Mine Production Scheduling through Heuristic
Memory Based, Improved Simulated Annealing

Yuksel Asli Sari
Department of Mining and Materials Engineering
McGill University, Montreal, Quebec
November, 2014

A thesis submitted to McGill University in partial fulfillment
of the requirements for the degree of MEng

©Yuksel Asli Sari, 2014
Abstract

Because of high value added potential, mine production scheduling has been an attractive topic in mining engineering area for 50 years. Mine production scheduling is essentially a set of problems: block sequencing, ore-waste discrimination and production rates. Previously the machine learning technique Simulated Annealing (SA) was used to solve the classic mine production scheduling problem together with Lerchs-Grossman method. We are taking this approach one step forward to solve poly-metallic, multiple process destinations and multiple rock types mine production scheduling problem using the equivalent grade factors. A transportation penalty approach was taken to smooth the fluctuating structure that emerges as a result of SA’s nature and to unify the same periods to decrease transportation costs. Moreover, a novel, heuristic memory based SA technique is proposed, where a foreseen variable is nominated to maximize the objective function and this variable’s questionable contribution is evaluated by distribution fitting to online gathered data points. If indeed this variable is influencing the objective function, its effect is boosted to also maximize this variable. SA with heuristic based memory converges faster to the solution and produces better results compared to original SA technique within the same frame of time. Success of the approach is demonstrated by two case studies, comparing the revenues of the base case (ranked positional weight algorithm), standard SA and heuristic memory based SA. In all cases, NPV is higher with the heuristic memory based SA.
Résultat du fort potentiel, la planification de la mine a été un sujet attrayant dans le secteur du génie minière depuis 50 ans. Planification de la mine est essentiellement un ensemble de problèmes : le séquençage de bloc, la discrimination minerai déchets et les taux de production. Auparavant, la technique d’apprentissage machine Simulated Annealing (SA) a été utilisé pour résoudre le problème classique de la planification de la mine avec la méthode Lerchs-Grossman. Nous prenons cette approche un pas en avant pour résoudre poly-métallique, plusieurs destinations de processus et le problème de types de roches production minière d’ordonnancement multiples en utilisant les facteurs de qualité équivalentes. Une approche de la sanction de transport a été prise pour lisser la structure fluctuante qui émerge à la suite de SA’s nature. En outre, une nouvelle technique basée SA mémoire heuristique est proposé, où une variable prévu est nommé pour maximiser la fonction objectif et la contribution douteuse de cette variable est évalué en ajustant à la ligne des points de données recueillies distribution. En effet, si cette variable influence la fonction objectif, son effet est renforcé pour maximiser également cette variable. SA avec de la mémoire en fonction heuristique converge plus rapidement à la solution et produit de meilleurs résultats par rapport à la technique originale SA dans le même laps de temps. Le succès de l’approche est démontré par deux études de cas, comparant les revenus de l’affaire de base (algorithme classé position-poids), basée SA standard et mémoire
heuristique, SA amélioré. Dans chaque cas, la NPV est supérieure à la mémoire heuristique basée SA.
Acknowledgements

I would like to thank Prof. Mustafa Kumral for giving me this opportunity, his invaluable guidance and patience.

This thesis would not have been possible without the encouragements and supports of my brother and my parents. I would like to extend my gratitude to Koray, who was always there for me; Gizem and Gizem for being the greatest friends in the world and my other friends for their presence. There are no words to convey how much they mean to me. Special thanks to my brother for translating the abstract and Tuna, for helping me draw the figures.
Contributions of Authors

The author of this thesis is the primary author. Professor Mustafa Kumral was the supervisor of the author’s Master of Engineering degree and co-authored the paper “Mine Production Scheduling For Poly-Metallic Mineral Deposits: Extension to Multiple Processes” which originated from this research, is under review.
Contents

1 Introduction ................................................. 1
    1.1 Problem Statement ..................................... 1
    1.2 Research Objectives .................................. 2
    1.3 Originality and Success ............................... 3
    1.4 Social Impacts and Economic Benefits ............... 5
    1.5 Thesis organization .................................. 6

2 Open Pit Mine Production Scheduling ...................... 8
    2.1 Optimal Capacity ...................................... 10
    2.2 Cut-off Grades ........................................ 14
        2.2.1 Cut-off Grade for Multiple Destinations ...... 18
        2.2.2 Cut-off Grade for Multi-Metal Deposits ...... 19
    2.3 Block Sequencing ...................................... 20
        2.3.1 Heuristics ........................................ 21
        2.3.2 Extraction sequencing methods based on the pa-
              parameterization of an ultimate pit method .. 26
        2.3.3 Exact methods .................................... 31
        2.3.4 Dynamic programming ............................ 38
        2.3.5 Metaheuristics ................................... 40

3 Approach to Solve Open Pit Production Scheduling Problem 45
    3.1 Ranked Positional Weight Algorithm ................. 46
3.1.1 Finding Block Economic Values ........................... 46
3.1.2 Calculating Ranked Positional Weight ..................... 46
3.1.3 Period Classification ...................................... 48

3.2 Simulated Annealing Applied to Mine Production scheduling . 49
3.2.1 Calculating Mine Economic Value .......................... 49
3.2.2 Classification of Transition Destination Blocks ............... 52
3.2.3 Classification of Transition Period Blocks ................... 53
3.2.4 Simulated Annealing for Mine Production scheduling ... 53

3.3 Program Highlights ........................................... 56
3.3.1 Annealing Parameters ...................................... 56
3.3.2 Mine Economic Value Calculation ......................... 57
3.3.3 Handling Topography ....................................... 59

4 Case Studies ....................................................... 62
4.1 Case Study 1 .................................................. 63
4.2 Case Study 2 .................................................. 71

5 Improving Simulated Annealing by Adding Heuristic Memory ... 79
5.1 Introduction ................................................... 79
5.2 Method ....................................................... 82
5.3 Case studies extended to Heuristic Memory Based Simulated Annealing ........................................... 84
List of Figures

1.1 Interdependent sub-problems of mine production scheduling . 2
1.2 Interdependent characteristics of mine production scheduling . 4
2.1 A view of 3D Block model [Sevim and Lei, 1998] . . . . . . . 8
2.2 Result of different cut-off grades on NPV [Hustrulid et al., 2013]. 17
2.3 Possible multiple destinations of a deposit . . . . . . . . . . 19
2.4 Block combinations in which five and nine overlying block
   must be extracted to mine block previously, respectively [Khalokakaie,
   1999]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
2.5 Demonstration of ranked positional weight technique scoring . 25
2.6 A labeled block model transformed into a directed graph [Hus-
   trulid et al., 2013] . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
2.7 Labeling [Hustrulid et al., 2013] . . . . . . . . . . . . . . . . 29
2.8 Parameter space . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
3.1 Two types of slope angles . . . . . . . . . . . . . . . . . . . . . . 60
4.1 Cross-section (Sections: 48, 52, 0) . . . . . . . . . . . . . . 67
4.2 Cross-section (Sections: 56, 65, 0) . . . . . . . . . . . . . . . . 68
4.4 Average grades and number of blocks in each destination and
   period . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
4.4 Average grades and number of blocks in each destination and
   period (continued) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
4.5 Overall average grades compared among low, medium and high grade process destinations

4.6 Cross-section (Sections: 23, 0, 0)

4.7 Cross-section (Sections: 22, 46, 0)

4.8 Average grades and number of blocks in each destination and period

4.9 Overall average grades compared among low and high grade process destinations

5.1 Example 1: Correlation of the number of extracted blocks to NPV

5.2 Example 2: Correlation of the number of extracted blocks to NPV
List of Tables

2.1 Labeling the directed graph ............................................ 28
4.1 Parameter File for Case Study 1 ...................................... 64
4.2 Comparison of resultant NPVs ......................................... 65
4.3 Parameter File for Case Study 2 ...................................... 73
4.4 Comparison of resultant NPVs ......................................... 73
5.1 Case 1: Comparison of resultant NPVs .............................. 85
5.2 Case 2: Comparison of resultant NPVs .............................. 86
Abbreviations

ACO  Ant Colony Optimization

MCAF  Mining Cost Adjustment Factor

NPV  Net Present Value

PCAF  Processing Cost Adjustment Factor

SA  Simulated Annealing
1 Introduction

1.1 Problem Statement

A significant amount of research have been devoted to mine production scheduling and reasonable progresses have been achieved over the last 40 years and many commercialized products have been released. At the same time, high grade deposits have been depleting, extraction costs have swiftly increased, environmental concerns have imposed additional costs and competition between mining companies became tougher. Therefore, more efficient use of mineral deposits are becoming more and more important. One way to increase efficiency is to deal with shortcomings associated with mine production scheduling.

In current applications, optimal production rates, ore - waste discrimination and block sequencing problems are solved in a sequential way (Figure 1.1) despite these problems being interdependent; in other words, one sub-problem cannot be solved if the other is not solved previously. In common applications, one of the problems is treated as it is solved by assuming a predicted result and the other is solved under this assumption. This leads to sub-optimality. This research attempts to develop an approach that solves ore-waste and block sequencing problems simultaneously. Thus, (i) harmony between mining and mineral processing process can be ensured, (ii) Net present value NPV can be increased and (iii) mineral deposit can be used
1.2 Research Objectives

The objective of this research is to develop a new mine production scheduling approach based on simulated annealing with heuristic memory such that ore - waste discrimination and block sequencing sub-problems are simultaneously optimized. The addition of a heuristic memory improves simulated annealing method (SA) and a more intelligent search is conducted. The research also aims to extend to more complex problems such as poly-metallic ore bodies and multiple processes cases. Poly-metallic reserves are preferred because poly-metallic orebodies aid the risk diversification; one metal can support the other in reaction to financial fluctuations.
To summarize, this thesis aims to develop an efficient SA approach to solve ore-waste discrimination and block sequencing concurrently in such a way as to consider mining operations with poly-metallic and multiple destinations cases.

1.3 Originality and Success

In mining industry, economies of scale rule applies. As production rates and capacities increase, average cost of production decrease. If capacities are not known, production costs cannot be known. However, Lerchs and Grossmann approach that many commercial software are based on has no capacity constraint. Therefore, production costs used in Lerchs and Grossmann are quite arbitrary. This can result in unrealistic NPV. Furthermore, ore-waste discrimination is a function of production costs. Since production costs are not known, ore-waste discrimination cannot be reasonably fulfilled. Finally, if ore and waste are not discriminated, blocks cannot be sequenced such that capacities are satisfied. Interdependence among variables is illustrated in Figure 1.2.

Since Lerchs and Grossmann cannot address the interdependent characteristics of mine production scheduling, SA is re-visited. SA has been used in mine production scheduling for 15 years and its shortcoming has been observed especially in the speed of convergence. The originality rests on:

(i) A more efficient SA module is developed by adding a heuristic memory.
This enables SA to conduct more efficient search such that the better results can be found in the same amount of time.

(ii) Ore - waste and block sequencing problems are solved concurrently.

(iii) Effective block destination and period transition (perturbation) mechanisms are proposed such that the feasible solutions are found at all iterations.

(iv) The approach proposed can generate two solutions: (a) maximization of NPV and (b) maximization of capacity utilization. These two objectives
are governed by weights. While a weight value of 0 is maximizing NPV, a weight value of 1 is maximizing capacity utilization. The deviations from capacities are handled through adding opportunity cost term into the objective function.

(v) The technique proposed is able to quantify goodness of a mine production schedules. SA’s initial solution is provided by the positional weight algorithm, which is equivalent to Lerchs and Grossman’s parameterized approach in the sense that they both focus on optimizing the NPV of the pit and both of them cannot handle the process capacity constraint optimally. Comparing SA’s initial solution and final solution yields us the improvement that SA introduces.

1.4 Social Impacts and Economic Benefits

Over last 30 years, we witnessed significant changes in science, mining, industry and technologies:

(a) Focus on computer and operations research technologies have swiftly increased.

(b) Mineral deposits having high grades and easy geologic conditions have decreased.

(c) Financial instruments and capital markets have become more sophisticated. Therefore, risk in mining business emerges as a premier concern
in mining business. This forces the mining players to consider complex deposits and processes.

(d) Mining industry is highly globalized thus the competition has grown due to international exposure.

(e) Sustainability is now an inevitable part of mining operations. Environmental compliance is a priority of mining operations.

These changes put a specific emphasis on mine production scheduling in relation to more effective and more profitable mining practices. Given that the depletion of resources is also important concern, the resources should be used more efficiently. Mine production scheduling incorporating new computer and operations research technologies will be value adding component of the operations.

1.5 Thesis organization

This thesis is divided into 6 sections:

Section 1 focuses on the problem being researched, its original contributions and its scientific and economical relevance.

Section 2 reviews the previous research devoted to mine production scheduling.
Section 3 describes the approach taken to solve mine production scheduling problem through combining a sub-optimal solution and SA.

Section 4 demonstrates the schedule and the improvement that this combined solution brings compared to the positional weight algorithm.

Section 5 explains in detail how SA is improved by adding heuristic memory.

Section 6 concludes by discussing the general advantages and disadvantages of the approach and how can it be improved through future work.
2 Open Pit Mine Production Scheduling

Starting in 1960, open pit design and scheduling problem has been attempted to be solved using mathematical large-scale optimization techniques in theory. Implementation of these techniques in computers to be tested in real-world data started in 2000s, because of the computer hardware limitations before this time.

The ultimate goal is to determine the extraction sequence of the blocks that will maximize the net present value (NPV) and decide if it is feasible to proceed with mining. By modifying the access constraint, this approach can be adapted to underground mines as well.

Figure 2.1 – A view of 3D Block model [Sevim and Lei, 1998]
In Figure 2.1, we can see a geologic block model, where a reserve is divided to blocks that contain geologic model of similar content. Block sizes are determined by various factors such as equipment size, geology, data spacing, bench geometry and blasting pattern selection. Blocks are assigned grades based on the outcome of kriging technique, which will be used as ore-waste discriminator in the second step of mine production scheduling [Dagdelen, 2001].

Amankwah [2011] states that if the block values and the slopes are fixed, it is possible to determine an optimal outline. As block value increases, the optimal pit grows because it will yield a higher NPV, and an increase in the slopes will make the optimal pit grow deeper. Thus, it is very important to set the block values and slopes as accurate as possible, otherwise the calculated optimal pit will be incorrect. However, for an optimal calculation, we need to know the pit outline and the mining sequence to accurately know the block value and likewise, we need to know the block value to calculate the optimal outline [Amankwah, 2011]. This creates a chicken and egg problem. This is solved by first assuming that a block is uncovered and it will be mined, then assuming that if mining were stopped, any ongoing cost would stop should be included [Whittle, 1990].

Mine production scheduling consists of three major steps. First step is sequencing, where it is decided the time period a block will be extracted. The step of deciding which destination this extracted block will be sent to is
called optimal cut-off or ore-waste discrimination. In this step, a cut-off grade is decided based on calculating the grade that balances only the expenses and costs. All blocks above this grade are considered as ore and below this grade are considered as waste blocks. The last step is deciding on the production rate; how much material will be extracted in each period [Dagdelen, 2001].

2.1 Optimal Capacity

Mine production scheduling is a decision making process concentrating on the determination of production rates, ore-waste discrimination and block sequencing. A mine schedule should be seen as a blueprint in the sense of understanding whether the project is profitable. As such, the planning procedure is strategic rather than operational. However, mine production scheduling also aids to the decision making process of grade control, blending and processing design. It can be used by mine company managers, financial analysts, shareholders, and planners/schedulers. Practically, the capacity is first decided depending on the mining and mineral processing limitations. Having decided on the capacity, operation costs are calculated [Kumral, 2013] [Kumral, 2012] [Kumral, 2010].

In a mining operation, there are at least two production rates: Mining and mineral processing. Of course, the operation may have multiple waste and processing destinations. The determination of production rates is a more complex task compared to other industries. In many industries, production
rates are determined on the basis of product demand because there will not be the problem on the availability of raw material [Smith, 1997].

The difficulty of estimating the production rate is associated with three reasons:

(i) There is a slope (access) constraint. It means that you cannot access the material as you wish. To access a parcel, overlying materials must be removed previously. Therefore, there would be fluctuations in ore production, hence mineral processing production rates.

(ii) The ore material and its quality are not homogenous. As the heterogeneity in terms of grade and ore quantity increases, ore material availability will fluctuate throughout the periods.

(iii) The cut-off grade is a function of price, processing cost and recovery. All these variables are dynamically changing over time due to market structure and ore characteristics. The cut-off grade and hence, ore-waste amounts are subject also change. The production rates should be compatible with cut-off changes.

Difference between capacity and production rate will be an opportunity cost to the mining company. Capacities and corresponding production rates should ideally be equal. In this case, opportunity cost will be zero. This problem will be more complicated for multiple destination cases. In an ore-body, even though the quantity of material and associated grade are most
likely unevenly distributed, the supply should still satisfy qualitative and quantitative requirements.

Dowd [1994] and Dowd [1976] reported a work on optimal capacity as a function of production rate or mine life. Optimal profit is a function of the number of production periods $N$, the yearly production $t$ and the mean grade and the ore reserve $T(m)$ at this grade. If it is assumed that the profit is a function of the number of production years, there will be a constant ore reserve for a given cut-off grade and the optimal life of mine can be determined. According to Dowd [1976] the investment function can be expressed as:

\[
I(t) = C_0 + C_1 t
\]  \hspace{1cm} (2.1.1)

Where $C_0$ and $C_1$ is the fixed and variable investment costs, respectively.

This investment is depreciated over the life of mine:

\[
I(t) = (C_0 + C_t) \frac{i}{((1 + i)^N - 1)}
\]  \hspace{1cm} (2.1.2)

A profit function for life of mine can be given as:

\[
B(N) = b_0R - a_1N - C_0N \frac{i}{((1 + i)^N - 1)} - C_1N \frac{i}{((1 + i)^N - 1)}
\]  \hspace{1cm} (2.1.3)
Where B is profit, \( b_0 \) is price, \( R \) is metal quantity, \( a_1 \) is extraction costs, \( N \) is total tonnage to be extracted.

If \( \alpha = \frac{i}{(1+i)^N - 1} \) \hspace{1cm} (2.1.4)

\[
B(N) = b_0R - a_1N - C_0N\alpha - C_1N\alpha
\] \hspace{1cm} (2.1.5)

Given that the objective is to determine optimal number of years, \( N_m \).

Differentiating \( B \) with regard to \( N \) and setting the equation to zero.

\[
\frac{\partial B}{\partial N} = -\alpha_1 - C_1R\frac{\partial \alpha}{\partial N} - C_0N\frac{\partial \alpha}{\partial N} - C_0\alpha = 0
\] \hspace{1cm} (2.1.6)

Where

\[
\frac{\partial \alpha}{\partial N} = \frac{-\ln(1+i)}{i}\alpha(1+\alpha), \hspace{1cm} (2.1.7)
\]

\[
\varepsilon = \frac{-\ln(1+i)}{i}
\] \hspace{1cm} (2.1.8)

and \( \alpha_m = \frac{i}{(1+i)^{N_m-1}} \) \hspace{1cm} (2.1.9)

\( N_m \) is

\[
-\alpha_1 + C_1R\varepsilon\alpha_M(1 + \alpha_M) - C_0N_M\varepsilon\alpha_M(1 - \alpha_M) - C_0\alpha_M = 0
\] \hspace{1cm} (2.1.10)
Hence

\[ R = \frac{\alpha_1}{C_1} \frac{1}{\varepsilon \alpha_M(1 + \alpha_M)} - \frac{C_0}{C_1} \left[ N_M \frac{1}{\varepsilon (1 + \alpha_M)} \right] \] (2.1.11)

\[ R = \frac{\alpha_1}{C_1} \phi - \frac{C_0}{C_1} \mu \] (2.1.12)

\( \phi \) and \( \mu \) are the functions of \( N_m \). This value is independent of the characteristics of the particular mining process. However, \( \frac{\alpha_1}{C_1} \) and \( \frac{C_0}{C_1} \) are dependent upon the characteristics of the mining operation.

### 2.2 Cut-off Grades

To discriminate whether an ore body is ore or waste, a threshold grade should be decided on. This grade is called a cut-off grade [Dagdelen, 1993a] [Dagdelen, 1993b]. Above this threshold, a block is considered as an ore and below this threshold, it is considered as waste. Ore blocks are sent to the mill or processing plant to be crushed, grinded and up-graded of metal content. These blocks are the economical portion of the deposit, whereas the waste blocks are trashed [Asad, 2007].

In order to maximize the net present value, cut-off grade should be optimized as cut-off grade influences the determination of tons of ore and waste. NPV, thus the cut-off grade depends on the factors such as metal price, operating costs, mine capacity, processing, mill and refining stages and the grade
distribution of the deposit [Asad, 2007]. Using these parameters, calculation
of cut-off grade is formulated as the following:

\[ \text{Max NPV} = \sum_{i=1}^{N} \frac{P_i}{(1 + d)^i} \]  \hspace{1cm} (2.2.1)
Subject to:

\[ Qm_i \leq M \text{ for } i = 1, \ldots, N \] (2.2.2)

\[ Qc_i \leq C \text{ for } i = 1, \ldots, N \] (2.2.3)

\[ Qr_i \leq R \text{ for } i = 1, \ldots, N \] (2.2.4)

Where:

\[ P_i = (S_i - r_i) \times Qr_i - c_i \times Qc_i - m_i \times Qm_i - f_i \] (2.2.5)

[Asad, 2007]

\[ P_i \] is the profit function of an ore block. If it is a waste block however, it doesn’t have refining cost and milling cost, but it still has the mining cost as well as the fixed cost. Additionally, there is no selling price, so there is no profit but there is loss from the waste blocks. Thus, it is very important to adjust the cut-off grade as it has a direct influence on maximizing the NPV.

Mining operations have three stages, and each stage has its own optimal cut-off grade. As can be seen in the Figure 2.2, limiting capacities for mining, processing and refining have different peaks. The optimal cut-off grade is at the peak formed by considering all three curves.
Figure 2.2 – Result of different cut-off grades on NPV [Hustrulid et al., 2013].

For the calculation of the cut-off grade, mining cost and the fixed cost can be ignored, because these costs are operative both for ore blocks and waste blocks. Discrimination between them will not change the result. Ignoring these costs, we would be left with:

\[ P_i = (S - r) \times Qr - c \times Qc \quad (2.2.6) \]

Since

\[ Qr = g \times y \times Qc \quad (2.2.7) \]

Where \( g \) is the cut-off grade, by combining Equation 2.2.6 and Equation 2.2.7, we get [Hustrulid et al., 2013]:

17
Another problem is optimizing cut-off grade and estimated capacity at the same time, and it is generally solved by combining possible values of cut-off grade \( g \) and estimated capacity \( M \) within a pre-set range of variability such that the net present value is maximized [Satybadiev and Freidin, 2006].

### 2.2.1 Cut-off Grade for Multiple Destinations

So far, we assumed that a block is either ore or waste, and it goes to processing if it’s ore or to trash if it’s waste. In practice, there might be more than one destination for ore and waste blocks depending on their average grades. The reason for this is the fluctuations in the metal prices. Ore price might increase in time, which would make lower grade material also valuable. In this case, lowering the cut-off grade and keeping higher grade waste blocks in a storage will be more profitable when the metal price increases.

Having multiple destinations is also desired because there are high grade processing plants and low grade processing plants. High grade processing plants have a higher recovery rate, however they are more costly whereas low grade processing plants have lower recovery rate but lower cost. An example of such categories are given in Figure 2.3.
In this case, cut-off for each destination is given as below:

\[
\text{cut-off}_d = \frac{\text{proc\_cost}_d - \text{proc\_cost}_{d-1}}{\text{price} \times (\text{rec}_d - \text{rec}_{d-1})} \quad d = 2, \ldots, D \tag{2.2.9}
\]

Where \( \text{proc\_cost} \) is the mineral processing cost and \( \text{rec}_d \) is recovery at destination \( d \). Each block grade that is between cut-off\(_{d-1} \) and cut-off\(_d \) is sent to destination \( d - 1 \) [Richmond, 2003].

### 2.2.2 Cut-off Grade for Multi-Metal Deposits

There might be more than one precious metal in deposits. In these kind of deposits, the question becomes which metal to consider or how to consider them both. Osanloo and Ataei [2003] propose a method for considering both metals to find the cut-off grade by setting the most precious metal (with high selling price) as the primary metal. The algorithm is as follows:

Step 1 Calculate average grade of each metal in the given intervals of the primary metal.
Step 2 Calculate the equivalent factor \(F_{eq}\) for each metal to primary metal.

Step 3 Add all grades, multiplying by \(F_{eq}\) to find the equivalent total grade.

Step 4 Calculate cut-off according to primary metal, looking at the interval the cut-off grade falls, use interpolation to find where it corresponds in primary metal.

Step 5 Rest of the calculation can be done using only the primary metal

\[ F_{eq} \text{ is defined as:} \]

\[ F_{eq} = \frac{(s_2 - r_2) \times y_2}{(s_1 - r_1) \times y_1} \quad (2.2.10) \]

Where \(s\) is price and \(r\) is refining cost. Subscript 1 corresponds to the primary metal and subscript 2 corresponds to each other metal in the deposit.

### 2.3 Block Sequencing

Block sequencing refers to timing order of extraction of blocks. In practice, the focus is to determine extraction year of the blocks rather than block-by-block extraction order. The problem is formulated as the maximization of NPV of the mining venture. Since a discount rate is applied to each extraction period for NPV maximization, as block extraction period extends the value of block decreases. To avoid this, the optimization process forces valuable blocks to extract at earlier periods. However, due to the slope, or access, constraint, the block accessibility is restricted. In other words, to extract a
block, overlying blocks should be extracted previously. The number of blocks depends on slope angles. There are two general combinations: 1:5 and 1:9 block patterns (Figure 2.4). To access to one block in 1:5 pattern, five over-lying blocks, one-up and one-over, must be removed previously. Similarly, to access to one block in 1:9 pattern, nine over-lying blocks must be removed previously. The combination of these patterns can be also considered.

Figure 2.4 – Block combinations in which five and nine overlying block must be extracted to mine block previously, respectively [Khalokakaie, 1999].

The techniques to solve extraction sequencing problems can be classified into three groups:

2.3.1 Heuristics

Heuristics developed 60s and 70s when computer and operations research technologies were not adequate to address the problem. These approaches do not guarantee the optimality. However, sub-optimal plans can be effectively generated. The drawbacks associated with these approaches were
well-documented.

**Moving (Floating) cones**

This is the simplest heuristic algorithm for determining the optimal pit developed by Carlson et al. [1966]. Starting from the topmost level, for each positive valued block, a cone is constructed that includes the preceding blocks. If the total value of the cone is positive, it is decided that this cone will be mined. This steps are repeated until no positive cones remain.

Although very simple and faster than any other method [Khalokakaie, 1999], this method does not yield an optimum pit. Depending on the search order of cones, different solutions may come out.

**Korobov algorithms**

Korobov algorithm [David et al., 1974] is a cone-based approach. Corrected form of the algorithm [Dowd and Onur, 1992] overcomes the problem of overlapping cones. It is an algorithm that starts from the top and evaluates positive net value blocks at each level by decreasing one level at a time [Ataeepour, 2000]. The algorithm can be briefly described as this [Khalokakaie, 1999]:

**Step 1** Start from the first level, for each block, construct an extraction cone.
Step 2 Allocate positive values in each cone against negative value, until no negative block remains.

Step 3 If value of cone is positive, the cone is in the optimum solution set, remove this cone

Step 4 If any positive block remains, return to Step 1

Adding heuristics to this method [Gershon, 1987a], this algorithm can also be applied in the case of multiple periods by a slight modification as shown in Algorithm 2.1.

Algorithm 2.1 Korobov algorithm for multiple periods

Given number of periods $n$

Current year is $y$

for $i = 0$ to $n$ do

Calculate the NPV of blocks for year $y + i$

Call algorithm Korobov with the calculated blocks

end for

This method will yield $n$ nested pits to be extracted in successive periods. The problem with this method is we are moving away from optimality. Another problem is that we cannot specify a capacity for each period. One way to overcome this latter problem could be increasing the period numbers to get plenty of nested pits. Then, these pits can be aggregated together in such a way to satisfy the capacity constraints.

Ranked positional weight
In other pit design methods, cones are generated upward from each block because first, all preceding blocks have to be mined to reach a given block. This method has a different approach. It draws a downward cone from each block and the block gains a score according to the economical values of the blocks in the downward cone. This approach follows the logic that if a block has a valuable underlying block, it should gain more score as the removal of this block leads the way to the underlying valuable block [Gershon, 1987b].

In Gershon’s method, the value of a block depends on:

i Value of the block itself

ii Value of the cone underlying the block in apex

iii Depth of the block, because of the increase of the transportation costs with depth

An example of scoring of this method can be given in Figure 2.5. The pit is given in Figure 2.5(a) and two sample calculations are given in Figure 2.5(b) and Figure 2.5(c).
After the scoring has been completed, starting from the first level, the highest scored blocks will be mined. This heuristics method is also sub-optimal but it is very intuitive and applicable. It may be combined with another technique for an optimal solution but it is a very powerful starting point.

Figure 2.5 – Demonstration of ranked positional weight technique scoring
2.3.2 Extraction sequencing methods based on the parameterization of an ultimate pit method

These approaches can be seen in two groups: graph theory and maximum flow based approaches. The commercial software used in mining industry is usually based on these approaches because of fast processing time and easy formulation. Notably, graph theory based Lerchs-Grossmann is most frequently used approach in commercial software.

The most important challenge associated with these methods is lack of capacity constraints. Therefore, the results generated can be infeasible. However, there are researches to address this problem. Dagdelen and Johnson [1986] attempted to handle the problem by incorporating the Lagrangean multiplier. The selection of Lagrangean multiplier is a significant problem and the viability of sequence generated depends on this selection. There is no clear way to determine the multiplier such that the NPV of project is maximized.

Graph Theory

The most accepted method based on graph theory is Lerchs-Grossman method, because only this method always yields to the optimal solution. However, this method has a high complexity, increasing the computing time. Also it is difficult to assign different pit slopes and determine mining and processing capacities for each period.
In this approach, first the block model in Figure 2.6(a) is converted to a directed graph in Figure 2.6(b). This directed graph is composed of nodes, which are the circles that represent blocks, and arcs, the arrows from a created root node $x_0$ that is not a block, points to all nodes in the graph. Each node has a weight associated with it, which is the parameter to maximize such as net economic value. This value might be positive or negative [Hustrulid et al., 2013].

![Block model](image1)

![Directed graph](image2)

Figure 2.6 – A labeled block model transformed into a directed graph [Hustrulid et al., 2013]
Each arc is labeled “plus” if the arc is going pointing away from the root and labeled “minus” if pointing to the root. Then each arc is labeled again according to the Table 2.1:

Table 2.1 – Labeling the directed graph

<table>
<thead>
<tr>
<th>Case</th>
<th>Direction</th>
<th>Cumulative Weight</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Plus</td>
<td>Positive</td>
<td>Strong</td>
</tr>
<tr>
<td>2</td>
<td>Plus</td>
<td>Null or negative</td>
<td>Weak</td>
</tr>
<tr>
<td>3</td>
<td>Minus</td>
<td>Positive</td>
<td>Weak</td>
</tr>
<tr>
<td>4</td>
<td>Minus</td>
<td>Null or negative</td>
<td>Strong</td>
</tr>
</tbody>
</table>

For example, assigned block values in Figure 2.7(a) becomes as in Figure 2.7(b) after labeled to above description.

Step 1 Select a strong branch (the part of tree that doesn’t contain the root when a tree is divided in two). Draw an arc to each of the weak nodes that must be removed to mine that strong branch. Relabel the graph. Remove the strong-labeled arc that points to the strong branch.

Step 2 Normalize the tree. All strong arcs should only be branched from the root. If there is a strong minus arc \((X_K - X_L)\), remove this arc and create a dummy arc \((X_0 - X_L)\) to prevent unnecessary support to a strong branch. If there is a strong plus arc \((X_K - X_L)\), remove this arc and create a dummy arc \((X_0 - X_L)\) to prevent supporting a strong branch with a weak branch.
Figure 2.7 – Labeling [Hustrulid et al., 2013]
Step 3 If there is an overlying node connected to the root with a strong arc, 
go to step 1

Step 4 The resulting branches that are connected to the root by a strong arc will be extracted.

The main problem with this method is the generation of nested pits for consecutive periods. To solve this problem, it is possible to use parametrization, which is similar to Lagrangean parametrization, therefore it bears the same obstacles as the Lagrangean. In parametrization, a parameter used in the calculation of economic block model (usually commodity price) is slightly reduced iteratively to generate nested pits. However, because of the parametrization is done heuristically, the solution is no longer optimal.

Maximum flow algorithm-based approaches

Maximum flow problem had existed in the fields computer science and operational research. Johnson [1968] applied this technique to solve the ultimate pit outline problem first in 1968. He formulated the pit to the problem by setting each block as nodes. In addition, a source node and a terminal node is added to the graph. All positive valued block nodes are connected to the source node by arcs coming from the source. Negative valued block nodes are connected to their overlying positive block nodes by arcs from the positive nodes. All the negative valued block nodes are also connected to the terminal
by arcs from the negative nodes. Then, this becomes a maximum flow problem. This problem can be solved by the most common technique developed by Ford and Fulkerson [1956] or by pseudoflow algorithm [Hochbaum, 2008].

2.3.3 Exact methods

Exact optimization methods guarantee finding an optimal solution. The exact optimization methods will be selected if an optimization problem grows polynomially with the problem size. These problem are called class P problems. If problem size exponentially increases this is called an NP complete problem. For even medium size data, the problem cannot be solved via exact methods because exact optimization methods require exponential endeavor [Rothlauf, 2011].

Given economies of scale and decline of high grade mineral deposits, mine sizes have swiftly increased last 20 years. A mine scheduler may have to deal with millions of blocks. This make the use of exact methods almost impossible. However, the innovations in exact methods should be closely monitored.

For pre-defined cut-off grades, the mine production planning problem is formulated as follows [Kumral, 2012]:

$$\text{Max } f(x) = \sum_{i=1}^{T} \sum_{j=1}^{N} V_{ij}(m)x_{ij} \quad (2.3.1)$$
\[ m = \begin{cases} 1 & \text{if the grade of block } j \geq \text{cut-off}_i \\ 0 & \text{otherwise} \end{cases} \]

\[ V_{ij}(1) = \left( \frac{[\text{price} \times \text{recovery} \times \text{ore tonnage of block } j \times \text{grade of block } j] - [(\text{mining cost} + \text{mineral proc. cost}) \times \text{tonnage of block } j]}{(1+n)^{-i}} \right) \times (1+n)^{-i} \]  

(2.3.2)

Mining cost consists of drilling, fragmenting, loading and hauling costs. This cost is generally adjusted by a factor known as mining cost adjustment factor (MCAF) as the extraction depth increases because hauling costs increase. Mineral processing costs are the costs associated with blending, crushing, grinding and concentration. This cost is also adjusted by a factor known as processing cost adjustment factor (PCAF). This is used for varying grade fluctuations.

\[ V_{ij}(0) = - (\text{mining cost} \times \text{tonnage of block } j) \times (1 + n)^{-i} \]  

(2.3.3)

\[ x_{ij} = \begin{cases} 1 & \text{if the block } j \text{ is removed in period } i \\ 0 & \text{otherwise} \end{cases} \]

Subject to:
- **Slope (Access) constraint:** As previously mentioned, to gain access to a block, a number of overlying blocks must be removed depending on the pattern used.

\[
x_{li} \geq x_{ji} \quad i=1,\ldots,T; \quad j=1,\ldots,N \quad \text{and} \quad l\epsilon L_j
\]  

(2.3.4)

- **Mining capacity constraint:** This comprises the maximum amount of material (ore and waste) to be extracted. The mining capacity is determined by available equipment. The selected truck-excavator combinations in terms of the number, capacities and maintenance plans infer mining capacity.

\[
\sum_{j=1}^{N} (r_{ij} + v_{ij}) x_{ij} - C \leq 0 \quad i=1,\ldots,T
\]  

(2.3.5)

- **Mineral processing capacity constraint:** This comprises the maximum amount of ore to be extracted. Since the processing flows are determined previously, ore amount to be extracted must not exceed the installed capacities. If there is an inventory (stockpile) option, sum of mineral processing capacity and stockpile capacity cannot be exceeded for the period under consideration. However, stockpile option is important consideration for short term planning.

\[
\sum_{j=1}^{N} (r_{ij} - A_p) \leq 0 \quad i=1,\ldots,T \quad \text{for all} \quad p
\]  

(2.3.6)
- **Process control constraint**: This constraint is also called as the blending constraint. To consider mineral processing quality requirements, the quality of ore to be produced must be between lower and upper grade limits. Exact optimization methods may not find a solution if the range between lower and upper limits narrows. The relationship between the heterogeneity of ore quality within mineral deposit and the quality requirements of mineral processing plant should be understood well. If grade specifications do not regard the quality heterogeneity of mineral deposit, the processing plant may encounter unexpected costs in the operation.

\[
\sum_{j=1}^{N} c_j x_{ij} r_j \leq H_u \sum_{j=1}^{N} x_{ij} r_{j} \quad i = 1, \ldots, T \quad (2.3.7)
\]

- **Block conservation constraint:**

\[
\sum_{i=1}^{T} x_{ij} \leq 1 \quad j = 1, \ldots, N \quad (2.3.8)
\]

- **Binary constraint:**

\[
x_{ij} \in \{0, 1\} \quad i = 1, \ldots, T \quad j = 1, \ldots, N \quad (2.3.9)
\]

where \( N \) is the number of blocks considered for planning, \( T \) is the number of periods considered for planning, \( V_{ij}(m) \) is the present value of block \( j \) in period \( i \) for material identity \( m \), \( n \) is the discount rate, \( r_j \) is the ore
amount in block \( j \), \( A \) is the mineral processing capacity, \( v_j \) is the waste amount in block \( j \) and \( C \) is the mining capacity, \( L_j \) the set of blocks that must be extracted prior to the mining of block \( j \), \( H_u \) and \( H_l \) are the lower and upper limits on quality and \( c_j \) is the quality of block \( j \).

Kumral [2011] discussed the rationale behind NPV maximization as an objective function. He remarked that a mineral processing operation is installed according to ore consistency in the sense of certain specifications of ore material (e.g. grade, impurities and grindability). If NPV maximization is selected as the objective function, ore extraction will be ranked in descending order of quality since high quality blocks are pushed to be extracted at earlier periods in optimization. Quality ranking in mining is due to NPV maximization causing extra cost in mineral processing stage.

Given that mineral processing cost is generally higher than mining costs, this problem can be highly significant. In other words, the benefit obtained by the mining stage could be lost exceedingly in the mineral processing stage. He concluded that if the operation has mining operation only, NPV maximization can be a rational objective function. However, an operation have mining and mineral processing stages, NPV maximization as an objective function is arguable. To deal with this problem, Kumral [2011] proposed maximin as an objective function to extract more uniform metal quantity among periods such that mineral processing requirements are sat-
sified. Some researchers have used blending constraint that reduces quality fluctuations among periods. As the range of upper and lower limits is narrowed, the output of formulation of NPV maximization will approach to the output of maximin formulation [Kumral, 2011]. The model formulated as NPV maximization will be time consuming in comparison to maximin formulation.

Previous optimization model can be extended to simultaneous optimization of extraction sequencing and ore-waste discrimination. In this model, the decision variable is not only extraction period but also block destinations. In other words, ore–waste discrimination is managed by the optimization process. The model can be given at [Kumral, 2012]:

$$\text{Max } f(x) = \sum_{i=1}^{T} \sum_{j=1}^{N} \sum_{d=1}^{D} V_{ijd} \times x_{ijd}$$  \hspace{1cm} (2.3.10)

- Access constraint:

$$\sum_{d=1}^{D} x_{ild} \geq \sum_{d=1}^{D} x_{ijd} \quad i=1, \ldots, T; \quad j=1, \ldots, N \text{ and } k \epsilon L_j$$ \hspace{1cm} (2.3.11)

- **Destination capacity constraint:** In previous model, we assumed that there were two destinations, namely waste and processing destinations. In this model, these destinations were determined by applying a cut-off grade. Since a cut-off grade was not used, ore-waste discrimination can-
not be done earlier. Furthermore, there would be many destinations. For example, waste, sub-grade material, marginal material, low grade processing and high grade processing are possible destinations. The constraint is formulated as follows:

\[
\sum_{j=1}^{N} f_j x_{ijd} - Upp_d \leq 0 \quad i=1, \ldots, T \text{ and } d = 2, \ldots, D \quad (2.3.12)
\]

\[
\sum_{j=1}^{N} f_j x_{ijd} - Low_d \geq 0 \quad i=1, \ldots, T \text{ and } d = 2, \ldots, D \quad (2.3.13)
\]

**- Process control constraint:**

\[
\sum_{j=1}^{N} c_j f_j x_{ijd} \leq H_{du} \sum_{j=1}^{N} f_j x_{ijd} \quad i=1, \ldots, T \text{ and } d = 2, \ldots, D \quad (2.3.14)
\]

\[
\sum_{j=1}^{N} c_j f_j x_{ijd} \geq H_{dl} \sum_{j=1}^{N} f_j x_{ijd} \quad i=1, \ldots, T \text{ and } d = 2, \ldots, D \quad (2.3.15)
\]

**- Block conservation constraint:**

\[
\sum_{i=1}^{T} \sum_{d=1}^{D} x_{ijd} \leq 1 \quad j=1, \ldots, N \quad (2.3.16)
\]

**- Binary constraint:**

\[
x_{ijd} \in \{0, 1\} \quad i=1, \ldots, T, \; j=1, \ldots, N \text{ and } d = 1, \ldots, D \quad (2.3.17)
\]
Where $D$ is the number of destinations that should be at least two (waste dump and one processing destination), $V_{ijd}$ is net present value of block $j$ in period $i$ in destination $d$, $f_j$ is tonnage of block $j$ and $x_{ijd}$ is a binary variable (if block $j$ is sent to destination $d$ in period $i$, the variable is 1; otherwise, it is zero). It is thought that mine production scheduling ensures more efficient use of resources.

### 2.3.4 Dynamic programming

Koenigsberg [1982a], Omur and Dowd [1993], Tolwinski and Underwood [1996], Wang et al. [2013], Roman [1973] and Dowd and Elvan [1987] used dynamic programming for various elements of mine production scheduling. Dynamic programming is a strong method for sequential decision making problem. In this approach, the block sequencing problem is divided into simpler subproblems problems (is called stages). Each stage is optimized. Bellman [1956] developed method. Bellman’s Principle of Optimality prescribes, at a certain time, the value of a decision problem depends on the payoff of initial choices and the value of the remaining decision problem. Koenigsberg [1982b] used dynamic programming for three-dimensional case. Later, Wright and Weiss [1989] modified Koeninsberg algorithm.

The algorithm is the following [Khalokakaie, 1999]:

Step 1: Add a row to the top of the mine. Set the values of these blocks to zero.
Step 2: For each column, starting from the top, change the block value to cumulative total of all block values above, that is:

\[ M_{ij} = \sum_{k=1}^{i} m_{kj} \]  

(2.3.18)

Where

\( m_{ij} \) is the block value at row \( k \) and column \( j \)

Step 3: Start from the leftmost column. For each block in the column, calculate the overall cumulative value \( P_{ij} \) which is the sum of \( M_{ij} \) and the highest value among:

a - block on the upper left
b - block on the left
c - block on the lower left

\[ P_{ij} = M_{ij} + \max \{ P_{i-1,j-1}, P_{i,j-1}, P_{i+1,j-1} \} \]  

(2.3.19)

Draw an arrow from the highest value block to the original block.
Replace the block value with \( P_{ij} \). Repeat for all blocks.

Step 4: Take the block with maximum positive value in the first row, follow the arrows from and to the left of the block. This forms the optimal pit. If there is no positive block in the first row, this is not a profitable pit.
This method assumes that the pit is rectangular and adds a row above it. In fact the pit might be irregular or on a slope. In these cases, the solution might not be optimal. Thus, this method does not seem ideal for mining conditions.

2.3.5 Metaheuristics

Simulated annealing

Simulated annealing was developed initially by Kirkpatrick et al. [1983] and Černý [1985], named based on an analogy in condensed matter physics, simulation of the annealing of solids and large combinatorial optimization problems. Simulation of the annealing of solids procedure involves heating a solid in a heat bath to a maximum value, where the solid transforms into a liquid phase. Then, the bath cooled sufficiently slowly so that particles arrange themselves in the low energy ground state [Van Laarhoven and Aarts, 1987]. It is a combination of hill climbing and random walk methods [Russell et al., 1995]. Application of simulated annealing method to the open pit mine production scheduling is accomplished by Kumral and Dowd [2005] by the following steps:

Step 1: Start with a non-optimal feasible solution.

Step 2: Select a portion of the blocks.

Step 3: **Possibility 1:** Modify ore-waste discrimination. Change ore blocks
to waste, and waste to ore by some probability .

**Possibility 2:** Modify period of the given block to a previous or following period by some probability.

Step 4: Recalculate NPV for the newly found solution.

Step 5: Apply *Metropolis* criterion as the acceptance criterion.

Step 6: If NPV has not increased for last $n$ steps, terminate. Else, go to Step 2.

Metropolis criterion [Metropolis et al., 2004] is a criterion that takes two solutions and temperature $T$ as inputs:

$$
\min \left( 1, \exp \left\{ -\frac{(E - E_0)}{T} \right\} \right) \tag{2.3.20}
$$

$T$ should be chosen high at first and then decreased slowly. If $T$ decreases slowly enough, global minimum will be reached [Abecasis, 2006]. According to how the Metropolis criterion is set up, at higher temperatures, the criterion tends to accept the solutions that are not improving as well as those are improving. This stage is called exploration of the parameter space as shown in Figure 2.8(a).
At $T$ is lowered, there is less chance of accepting solutions that are not improving. If $T$ is not decreased slowly enough, there is a chance of getting stuck at a local minimum as shown in Figure 2.8(b).

**Genetic algorithms**

Genetic algorithms are introduced by John Holland [Khalokakaie, 1999]. These algorithms are based on the principals of natural selection. It involves combining structured but partially randomized different solutions to create new solutions, then applying the natural selection rule on the set of solutions. The steps of this algorithm are as follows [Clement and Vagenas, 1994]:

**Step 1:** Start with a set of feasible solutions.

**Step 2:** Reproduction: Each solution has a fitness value $f_i$. Select a random
solution from the distribution on fitness values.

Step 3: Crossover: Take two of the solutions. Exchange genetic material at a randomly selected point between these solutions with a probability $p_c$ to create two child strings. If cross-over does not take place, parent genes are transferred directly to the child genes with no modification. If it does take place, both child strings will receive some of the genetic material from both parents.

Step 4: Mutation: Allow random flip of binary positions in the string with a probability of $p_m$.

Step 5: Calculate fitness values, go back to Step 2

Application of this technique to the optimal pit problem is accomplished by Denby and Schofield [1995] and Denby and Schofield [1994] by coding random pits as genes. Fitness function included maximizing NPV, minimizing early stripping and for multiple mineral case, balancing ore production. Population size of 20-50 pits was used. In the crossover step, 40%-60% the schedules were crossed over. 1%-5% mutation probability applied. After these steps, it was made sure the extraction constraints were not violated. Then the fitness of each pit was improved by local optimization of pits [Os-anloo et al., 2008].
Ant colonies

The Ant Colony Optimization (ACO) was proposed to solve the mine production scheduling problem [Sattarvand, 2009] [Sattarvand and Niemann-Delius, 2013] [Shishvan and Sattarvand, 2015]. ACO was developed on the imitation of the foraging mechanism of ants. To simulate the ACO process for mine production scheduling problem, pheromone trails were used for the blocks as programming variables, which are equal to the number of production periods. Similar to other meta-heuristics, the approach was combined with a traditional method in sense that Lerchs-Grossmann was used to generate an initial image. Mine schedules were generated through ACO iterations based on pheromone trails. The author proposed various ways to implement ACO optimization iterations. To see the performance of the algorithm, a computer program has been written and a case study was implemented for a small data set.
3 Approach to Solve Open Pit Production Scheduling Problem

A computer software was written to plan open pit mines. In this section, a detailed overview to the software will be given. A summary is given in Algorithm 3.1.

**Algorithm 3.1** Mine Production scheduling Using Positional Weight and Simulated Annealing

<table>
<thead>
<tr>
<th>Input:</th>
<th>Grade of each block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine cutOff grade using [Osanloo and Ataei, 2003]</td>
<td></td>
</tr>
<tr>
<td>function RANKED POSITIONAL WEIGHT(blockGrades)</td>
<td></td>
</tr>
<tr>
<td>Call Algorithm 3.2 to produce initial output</td>
<td></td>
</tr>
<tr>
<td>return initialOutput</td>
<td></td>
</tr>
<tr>
<td>end function</td>
<td></td>
</tr>
<tr>
<td>function SIMULATED ANNEALING(blockGrades, initialOutput, cutOff)</td>
<td></td>
</tr>
<tr>
<td>Call Algorithm 3.3 to optimize the initial output</td>
<td></td>
</tr>
<tr>
<td>return finalOutput</td>
<td></td>
</tr>
<tr>
<td>end function</td>
<td></td>
</tr>
</tbody>
</table>

**Output:** finalOutput

Algorithm 3.1 first calculates the equivalent cut-off grade for multiple metal deposit using Osanloo and Ataei [2003]’s method of using equivalent grade factors as described in Section 2.2.2. The algorithm then calls the Ranked Positional Algorithm that is illustrated in Algorithm 3.2. This function returns an initial output that is later passed as a parameter to the Simulated Annealing function. Having received the initial input and the cut-off grades for each destination, Simulated Annealing algorithm shown in Algorithm 3.3 produces the final mine plan.
3.1 Ranked Positional Weight Algorithm

This algorithm is used to produce a sub-optimal feasible solution for the mine deposit in question. In this section, all referrals are to the line numbers of Algorithm 3.2.

3.1.1 Finding Block Economic Values

First, blockGrades are read from the data. For each block, the economic value is calculated in the function at line 1 using the equation 3.1.1 where \( i, j \) and \( k \) are the coordinates of the block. Then, the net present value is calculated according to Equation 3.1.2. This value is saved for each block.

\[
value_{ijk} = \begin{cases} 
  [\text{price} \times \text{recovery} \times \text{ore tonnage}_{ijk} \times \text{blockGrade}_{ijk}] \\
  - [(\text{mining cost} + \text{mineral proc. cost}) \times \text{tonnage}_{ijk}] & \text{blockGrade}_{ijk} \geq \text{cutOff} \\
  - (\text{mining cost} + \text{mineral proc. cost}) \times \text{tonnage}_{ijk} & \text{blockGrade}_{ijk} < \text{cutOff}
\end{cases}
\] (3.1.1)

\[
value_{ijk} \leftarrow value_{ijk} \div (1 + \text{discountFactor})^{\text{period}}
\] (3.1.2)

3.1.2 Calculating Ranked Positional Weight

In the next function at line 9, positional weight score of all blocks are calculated. Instead of the traditional upwards cone, a downwards cone is drawn for each block. The angles of the cone are the opposite angles of the
Algorithm 3.2 Ranked Positional Weight Algorithm

Input: blockGrades, noOfPeriods, miningCapacity, cutOffs

1: function FIND BLOCK ECONOMIC VALUES(blockGrades, cutOffs)
2:     for all blocks do
3:         for all metals do
4:             Calculate blockEconomicValue using Equation 3.1.1
5:         end for
6:     end for
7:     return blockEconomicValue
8: end function

9: function CALCULATE POSITIONAL WEIGHT(blockGrades)
10:     for all blocks do
11:         Draw a cone with the blocks angles
12:         Put all the blocks in the cone in the array belowBlocks
13:         for all belowBlocks do
14:             blockScore ← blockScore + k × belowBlock.economicValue ÷ level
15:         end for
16:     end for
17:     return blockScores
18: end function

19: function PERIOD CLASSIFICATION(blockGrades, blockScores)
20:     Form array: reacheableBlocks
21:     period p ← 0
22:     while reacheableBlocks.size() > 0 and p < noOfPeriods do
23:         Mine the block with highest blockScore at period p
24:         if numberOfBlocks(p) > miningCapacity then
25:             p ← p + 1
26:         end if
27:     end while
28:     Reform array: reacheableBlocks
29: end function

Output: initialOutput
preceding block angles. In other words, we are including in the cone the blocks that will have one less preceding block after this block is removed. Because in theory, the current block is as valuable as its own value, plus removal of this block will lead the way of removing the underlying blocks, thus also a part of their value. This effect is less in each level that gets further from the current block because of the increasing number of preceding blocks, resulting in the assignment on line 14, \( k \) being the chosen constant to apply the magnitude of the effect.

### 3.1.3 Period Classification

In this last step, iteratively highest scored reachable block is removed and put in the current period. When the number of blocks in the period reach the mining capacity, mining is continued from the following period. If number of maximum periods is reached or if there are no more blocks to extract, program stops. The output file consists of which period each block location corresponds to. Reachable blocks list are kept as an array and to speed up the program, when a block is extracted, instead of a complete search for reachable blocks, only the blocks under the extracted block are searched and added to the list.

In Positional Weight portion of the program, destination classification is not performed because the algorithm is not suited to this purpose. In the beginning of Simulated Annealing portion, it is assumed that each block
belongs to the highest priority destination that’s cut-off grade is less than the block’s grade. If the block’s grade is less than the cut-off of any destination, then the block is considered as waste.

In general, finding block economic values function is numberOfBlocks × metals and period classification is numberOfBlocks in complexity. Most time consuming part of the algorithm is calculating positional weights, which in worst case has numberOfBlocks\(^2\) computation time complexity.

### 3.2 Simulated Annealing Applied to Mine Production scheduling

This algorithm takes the initial solution by Ranked Positional Weight algorithm and optimizes the solution in terms of assigned period and destinations. In this section, all referrals are to the line numbers of Algorithm 3.3.

#### 3.2.1 Calculating Mine Economic Value

The algorithm starts out by calculating the mine value at line 1 using the solution generated by the Ranked Positional Weight algorithm. Mine Value is calculated by adding up all the block economic values that are defined at Equation 3.1.1.
Algorithm 3.3 Simulated Annealing Algorithm Applied to Mine Production scheduling

**Input:** blockGrades, initialOutput

1: function Calculate Mine Value(blocks)
2:     for all blocks do
3:         mineValue ← mineValue + blockEconomicValue
4:     end for
5:     return mineValue
6: end function

7: function Classify Transition Destination Blocks(blockGrades)
8:     for all blocks do
9:         for all cutOffs do
10:             if blockGrade > cutOff(i) × cutOffRange then
11:                 Put this block in transitionDestBlocks array
12:             end if
13:         end for
14:     end for
15:     return transitionDestBlocks
16: end function

17: function Classify Transition Period Blocks(blocks)
18:     for all blocks do
19:         if precedingBlocks.period < block.period then
20:             Put this block in transitionPeriodBlocks array for earlier period
21:         end if
22:         if succeedingBlocks.period > block.period then
23:             Put this block in transitionPeriodBlocks array for later period
24:         end if
25:     end for
26:     return transitionPeriodBlocks
27: end function
Simulated Annealing Algorithm Applied to Mine Production scheduling (continued)

28: function **SIMULATED ANNEALING FOR MINE PRODUCTION SCHEDULING**(blocks)

29: while solution is still improving do

30: if number of solutions at this $T > maxNoOfSolutions$ then

31: $T \leftarrow T \times TFactor$

32: end if

33: Randomly select some blocks either from transitionDestBlocks or transitionPeriodBlocks

34: Replace these blocks with the values in blocks array to create a newBlocks array

35: if solution not feasible then

36: goto line 33

37: end if

38: newMineValue $\leftarrow$ **CALCULATE MINE VALUE**(newBlocks)

39: newObjFunction $\leftarrow k_1 \times$ newMineValue $- k_2 \times$ (processCapacity $-$ noOfBlocks)

40: if **METROPOLIS**(objectiveFunction, newObjFunction, $T$) then

41: blocks $\leftarrow$ newBlocks

42: mineValue $\leftarrow$ newMineValue

43: objectiveFunction $\leftarrow$ newObjFunction

44: if mineValue $>$ bestMineValue then

45: bestMineValue $\leftarrow$ mineValue

46: end if

47: **CLASSIFY TRANSITION PERIOD BLOCKS**(blocks)

48: end if

49: end while

50: return bestMineValue

51: end function

**Output:** FinalOutput
3.2.2 Classification of Transition Destination Blocks

Cut-off grades are favorable guides for assigning blocks to process destinations or waste dump. However, there are some troubles with using cut-off grades. Firstly, as mentioned before, costs are approximated and some costs might be neglected, therefore they are not optimal. Secondly, when process capacities are taken into account, if the algorithm binds strictly to the cut-off grades, because of the capacity constraints, reaching the high grade areas located deeper in the mine will be more difficult.

To solve this problem, we provide a solution that permits elasticity of the cut-off grades. An adjustable value of cut-off range is defined to allow flexibility and the blocks that fall to the flexible area are marked as transition blocks. Whether they belong to the upper or lower process grade are measured by the effect on the objective function.

In the program, this is managed as follows. For one time only, transition destination blocks should be selected for the annealing process. This is accomplished by the function at line 7. This function evaluates each block to observe if they are within a predefined range of the cut-off grades. If this evaluation yields positive, the block in question is marked as a transition block and added to the transition block array.
3.2.3 Classification of Transition Period Blocks

There is also a capacity restraint on each period. One of the crucial parameters that influences the size of the pit and the mine economic value is period classification. Thus, period classification is an important decision variable.

In the program, likewise to the transition destination classification, transition blocks for the periods are determined at line 17. In the function, first every block is checked for their preceding blocks. If the block’s all preceding blocks are less than the period of the current block, it means the block could have belonged to the previous period without violating any access constraints. In like manner, a similar procedure is repeated for the succeeding blocks where in this case, if all the succeeding block extraction time periods are later than the current block, the block can as well be mined in the next period. These possibilities are added to the transitionPeriodBlocks array.

Unlike classification of transition destination blocks, instead of having called just in the beginning of the program, it is called after each time there is a change in any of the block periods because a change in a particular block might change the transition status in the preceding and succeeding blocks.

3.2.4 Simulated Annealing for Mine Production scheduling

This function at line 28 is where the improvement is performed. The function runs a pre-set number of times or as long as there is no more im-
provement is taking place. Each iteration produces one solution and if the solution is not feasible, it is not evaluated. If it is feasible, it is accepted based on the Metropolis criterion.

The function starts out by selecting blocks randomly from transition destination blocks or transition period blocks, performing the changes proposed in functions at lines 7 and 17. A newBlocks array is constructed with the inclusion of these changes and unchanged blocks. If this new solution is not feasible, such as it exceeds the period capacity or violating some other constraint, the solution is not evaluated, the program returns to line 33 to propose another solution instead.

After newBlocks solution is introduced, the success of the solution is evaluated through first calculating the newMineValue yielded with this solution. The evaluation is done by the objective function at line 39 which has two component. First component is the yielded mine economic value of the new solution, second component is number of blocks that are short-coming of the process capacity. The objective is to maximize the mine economic value and at the same time to penalize under capacity production. Coefficient of the first component is set to 1 and the second component is set by the user depending on how much emphasis on the production quantity the user desires.
The magnitude of the objective function is then compared to the preceding solution’s objective function. The evaluation is accomplished by the Metropolis criterion given in Equation 2.3.20. Basically, this evaluation consists of accepting the improving solution and if the solution is not improving, accepting it with a probability that depends on how far off the new solution is and the temperature parameter $T$ that decreases in time.

If the Metropolis criterion accepts the new solution, $blocks$, mine value and the objective function are updated. Also, if this mine value is the highest valued mine value of all solutions, then this is stored in the memory to be returned at the end of the program. As a result of the changes in $blocks$, transition period blocks might have altered. To revise the transitionPeriodBlocks array, Classify Transition Period Blocks function at line 17 is called.

If, counting this one, number of solutions exceeded a pre-set value, $T$ value is decreased. In the end, after the termination of the while loop, best mine economic value is returned. As the final output, the $blocks$ configuration for this particular economic value is written to the file with the corresponding extraction periods and process destinations. The mine will be excavated according to this plan.
3.3 Program Highlights

3.3.1 Annealing Parameters

**Cooling Schedule**  Initial temperature is set by taking the first two solutions and finding the $T$ in Equation 2.3.20 such that the equation will be equal to 0.5. In other words, after the first two solutions, the third solution will have a 50% chance to be accepted even if it is not improving. Decrementing $T$ is accomplished by $T ← T \times 0.9999$ to ensure it decreases slowly enough to accept more solutions. Each time a number of solutions are found ($10 \times \text{noOfPeriods}$), $T$ is updated as described.

**Stopping Criteria**  There are two conditions that can stop the annealing loop. First condition: when Metropolis Criterion does not accept the solution for a pre-set amount of iterations (in our case, noOfPeriods). Second condition: the program loops for a user set amount of value. The second one exists to produce solution under limited time. However, the longer the program is let to run, the better results will be obtained.

**Maximum Number of Solutions at Each Temperature**  This is a parameter that sets the number of generated solutions before decreasing the temperature. This should depend on the size of the data so in our program, we set it to $50 \times \text{noOfPeriods}$.
3.3.2 Mine Economic Value Calculation

For each block, the calculation is done differently depending on the period and process destination, rock type, processing cost adjustment factor and mining cost adjustment factor of the block.

Multiple Metals In poly-metal mineral deposits, blockGrade is multiple; one for each metal. Hence, Equation 3.1.1 is applied to each metal shown in Equation 3.3.1. Of course, then the discount rate is applied to find the NPV as in Equation 3.1.2.

$$\text{value} = \sum_{i=1}^{\text{metals}} \text{blockValue}[\text{metal}] \quad (3.3.1)$$

As the discount rate is multiplied by period to imitate the value loss of money in time, extraction period influences the block, thus the mine economic value. In general, as the process capacity permits, it is more beneficial to mine the blocks earlier due to this loss.

Multiple Process Destinations The program is able to work with more than one process destinations. This is practically very useful as lower grade blocks might not cover the processing costs. Multiple cut-off grades are calculated according to the mineral processing cost, mining cost, recovery and specific gravity (thus the ore tonnage) and each grade is sent to the corresponding process.
**Cut-off Range**  SA uses the guidance of the cut-off grades. However, it does not bind to them strictly. During the generation of transition destinations process, the cut-off range is used to decide to which extent the blocks out of the limits of the cut-off grade could be accepted. This parameter may have a lot of effect on the results. If set high enough, it can remove cut-off grade boundaries altogether.

**Rock Types**  The program is able to work with more than one rock type and thus has to define processing costs and recoveries for each rock type. This option makes the program more practical as the costs to process different rocks vary.

**Processing Cost Adjustment Factor**  Ore mining does not have the same cost as the waste mining. Processing Cost Adjustment Factor (PCAF) is used to reflect this difference. In our program, mining cost for each destination are entered by the user. Mining cost is decided depending on the process destination.

**Mining Cost Adjustment Factor**  Modifiable mining cost adjustment factor (MCAF) is used to reflect increased cost of transport in deeper levels of the deposit. MCAF is entered by the user and the MCAF is added to the mining cost using the formula in Equation 3.3.2.

\[
\text{miningCost} \leftarrow \text{miningCost} + \text{level} \times \text{MCAF} \quad (3.3.2)
\]
3.3.3 Handling Topography

**Slope Angles** There are two approaches to define slope angles. First approach, called variable slope angles, shown by the red line in Figure 3.1(a), in which the angles are considered to be pit-wise. For four horizontal directions (north, south, west, east), different angles can be defined. Second way of defining slope angles is called multiple variable slope angles which is shown by the red line in Figure 3.1(b). In this model, each individual block has its own defined angles to preceding blocks, mostly due to its rock type [Shishvan et al., 2014].

In our program, slope angles are defined as the second method mentioned above. However, the user can easily convert variable slope angles to multiple variable slope angles by taking the deepest level of the pit and adjusting the block angles at this level equal to the pit-wise angles. For each level of the pit, this angles will change slightly depending on the level the block is located. It can be found by taking the tangent of the depth level and the distance from the furthest point of the pit in each direction.

**Pit Shape** The program needs to be inputted a cubic model. Since pits can present in any shape, the shape is completed to a cubic model by assigning air (empty) blocks to the empty spaces. In the program, when access and capacity constraints are checked, as well as the mine economic value is calculated, these air blocks are skipped.
(a) Variable Slope Angles

(b) Multiple Variable Slope Angles

Figure 3.1 – Two types of slope angles
**Transportation Penalty**  There is a module in the program with the option to include a transportation penalty or not. This may be needed in some cases where the nested period pits are not very smooth or disjoint and not cost efficient in terms of transportation. In the program, this is managed though looking at the neighbours of each block and if their periods do not match, adding a penalty term. The total of these penalties are incorporated into the objective function as a negative value.
4 Case Studies

To demonstrate the performance of the approach, two case studies have been conducted. Figures of the cross sections given in each case study is generated using the Stanford Geostatistical Modeling Software (SGeMS) [Remy et al., 2009]. For each study, corresponding parameter file is given. This parameter file gives information about the problem size, process destinations, costs, prices and some parameters that are used within the program. All except the program parameters are self explanatory. Explanations of these program parameters are as follows:

**Cut-off range**  As previously described in detail, in SA part of the program, cut-offs are not strictly followed. Within a range they are rather treated as if they can belong to the previous or the latter destination. In this parameter, this range is defined. Best decision might be taken by taking the histogram of the data.

**Number of iterations**  SA terminates either when there is no improvement or when the given number of iterations are reached. Mining problem sizes are very large and thus usually number of iterations are reached sooner than settling on the ideal solution. This parameter should be selected as large as possible as time permits.
Short-coming process blocks effect  This parameter specifies how important it is to fulfill the process capacities. This value ranges between 0 where it is not considered and 1, where this criterion is all that matters.

4.1 Case Study 1

This is a large case study generated from a public drill hole dataset in http://www.kriging.com/datasets/ [2014]. The dataset contains samples from two metals (copper and molybdenum). Using sequential Gaussian simulation, a 3D block model of 595,046 blocks was created where each block is 5x5x5 m in size. The mining company is imagined to have one waste dump and three process destinations (low, middle and high grade processing), where the mineral is processed by different procedures thus their costs and recoveries are different. The slopes are 45 degrees in four directions (north, south, east, west). Parameters that are used are given in Table 4.1. With 595,046 blocks, 4 periods and 4 total destinations there are 595,046 × 4 × (4+1) = 11,900,920 decision variables. In the calculation, destinations are incremented by one because the block can also be decided not to be extracted.

Cut-off grades were calculated to be 0.4859 0.6006 0.7257 % respectively for each process. First, positional weight algorithm was run to output a sub-optimal initial result. Then, this result is inputted to the SA as an initial solution. This case is run twice, once without including the transportation penalty effect and one including. The results produced by each algorithm
### Parameters

<table>
<thead>
<tr>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 5 5</td>
<td>block dimensions (m)</td>
</tr>
<tr>
<td>98 132 46</td>
<td>number of blocks in x,y and z directions</td>
</tr>
<tr>
<td>4</td>
<td>number of total destinations</td>
</tr>
<tr>
<td>3</td>
<td>number of ore destinations</td>
</tr>
<tr>
<td>1</td>
<td>number of waste destinations</td>
</tr>
<tr>
<td>4</td>
<td>number of periods</td>
</tr>
<tr>
<td>2</td>
<td>number of rock types</td>
</tr>
<tr>
<td>3.0 3.0 3.0 3.0</td>
<td>mining cost of each destination rock type 1 ($/tonne)</td>
</tr>
<tr>
<td>4.0 4.0 4.0 4.0</td>
<td>mining cost of each destination rock type 2 ($/tonne)</td>
</tr>
<tr>
<td>0.0 15.0 30.0 50.0</td>
<td>mineral processing cost of each destination rock type 1 ($/tonne)</td>
</tr>
<tr>
<td>0.0 18.0 38.0 53.0</td>
<td>mineral processing cost of each destination rock type 2 ($/tonne)</td>
</tr>
<tr>
<td>0.0 60.0 70.0 80.0</td>
<td>sales cost of each destination ($/tonne concentrate)</td>
</tr>
<tr>
<td>5.0 5.0 5.0 5.0</td>
<td>specific gravity of each destination (tonne/m$^3$)</td>
</tr>
<tr>
<td>60000</td>
<td>mining capacity</td>
</tr>
<tr>
<td>25000 20000 15000</td>
<td>processing capacity of each destination</td>
</tr>
<tr>
<td>0.01</td>
<td>cut-off range</td>
</tr>
<tr>
<td>2</td>
<td>number of metals (Cu and Mo)</td>
</tr>
<tr>
<td>0.01 0.03</td>
<td>ore price ($/gr) (Cu and Mo)</td>
</tr>
<tr>
<td>0.0 0.40 0.70 0.95</td>
<td>recovery for each destination metal 1 rock type 1</td>
</tr>
<tr>
<td>0.0 0.30 0.40 0.75</td>
<td>recovery for each destination metal 2 rock type 1</td>
</tr>
<tr>
<td>0.0 0.40 0.80 0.95</td>
<td>recovery for each destination metal 1 rock type 2</td>
</tr>
<tr>
<td>0.0 0.30 0.40 0.75</td>
<td>recovery for each destination metal 2 rock type 2</td>
</tr>
<tr>
<td>0.1</td>
<td>discount rate</td>
</tr>
<tr>
<td>300</td>
<td>number of iterations</td>
</tr>
<tr>
<td>0.9</td>
<td>short-coming process blocks effect</td>
</tr>
<tr>
<td>0.1</td>
<td>MCAF</td>
</tr>
</tbody>
</table>

Table 4.1 – Parameter File for Case Study 1
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Resultant NPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranked Positional Weight</td>
<td>$1,613,205,645</td>
</tr>
<tr>
<td>SA with Transportation Penalty</td>
<td>$1,687,973,927</td>
</tr>
<tr>
<td>SA without Transportation Penalty</td>
<td>$1,688,949,559</td>
</tr>
</tbody>
</table>

Table 4.2 – Comparison of resultant NPVs

can be seen in Table 4.2.

From ranked positional weight algorithm to SA, there is an improvement of $75,743,954 which is 5%. SA with transportation penalty is slightly lower than SA with transportation penalty because objective function is shared with transportation penalty. Not as much emphasis is given on maximizing NPV, thus in fact, this module of the program needs more time to arrive at the equal NPV as without the transportation penalty.

In Figures 4.1 and 4.2, cross-sections of all algorithms are given. In these figures, there is a color scale on the right that indicates which color corresponds to which period. It can be seen from these figures that compared to ranked positional weight algorithm, SA is inclined to mine the blocks in the earlier periods to increase the NPV. However, SA results look less smooth than ranked positional algorithm’s result. This is mainly because of the structure of annealing, where blocks are switched between the periods one by one, causing the sections to look rugged. Transportation penalty is meant to deal with this; rugged transport is most costly compared to plain routes. Indeed, SA with transportation cost tended to unify the same period blocks
thus overall, more smooth pits are formed.

Detailed output of SA without transportation penalty summary has been provided in Figure 4.4. SA with transportation penalty algorithm’s output is very similar, hence not provided. As can be seen in each process, the average grade is maintained throughout the periods, which is important for mine’s economic stability.
Figure 4.1 – Cross-section (Sections: 48, 52, 0)
Figure 4.2 – Cross-section (Sections: 56, 65, 0)
Figure 4.4 – Average grades and number of blocks in each destination and period
It is clear that process capacity push is working well, except for the high grade process as can be seen in Figure 4.4 (c). The reason for this is the pit is not homogeneous and does not have enough high grade blocks for the later periods. Most of the high grade material is near the surface of the mine thus the capacity at first period is completely filled.
Lastly, comparison of overall average grades of low, medium and high grade processes can be seen in Figure 4.5. As expected, high grade process has the highest average grade, followed by medium and low grades.

4.2 Case Study 2

This case study is a gold mine that has 60,372 blocks and each block is 10x10x10 m in size. In contrast to case study 1, this dataset contains only one grade for each block (gold) that are generated by sequential Gaussian simulation. The mining company is imagined to two process destinations (low and high grade processing) and a waste dump. As expected, recovery
is higher at high grade processing destination, however also the processing cost is higher. Thus, the blocks sent to high grade processing should have the enough grade to cover these costs. The slopes are 38 degrees in four directions (north, south, east, west).

Parameters that are used in this case are given in Table 4.3. With 60,372 blocks, 4 periods and 3 total destinations, there are $60,372 \times 4 \times (3 + 1) = 965,952$ decision variables. As explained before, destinations are incremented by one due to block may chosen to be not extracted.

Cut-off grades were calculated to be 0.4 and 1.7 g/tonne respectively for low and high grade processes. However, notably, cut-off range of this case was set to be 0.4 in Table 4.3, which means the cut-off grades are less strict.

First, the dataset and the parameters was input to positional weight algorithm was run to yield a sub-optimal initial result. Then, this result is inputted to the SA as an initial solution. This case is run twice, once without including the transportation penalty effect and one including. The results produced by each algorithm can be seen in Table 4.4.

As opposed to case study 1, in this study SA with transportation penalty reached a higher NPV in the same amount of iterations. The main reason for this the ranked positional weight algorithm gave an output that is rather scattered as can be seen in Figure 4.6 and Figure 4.7. SA with transportation
### Parameters

<table>
<thead>
<tr>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 10 10</td>
<td>block dimensions (m)</td>
</tr>
<tr>
<td>43 78 18</td>
<td>number of blocks in x,y and z directions</td>
</tr>
<tr>
<td>3</td>
<td>number of total destinations</td>
</tr>
<tr>
<td>2</td>
<td>number of ore destinations</td>
</tr>
<tr>
<td>1</td>
<td>number of waste destinations</td>
</tr>
<tr>
<td>4</td>
<td>number of periods</td>
</tr>
<tr>
<td>1</td>
<td>number of rock types</td>
</tr>
<tr>
<td>8.0 10.0</td>
<td>mining cost of each destination ($/tonne)</td>
</tr>
<tr>
<td>0.0 22.0 35.0</td>
<td>mineral processing cost of each destination ($/tonne)</td>
</tr>
<tr>
<td>0.0 2.0 2.0</td>
<td>sales cost of each destination ($/tonne concentrate)</td>
</tr>
<tr>
<td>3.2 4.0</td>
<td>specific gravity of each destination (tonne/m³)</td>
</tr>
<tr>
<td>3000</td>
<td>mining capacity</td>
</tr>
<tr>
<td>1000 1500</td>
<td>processing capacity of each destination</td>
</tr>
<tr>
<td>0.4</td>
<td>cut-off range</td>
</tr>
<tr>
<td>1</td>
<td>number of metals (Au)</td>
</tr>
<tr>
<td>35</td>
<td>ore price ($/gr)</td>
</tr>
<tr>
<td>0.0 0.75 0.82</td>
<td>recovery for each destination</td>
</tr>
<tr>
<td>0.1</td>
<td>discount rate</td>
</tr>
<tr>
<td>1000</td>
<td>number of iterations</td>
</tr>
<tr>
<td>0.8</td>
<td>short-coming process blocks effect</td>
</tr>
<tr>
<td>0.1</td>
<td>MCAF</td>
</tr>
</tbody>
</table>

Table 4.3 – Parameter File for Case Study 2

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Resultant NPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranked Positional Weight</td>
<td>$ 11,727,741,729</td>
</tr>
<tr>
<td>SA with Transportation Penalty</td>
<td>$ 17,497,070,305</td>
</tr>
<tr>
<td>SA without Transportation Penalty</td>
<td>$ 17,471,534,712</td>
</tr>
</tbody>
</table>

Table 4.4 – Comparison of resultant NPVs
penalty’s other objective that is uniting the periods and smoothing the overall image contributed especially well to the development of the mine due to its contrariness to the ranked positional weight algorithm’s output. Overall, highest SA (with transportation penalty) yielded $6,069,328,576 higher NPV than ranked positional weight algorithm, which is a 52% improvement.

In Figure 4.6 and Figure 4.7, color scales on the right correspond to the periods in each image. It can be observed that SA algorithms enlarged the pit, allowing to make use of the deposit as a whole. Similar to case 1, SA with transportation penalty smoothed the pits that are formed for each period.

Process grade averages are provided in Figure 4.8 and Figure 4.9 from the summary of SA without transportation penalty. Although slightly decreasing from period to period, the average grade is maintained as can be seen in Figure 4.8. Process capacity push maximized almost all blocks destined to processes except for the last period in Figure 4.8 (b). The reason is that the pit has reached its full potential and there are no more high grade blocks to send at that point.

When two destinations’ overall average grade is compared in Figure 4.9, there is a significant difference; high grade process receives notably higher ore blocks.
(a) Ranked Positional Weight Algorithm

(b) SA with Transportation Penalty

(c) SA without Transportation Penalty

Figure 4.6 – Cross-section (Sections: 23, 0, 0)
Figure 4.7 – Cross-section (Sections: 22, 46, 0)

(a) Ranked Positional Weight Algorithm
(b) SA with Transportation Penalty
(c) SA without Transportation Penalty
Figure 4.8 – Average grades and number of blocks in each destination and period

(a) Low grade process

(b) High grade process
Figure 4.9 – Overall average grades compared among low and high grade process destinations
5 Improving Simulated Annealing by Adding Heuristic Memory

5.1 Introduction

Simulated Annealing is used when the model is very complex and produces big data. Only improvement is tracked and the decisions are made based on improving the objective function. Thus, SA is memoryless [Glover and Kochenberger, 2003]. Tabu search [Glover, 1989] [Glover, 1990] introduced the concept of memory by creating a dynamic list of forbidden solutions. This helps SA to focus on improving solutions. It is crucial to balance improvement and randomness to avoid entrapment in local maxima.

Still, there is a lot of potential information that can be deducted from the big data produced by SA. This information can change the general SA trend of following the improvement to following the predicted objective function distribution. Because the models are usually very complex, in most cases it will be extremely difficult to map and deduct the distribution from the whole model configuration. Instead, we are proposing to inspect the effect of some heuristic parameters on the objective function. This will force SA to have an informed walk rather than a random improving walk.

In our model, a candidate heuristic is "Number of Extracted Blocks". Depending on the pit, increasing or decreasing the number of extracted...
blocks may have an impact on NPV. In an open pit mine example, where the orebody is not reachable with given angles, it is not profitable to mine the pit. Records of the SA during the search that demonstrates the relationship between the number of extracted blocks and NPV can be seen in Figure 5.1. It is evident that they are inversely correlated. In this case, it would be logical for SA to incline the search towards less extracted blocks and finally decide that the optimal NPV is 0.

In another example, as can be seen in 5.2, the function does not seem to be linear. However, still it can be fitted into a more complex function. If there
is no correlation between the predicted heuristic and the objective function, any deductions from this graph will not be helpful to us. Whether there is a correlation between these two variables can be measured by correlation coefficients such as Pearson’s correlation coefficient in linear cases [Pearson, 1895] or copulas in non-linear cases.

In summary, the approach consists of predicting some heuristic parameters and observing their correlation with the objective function. In addition to optimizing to the real output, we propose simultaneously optimizing to the direction of these correlations. In the meantime, there will still be random-
ness to avoid local maxima. In mine production scheduling, other heuristics can be number of ore blocks vs NPV, block grade vs process destination, coordinates of the main ore clusters vs NPV, mine depth vs NPV and mine life vs NPV.

5.2 Method

The algorithm can be generalized as illustrated in Algorithm 5.1. First, the heuristics that are believed to have an effect on the objective function should be entered as inputs. The heuristic can be any variable with a quantity that changes in each solution.

For each solution the SA produce, this heuristic is quantified and saved along with the value of the objective function. When enough data is produced, a function fitting method is performed to deduct information of how this heuristic influences the objective function. This influence, along with a balancing parameter, will be included in the objective function.

<table>
<thead>
<tr>
<th>Algorithm 5.1 Improved Simulated Annealing by Adding Heuristic Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Defined heuristics ( h[n] )</td>
</tr>
<tr>
<td>while SA produces a solution do</td>
</tr>
<tr>
<td>for ( i = 0 ) to ( n ) do</td>
</tr>
<tr>
<td>Save ( h[i] ) and objective function output pair</td>
</tr>
<tr>
<td>Perform function fitting for ( h[i] ) vs objective function graph</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>objective function(_{new}) ← objective function + ( k_1 ) * fitted function * heuristic</td>
</tr>
<tr>
<td>end while</td>
</tr>
</tbody>
</table>
In practice, there are some challenges. First challenge was that updating of the fitted function at each iteration had a misleading effect. Because the coefficients of the function change at each step, this sometimes discounts the effect of the parameter itself. In other words, because the Metropolis criterion only looks at two latest solutions and subtracts them, the important part of the objective function becomes not the objective function itself but the difference. If the difference in coefficient makes up for the part of this difference, the result becomes less relevant. The solution for this was found to update the function at every \( n \) iterations instead of each iteration. This way, it was possible to anchor the coefficient so that only the effect on the parameter was seen in the objective function.

As illustrated in Algorithm 5.1, in the objective function, there should exist a coefficient \( k_1 \) to balance the effect of this heuristic and other components of the objective function. \( k_1 \) and \( n \) mentioned in the above paragraph are chosen empirically. One possibility is that they might be chosen according to experimenter’s confidence on the heuristic. Also, \( R^2 \), the coefficient of determination can be a part of \( k_1 \) such that this part of the objective function will only be active as much as the fit exists and how well it is. Thus, if the predicted heuristic is not correlated with the objective function, this part of the objective function will be discarded having \( R^2 = 0 \).

In the very first iterations of the program, because there are not as many data points collected, the predictions are less reliable. Hence, in practice,
it is more favorable to start using the heuristics only after iterating and generating data points for a given amount of times.

Fitting functions can vary widely and the choice can be done according to the graph shape. Examples could be linear regression, polynomial regression with cross validation [Kohavi et al., 1995] to avoid overfitting. Expectedly, every case is different and will give various results depending on the type of function chosen.

5.3 Case studies extended to Heuristic Memory Based Simulated Annealing

Heuristic memory based SA is designed to produce better results in the same amount of time compared to standard SA. In these studies, linear regression is used as the fitting function. Only one heuristic is used, which is the number of blocks extracted. Its accomplishment is demonstrated through running the cases also with this module.

The data and parameters of Case 1 were input to heuristic memory based SA both for the case with and without transportation penalty. The comparison is done through evaluating the results obtained under being run for the same amount of time with other algorithms.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Resultant NPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranked Positional Weight</td>
<td>$1,613,205,645</td>
</tr>
<tr>
<td>SA with Transportation Penalty</td>
<td>$1,687,973,927</td>
</tr>
<tr>
<td>SA without Transportation Penalty</td>
<td>$1,688,949,559</td>
</tr>
<tr>
<td>SA with Memory and Transportation Penalty</td>
<td>$1,689,934,020</td>
</tr>
<tr>
<td>SA with Memory without Transportation Penalty</td>
<td>$1,690,591,884</td>
</tr>
</tbody>
</table>

Table 5.1 – Case 1: Comparison of resultant NPVs

The results are given in Table 5.1. If SA with transportation penalty is compared with SA with memory and transportation penalty there is an improvement of $1,642,325 and SA without transportation penalty is compared with SA with memory without transportation penalty there is an improvement of $1,960,093. In fact, even SA with memory and transportation penalty produced a better result than standard SA without transportation penalty. In other words, our approach of SA allows us to utilize more complex models, while taking shorter amount of time.

Similarly, all algorithms were tested in the data and parameters of Case 2. The results are given in Table 5.2. If standard SA with transportation penalty is compared to SA with heuristic memory and transportation penalty, there is an improvement of $313,956 and if standard SA is compared to SA with heuristic memory, there is an improvement of $5,451,993.

In summary, in all cases, SA with heuristic memory performs better than standard SA. The only small drawback with this approach that it requires some memory to store the data points. This memory accumulates as SA is
run for longer amounts of time. However, this growth is linear and storage is minimal (only two points are saved for every feasible solution). Also, there are ways to shrink the memory usage such as grouping very similar points and assigning them weights.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Resultant NPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranked Positional Weight</td>
<td>$ 11,727,741,729</td>
</tr>
<tr>
<td>SA with Transportation Penalty</td>
<td>$ 17,497,070,304</td>
</tr>
<tr>
<td>SA without Transportation Penalty</td>
<td>$ 17,471,534,712</td>
</tr>
<tr>
<td>SA with Memory and Transportation Penalty</td>
<td>$ 17,497,384,260</td>
</tr>
<tr>
<td>SA with Memory without Transportation Penalty</td>
<td>$ 17,476,986,705</td>
</tr>
</tbody>
</table>

Table 5.2 – Case 2: Comparison of resultant NPVs
6 Conclusions and Future Work

Developing techniques and algorithms for open pit mine production has started in 60s and improved a lot ever since. Especially bringing concepts from other fields such as Statistics, Mathematics, Computer Science and even Condensed Matter Physics, contributed a fair amount to finding optimal or almost-optimal pits. Overall, better evaluating the mine results in higher net present values thus better profits.

Having compared various techniques, it can be observed that iterative approaches and ultimate pit limit based algorithms are very successful and fast but yield sub-optimal results in multi-period models due to inability to specify capacity constraints. Exact methods returns the optimal pits, every kind of constraint can be included in these methods. However, with so many parameters and the large input space cause these techniques to be impractically slow. Improving technology will ultimately pave the way for these category of techniques. Heuristic based methods may seem insufficient by itself, but combining heuristic based methods with other methods such as simulated annealing practically yield very good results. This kind of combination seems to be the strongest option in especially very large deposits.

Using SA after a heuristic based method, guarantees that it will either produce a better solution, or return the initial solution. It is true that SA takes time to reach the optimal value. However, unlike exact methods, it
can be stopped at any point and best solution found so far can be returned. In other words, there is nothing to lose and so much to gain by using SA. Moreover, almost all parameters can be integrated into SA, such as process capacity, transportation cost and multiple process destinations, which are impossible to integrate in some other techniques.

In the case studies, it is observed that there can be very large gains (Case 2: 50%) of the revenue. The average grade and number of blocks sent to destinations were overall stable. Furthermore, smoothness of the pit was increased through assigning transportation penalty for distinct periods in adjacent blocks. SA was even more enhanced and sped up by our approach of heuristic memory. As demonstrated by extending the case studies, the reached solution’s revenue obtained in the same amount of time has increased in each case.

In the case studies of the heuristic memory based SA, a linear fitting function was used. Efficiency of the memory enhancement can be increased through improving this fitting function. Also, in our cases most parameters related to heuristic memory based SA was chosen empirically, such as when to produce the first function, how often to update the function and how to balance the optimal function with the heuristic. Research can be done to analyse how to optimize these parameters.
Largest issue in all empirical machine learning applications is the parameter selection. This is also true for all types of SA. Selection of SA related parameters such as the temperature, number of iterations, maximum number of solutions at each temperature. These parameters can effect the running time of the program to a great extent. If the parameters are poorly set, and the program is run for a short time, the results may not be optimal.

The program is optimized to work with metal ore mines and presently not applicable to coal mines as it uses grade data of the blocks and coal mine structure is different. The grades are assumed to be deterministic and NPV calculations are made based on this assumption. As these grades are generated by Gaussian simulation of the real samples and given that the deposit is not homogeneous, if number of samples are small, program’s generated solutions might be different from the reality. The range of this difference can be measured using risk analysis techniques by running the program multiple times with different grade data.

In the program, long term prices that are based on historical data are used, which is a common procedure in mine production scheduling. This price is assumed to be deterministic. Stochastic optimization methods has been proposed [Kumral, 2010] [Kumral, 2011] to deal with the fluctuations in price. However, these methods work only on small datasets and not realistically applicable on large orebodies. As the problem size issues are solved, these methods can be used. Risk analysis can also be used to estimate the range
of total NPV thus make a better informed decision. In general, although our program produces a long term mine production schedule, throughout the mine life, scheduling programs are run multiple times with as parameters vary in time.

Also, as the mining takes place, there is a constant new information flow from the drill holes. An artificial intelligence technique such as reinforcement learning can be introduced to learn from these updates and make better predictions. Reinforcement learning can incorporate these changes faster to make a new calculation and at the same time, it can decide which sample areas are less reliable.
References


Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and


