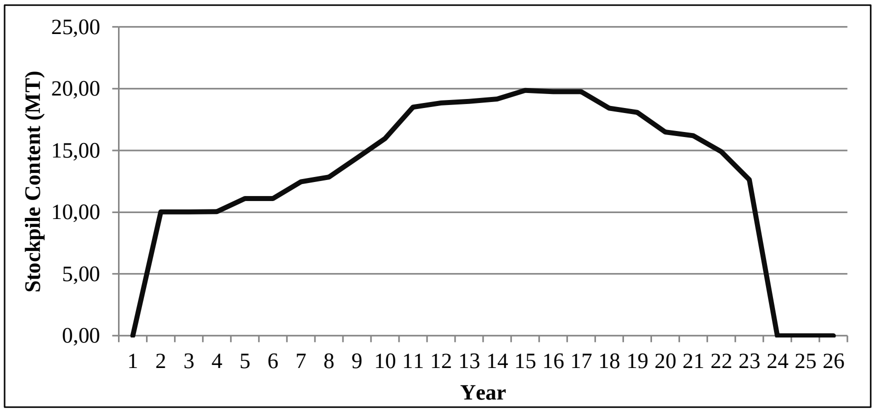

Figure 10: Stockpile content.

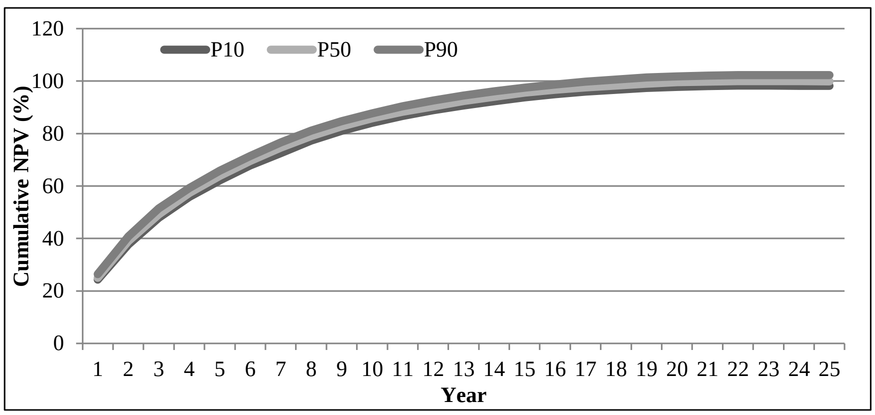

Figure 11: Cumulative NPV.

by Lamghari and Dimitrakopoulos (2013) has been extended to consider stockpiles and block destinations. A parallelized multi-neighbourhood Tabu Search algorithm has been implemented to efficiently provide a good-quality solution in reasonable amount of time. This method simultaneously searches to change a block‘s extraction period, the destination, the period and the destination of a block at once, or swapping the destination of two blocks. A granular approach is used to constrain the multiple neighbourhoods to only promising regions based on the processing capacity constraints. The method is applied on small- and medium-scale datasets. Results show that the proposed method generates solutions that are near-optimal and that the use of parallelism may improve the value of the project within the same time. An application at a realistic-sized deposit shows that the method is able to provide solutions and the risk profile assesses the ability of the method to meet production targets. Further work may include the ability to consider stockpiles as a destination, as in Ramazan and Dimitrakopoulos (2013), which considers more general blending constraints associated with multiple elements and applies this method to multi-element deposits.

7 References

Baker, C., and Giacomo, S. (1998). Resource and reserves: Their uses and abuses by the equity markets. In proceedings, Ore reserves and finance: A joint seminar between Australasian Institute of Mining and Metallurgy and ASX, Sydney.

Birge, J.R., and Louveaux, F. (2011). Introduction to stochastic programming. Springer, 2nd Ed.

Bley, A., Boland, N., Fricke, C., and Froyland, G. (2010). A strengthened formulation and cutting planes for the open pit mine production scheduling problem. Computers & Operations Research, 37(9), 1641–1647.

Bley, A., Boland, N., Froyland, G., and Zuckerberg, M. (2012). Solving mixed integer nonlinear programming problems for mine production planning with stockpiling. Technical Report 2009/21, Institute of Mathematics, TU Berlin.

Cacetta, L., and Hill, S.P. (2003). An Application of Branch and Cut to Open Pit Mine Scheduling. Journal of Global Optimization, 27, 349–365.

Dagdelen, K., and Johnson, T.B. (1986). Optimum Open Pit Mine Production scheduling by Lagrangian Parameterization. In proceedings 19th International Symposium 1986 – Application of Computers and Operations Research, 127–142.

David, M. (1988). Advanced Applied Geosttistical Ore Reserve Estimation. Elsevier.

Dimitrakopoulos R., Farrelly C., and Godoy, M. (2002). Moving forward from traditional optimisation: Grade uncertainty and risk effects in open pit mine design. Transactions of the IMM, Section A Mining Industry, 111, A82–A89.

Dowd, P. (1997). Risk in minerals projects: perception, analysis and management. Trans. Institude of Mining and Metallurgy, 106, 9–18.

Gershon, M. (1983). Mine Scheduling Optimization With Mixed Integer Programming. Mining Engineering, 35, 351–354.

Goovaerts, P. (1997). Geostatistics for Natural Resources Evaluation. Oxford University Press.

Glover, F., and Laguna, M. (1997). Tabu Search. Kluwer Academic Publishers.

Jin, J., Crainic, T.G., and Løkketangen, A. (2012). A parallel multi-neighbourhood cooperative Tabu search for capacitated vehicle routing problems. European Journal of Operational Research, 222(3), 441–451.

Journel, A. (1974) Geostatistics for Conditional Simulation or Ore Bodies. Economic Geology, 69(5), 673–687.

Lamghari, A., and Dimitrakopoulos, R. (2013). A diversified Tabu search for the open-pit mine production scheduling problem with metal uncertainty. European Journal of Operational Research, 222(3), 642–652.

Lamghari, A., Dimitrakopoulos, R., and Ferland, J.A. (2014). A variable neighbourhood descent algorithm for the open-pit mine production scheduling problem with metal uncertainty. Journal of the Operational Research Society, 65, 1305–1314.

Ramazan, S, 2001. Open pit mine scheduling based on fundamental tree algorithm, PhD thesis, Colorado School of Mines, Golden.

Ramazan, S., and Dimitrakopoulos, R. (2007). Stochastic optimisation of long-term production scheduling for open pit mines with a new integer programming formulation. Orebody Modelling and Strategic Mine Planning, Spectrum Series, 14, 359–365.

Ramazan, S., and Dimitrakopoulos, R. (2013). Production scheduling with uncertain supply: a new solution to the open pit mining problem. Optimization and Engineering, 14(2), 361–380.

Ravenscroft, P. (1992). Risk analysis for mine scheduling by conditional simulation. Trans. Institute of Mining and Metallurgy, 101, 104–108.

Tachefine, B., and Soumis, F. (1997). Maximal closure on a graph with resource constraints. Computers & Operations Research, 24(10), 981–990.

Tolwenski, B., and Underwood, R.(1996). A scheduling algorithm for open pit mines. IMA Journal of Mathematics Applied in Business & Industry, 7, 247–270.

Toth, P., and Vigo, D. (2003). The Granular Tabu Search and its application to the vehicle-routing problem. INFORMS Journal on Computing, 15(4), 333–346.

Vallee, M. (2000). Mineral resource + engineering, economic and legal feasibility = ore reserve. CIM Bulletin, 93, 53–61.

Wu, Q., Hao, J. K., and Glover, F. (2012). Multi-neighbourhood Tabu search for the maximum weight clique problem. Annals of Operations Research, 196(1), 611–634.