Western Australian School of Mines

A Mixed Integer Programming Approach for Transitioning from Open-pit to Underground Mining

Joyce Sze Yee Chung

This thesis is presented for the Degree of

Doctor of Philosophy

Of

Curtin University

November 2020

DECLARATION

I declare that this thesis contains no material previously published by any other person except acknowledged resources. This thesis contains no material which has been accepted for the award of any other degree in any other universities.

I acknowledge that copyright of all materials contained in the dissertation resides with the copyright holder(s) of that material.

Signature :

Date : 12/11/2020

PUBLICATIONS DURING CANDIDATURE

Chung, J., E. Topal, and O. Erten. 2015. "Transition from open-pit to underground - using Integer Programming considering grade uncertainty." The 17th Annual Conference of the International Association for Mathematical Geosciences, Freiberg, Germany, 5-13 September 2015, pp.268-277.

Chung, J., E. Topal, and A.G. Ghosh. 2016. "Where to make the transition from open-pit to underground? Using integer programming." *Journal of the Southern African Institute of Mining and Metallurgy* 116 (8):801-808.

Chung, J., M. Asad, Topal E., and A.G. Ghosh. 2016. "Determination of the Transition Point from Open-pit to Underground Mining." Ninth AusIMM Open-pit Operators' Conference 2016, Kalgoorlie, Australasia Institute of Mining and Metallurgy, pp.96-103.

ACKNOWLEDGEMENTS

I would like to extend my appreciate to the following people for providing me assistance in completing this research:

- Associate Professor Mohammad Waqar Asad, who has offered generous support and guidance as the main supervisor throughout this project.
- Professor Erkan Topal, who initiated and offered assistance and guidance throughout this research.
- All the staffs (academic and support) from WA School of Mines, Curtin University, who have continuously been encouraging and supporting me.
- Research scholars at the WA School of Mines whom I frequently have discussions with.
- My family and friends, for their support and encouragement throughout the process.

ABSTRACT

In the last decades, mine planning and optimization is predominantly focused on either open-pit or underground mining method only. It has neglected the other options as a viable option in the early stage. This situation commonly happened in cases where shallow deposits are mineralized to a considerable depth. These deposits are usually planned using open-pit mining. Subsequently, underground mining strategy has emerged as an option when the open-pit operation is approaching the ultimate pit limit. At this point of time, the 'transition problem' arose in which a decision on either to extend the pit or move to underground mining must be made. To solve this problem, it is suggested that a mine optimization process which considers both open-pit and underground mining be implemented simultaneously in order to generate economic benefits and provide a clear guide to the decision-making process during the operation. From the review of the current literature, it has revealed that there is a demand on methodology which can solve the transition problem. Hence, the aim of this research is to develop a mathematical model to solve the transition problem.

This research is focused to answer the question of '*where and when to make the transition*'. To incorporate the practicality aspects in the combination of open-pit and underground mining strategy, the framework of this research is to develop mathematical models which consider not only open-pit mining and underground mining concurrently, but also crown pillar placement. Two new mathematical models are developed. The first proposed optimization model which aims to generate the optimal mining layout and optimal transition point by maximizing the project value. This model has answered the first part of the question (*where* to make the transition). The second optimization model is developed to solve the transition problem by taking time-variant factors into account. This model can provide the optimal transition point, transition period and crown pillar location. Hence, it is possible to answer the questions of *where and when* simultaneously.

To solve the transition problem, the computation complexity is increased manifold compared to a standalone optimization problem. Thus, the scale of the problem is a major challenge in this research. Two main strategies to reduce the scale of the problem are presented in this research. Firstly, a stope-based methodology is implemented for

underground mining. This methodology is used to search and retain the profitable stopes and eliminate those unprofitable stopes. The second strategy is an agglomerative hierarchical clustering algorithm which is employed by open-pit mining. This strategy is manipulated to cluster blocks within the ultimate pit by using a similarity index. A new similarity index formula is proposed and employed in this research.

The proposed optimization models and hierarchical clustering algorithm are tested by using two-dimensional datasets. The results are verified. The outputs of the models proved that all the constraints that are designed for open-pit mining, underground mining and crown pillar are correctly formulated and fulfilled.

The implementations of the proposed models are presented in this research. The solutions confirm that the proposed models are capable to solve the transition problem by maximizing the undiscounted or discounted cashflow. The Transition Point Model achieves a maximized undiscounted cashflow of $3.74 billion and the Transition Period Model attains $2.60 billion of net present value with no production delay while making the transition from open-pit to underground mining. An additional scenario is completed while considering two schedule period of production delay during the transition, the result is $2.5 billion.

Additionally, an implementation of the hierarchical clustering algorithm along with the proposed optimizations are presented. The hierarchical clustering algorithm is utilized to clustering the blocks within the ultimate pit limit and it successfully reduces the problem size of open-pit mining by 85%. The implementation with the clustering algorithm output proves that the hierarchical clustering algorithm is capable of reducing the open-pit problem size and improve solution time. Along with the result of the hierarchical clustering algorithm, the proposed Transition Point Model and Transition Period Model have generated the result of approximately $191 million of undiscounted cashflow and $146 million of net present value, respectively.

In conclusion, two mathematical models have been developed, validated, and implemented in case studies. Besides, two-scale reduction strategies are incorporated into the research to manage the scale issue. It is proved that the models can solve the transition problem by maximizing the value of the project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

This chapter provides a framework for the remainder of the thesis and is divided into five sections as follows:

i. Problem statement (Section 1.1);
ii. Research objectives and motivation for the proposed research (Section 1.2);
iii. Significance of research (Section 1.3);
iv. Original contributions of this research (Section 1.4);
v. Thesis plan (Section 1.5).

1.1 PROBLEM DESCRIPTION

Open-pit mining is the most broadly applied mining strategy as it is generally and economically superior to underground mining methods. Open-pit mining is preferable because of its numerous advantages such as mining recovery, production capacity, mining capacity, dilution, safety and others. However, it leaves a large mining footprint and leads to environmental and social unfriendliness. Additionally, open-pit mining method is only suitable for shallow deposits due to its sensitivity to haulage and stripping costs.

In contrast, underground mining is more favorable in the social and environmental perspectives as it creates less disturbance to the earth topography. However, due to its higher mining cost than open-pit mining method, it needs to be more selective by minimizing the waste movement from underground to surface. Furthermore, underground mining requires huge up-front investment cost for pre-production development such as decline, ore drives and ventilation. Besides, underground mining has more complex mine operations in terms of production, planning, environment, and safety. Hence, underground mining is usually applicable for deep deposits where open-pit mining cost outweighs the underground mining cost.

There are some shallow deposits which change considerably in geometry along the strike. Most of these deposits are often planned and mined using open-pit mining. Afterwards, when these deposits become burdened with excessive stripping, transiting to underground mining then becomes a viable strategy to extract the remaining reserves. The implementation of both open-pit and underground mining methods for ore mining purposes is known as the combination of open-pit and underground mining strategy.

Conventionally, during the planning stage, open-pit and underground mining are often studied individually, e.g. start with open-pit mine project and initiate the underground mine study at a later stage. However, this approach will directly impact the value of the project and its resource utilization due to the arbitrary decision-making on crown pillar location and transition period. For instance, in some cases, if transition took place in an earlier stage, better economic outcome could have been possible. In the past decades, very few studies have been conducted to optimize the mine planning for the combination of open-pit and underground mining strategy which integrates both open-pit and underground mine planning into a single approach to maximize the resource utilization and mine project value.

The shallow deposits that extend to a considerable depth may potentially experience a 'Transition Problem'. The transition problem emerges when the decision needs to be made about whether to (1) expand the pit, (2) switch to underground mining to recover the deeper part of the deposit or (3) cease the mining operation. Thus, the transition problem is an indication of *when* and *where* to make the transition to capitalize on the value of the project. In this respect, the timing of the transition is known as 'Transition Period' while, the location to switch to underground mining is known as 'Transition Point'.

With the option (1) and (2) in place, the first option will incur significant haulage and stripping costs due to the large pit cutback and the second option may be the optimal strategy for the remainder of the deposit at a greater depth, as its mining cost is not as sensitive to depth as the open-pit mining method. Figure 1-1 shows the schematic of combination of open-pit and underground mining strategy and transition problem.

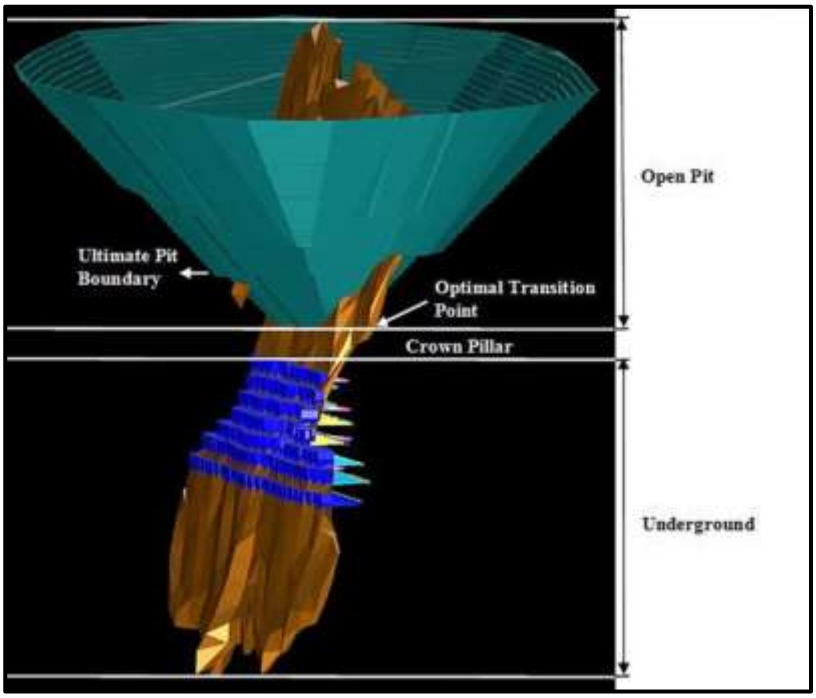

Figure 1-1: Schematic of combination of open-pit and underground mining strategy and transition problem (Chung, Topal, and Ghosh 2016)

The accomplishment of an optimal mine plan for the combination of open-pit and underground mining strategy is possible only if the feasibility of the study stage of the project establishes the optimal transition point and optimal transition period. Ideally, in the practice of the combination of open-pit and underground mining strategy, open-pit mine operation should cease when:

- open-pit mining cost is greater than underground mining cost;
- crown pillar location is considered and optimized;
- resource and reserve distribution are optimized.

In the case that underground mining method is neglected as the viable strategy at the initial stage, significant issues will emerge while completing the mine study for transition problem. The first issue is the crown pillar location. Ideally, the crown pillar should be located at the level or location with the least revenue (i.e. low-grade ore). However, in the conventional approach, the initial defined ultimate pit limit (UPL) will drive the crown pillar placement. This situation forces the crown pillar location to be arbitrary in the decision-making process. Due to the static crown pillar location, it may lead to two consequences which are loss of reserves and loss of project value.

Figure 1-2 shows a schematic of the resource distribution. Following the discussions above and schematic presented in Figure 1-2, a robust mine planning and optimization tool which can outline the resource distribution for open-pit and underground mining is required. The tool must consider the following aspects to ensure its practicality:

i. the integration of both open-pit and underground mining strategy and schedule;
ii. the ability to determine the optimal mining strategy for the mine project;
iii. the location of crown pillar;
iv. the capital required for underground development;
v. smooth transition from open-pit to underground mining;
vi. the ability to schedule production delays during transition; if required.

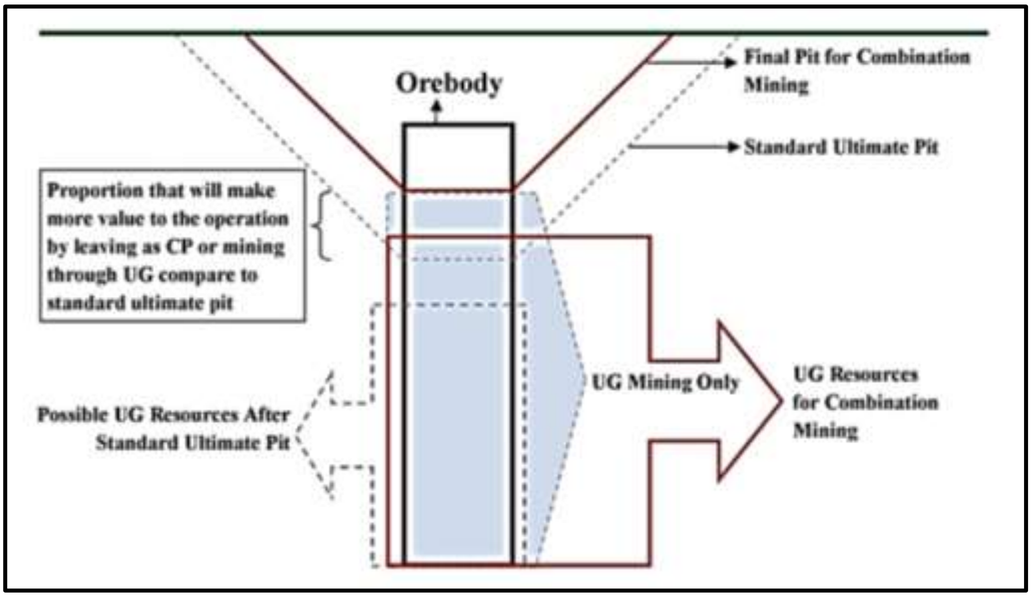

Figure 1-2: Schematic of resource distribution for combination of open-pit and underground mining strategy (Chung, Topal, and Ghosh 2016)

1.2 RESEARCH OBJECTIVES AND MOTIVATION

The primary goal of this research is to construct and implement mathematical models for the combination of open-pit and underground mining strategy which can resolve the transition problem while satisfying open-pit and underground mining constraints. The constraints including, but not limited to, reserve constraints, open-pit block sequence, underground mine design restrictions, underground vertical access limitations and crown pillar placement. Besides, the expected subsidiary outcome of the research will be able to provide guidance on an optimal mining strategy.

The objective of this research was reached through the following steps:

- Review the established approaches regarding the transition problem;
- Develop optimal and reliable mathematical programming models to determine the optimal transition point and transition period;
- Incorporate production delay option throughout the transition from open-pit to underground mining.

The utilization of a mathematical model to solve the complex transition problem is urged due to its performance. However, generally, mathematical models are computationally complex due to a large number of decision variables which lead to difficulty in solving these large-scale and NP-hard optimization problems. In order to handle the scale issue, the secondary objective of this research is to introduce data clustering approach that will reduce the number of decision variables in the mathematical model for the transition problem.

The motivation of this research originates from the often overlooked simultaneous mine planning and optimization process which combines open-pit and underground mining strategies in the past decades. In cases where a combination of open-pit and underground mining strategy can be practiced, a conventional approach (i.e. study open-pit and underground mining individually by initiating open-pit mine study first) is often preferred or chosen. The conventional approach tends to ignore the 'best' resource distribution for the deposit which leads to sub-optimal project value and reserve utilization.

1.3 ORIGINAL CONTRIBUTIONS OF THIS RESEARCH

The scope of this research project is limited to optimizing the transition problem for the combination of open-pit and underground mining strategy that maximizes the net present value (NPV) and generates an optimal mining layout for a mining operation. It aims to define the optimal transition point, subsequently, providing a transition schedule for the combination of open-pit and underground mining strategy. Additionally, to solve this NP-hard transition problem, development and implementation of a brand-new hierarchical clustering algorithm is formed as part of the scope in order to handle the large-scale problem. Commonly, input values to the models are subjected to uncertainty. Hence, those inputs can influence the reliability of the result that lies outside the scope of the study.

1.4 SIGNIFICANCE AND RELEVANCE

In the last decades, as many open-pit mines are reaching their critical stage, the transition problem becomes one of the most significant engineering issues for mining engineers. In addition, due to the increased demand for raw materials, the transition problem has been prioritized to ensure the continuity of the mining activities. There are a few examples of mines that have made the transition from surface to underground mines such as Chuquicamata mine in Chile (Flores and Catalan 2019), Grasberg mine in Indonesia (Sulistyo, Soedjarno, and Simatupang 2015) and Sunrise Dam in Western Australia (Opoku and Musingwini 2013). Hence, this research project offers a new approach towards the planning and optimization of the transition problem which benefits the mining industry in several aspects such as:

- maximize resource utilization;
- maximize project value;
- improve life-of-mine (LoM);
- smooth transition from open-pit to underground;
- schedule production delay or interruption during transition within LoM plan, if required.

1.5 THESIS OVERVIEW

The remainder of this thesis is divided into six chapters which are:

Chapter 2 studies the relevant literature in open-pit mining, underground mining and combination of open-pit and underground mining strategy. The chapter also discusses the challenges of the proposed model and tactics to handle the issues along with its relevant literature.

Chapter 3 presents the mathematical formulations for the transition point and transition period, which focus on the determination of the transition point and transition period. A validation process of the model is also included in this chapter.

Chapter 4 explores the implementation of the mathematical model using exact methods presented in Chapter 3 and establishes the computational complexity in realistic data sets.

Chapter 5 highlights the clustering algorithm for the transition problem and demonstrates its validity. This chapter also compares the result generated by the clustering algorithm with the results of exact methods.

Chapter 6 demonstrates the implementation of the clustering algorithm and the mathematical models using the large-scale case study.

Chapter 7 concludes all the outcomes of the research and proposes suggestions for further study.

Chapter 2

LITERATURE REVIEW

The objective of this chapter is to discuss the relevant literature and how the transition problem was addressed in this research. The chapter provides a brief overview on a typical mining process and will be followed by three substantive sections.

Section 2.2 discusses the related literature for open-pit mining, underground mining and combination of open-pit and underground mining strategy, which includes optimization of the mining layout and production schedule.

Section 2.3 provides an outline of the research methodology and briefly discusses the challenges of the transition problem.

Section 2.4 describes the related literature with respect to the clustering algorithms.

2.1 MINING PROCESS

A typical mining process starts from exploration, followed by orebody modelling. During the orebody modelling stage, a geological block model is generated. Then, the geological block model is transformed into an economic block model with consideration of economic and technical factors such as commodity price, recovery rate, mining cost and others.

The economic block model is used to complete the mine planning and optimization process. During the earliest stage of mine planning and optimization, it is often questioned about the most appropriate mining method (Topal 2008). Generally, the preferred mining method is pre-selected based on the knowledge of the engineer or characteristics of the orebody. Then, taking account of the selected parameters and capacities such as mining capacity and processing capacity, a mining layout and plan of extraction are produced. The mine planning and optimization process are critical as they provide guidance on how to extract the valuable material and attempt to optimize the project value over the LoM (Dagdelen and Johnson 1986; Caccetta 2007). The last step of the mining process is the execution of the plan.

2.2 OPEN-PIT AND UNDERGROUND MINE PLANNING AND SCHEDULING

Mine planning and optimization play a critical role in the mining process. This process directly, and indirectly, impacts the economic prospect of the project. Nowadays, numerous mine planning and optimization methodologies, techniques and approaches are available for open-pit and underground mining. The available techniques and approaches for each mining method will be discussed in the following sections.

2.2.1 Open-pit mining

In open-pit mining, UPL optimization is aimed at defining the size of extraction and volume of extracted material that maximizes the undiscounted value of a project subjected to pit slope ore precedence requirements. The general idea is to make sure that a given block is only extracted if the precedence blocks have been extracted. In the past decades, the three most notable methods to define the UPL are the floating cone (FC) method (Carlson et al. 1966), Lerchs and Grossmann (LG) method (Lerchs and Grossman 1964; Whittle 1990) and Pseudoflow method (Hochbaum and Chen 2000; Hochbaum 2001).

The FC is a heuristic approach which involves an iterative process. This method searches through the block model by assessing the value of the cone. The FC method is rarely used by the mining industry these days due to the lack of flexibility of the algorithm. It is unable to detect the mutual support among the different parts of the orebody, and it only considers those blocks within the cone. The LG algorithm (Whittle 1990) is the most notable algorithm which provides a computational and tractable method for open-pit mining layout optimization. The LG algorithm is based on graph theory. It aims to define the maximum closure of a weighted directed graph by using a maximum-weight closure algorithm to maximize profit. Hence, the vertices, weights and arcs in the algorithm represent the mining blocks, net profit, and slope constraints respectively. Likewise, the Pseudoflow method (Hochbaum and Chen 2000; Hochbaum 2001) is the most recent developed algorithm and it is widely utilized by many mine optimization software. The Pseudoflow algorithm inherited the LG algorithm normalized trees and further developed it to a general network flow model. It solves the maximum flow model on general graphs, hence, it is generally more computationally efficient compared to the LG algorithm.

Open-pit scheduling is the next process after the generation of UPL. This scheduling process defines the sequence of production that maximizes the discounted value of the operation while satisfying precedence and operational capacity constraints. Some formulations also include grade control and stockpile constraints. Most of the current available literature, can be divided into two main groups namely heuristic algorithms or exact mathematical models (Askari-Nasab, Awuah-Offei, and Eivazy 2010).

Various intelligence-based algorithms have been presented in the past. Some of the notable works in this area are proposed by Tolwinski and Underwood (1992), Denby and Schofield (1994), Askari-Nasab (2006) and Askari-Nasab and Awuah-Offei (2009). Tolwinski and Underwood (1992) suggested an approach which integrated dynamic programming, stochastic optimization and machine learning which was successfully implemented by Elevli (1995). Meanwhile, Denby and Schofield (1994) developed a genetic algorithm for the UPL and production scheduling problem. The recommended algorithm starts by populating random pits and then, assesses the function of the fitness of the populated pits. The algorithm is an iterative process as it stops when re-occurrence of a state happened, and no further improvement is generated. The major drawback of the heuristic-based algorithms is that the optimally of the solution is unable to be measured. In addition, most of the results are also unable to be reproduced since they are probability-based.

Furthermore, many operations research (OR) based methodologies such as linear programming (LP) and mixed-integer linear programming (MILP) are presented to solve the open-pit scheduling problem. There are several reasons as to why MILP and Integer Programming (IP) are attractive and there are:

- Cut-off grade can be optimized as it allows the model to determine if the material is mined and the mineable material is treated as ore or waste.
- They can integrate the optimization of a user-defined weighted function of the life-of-mine and NPV.
- They are flexible to cater for complex mine operations such as multiple products, destinations, and sites.
- They have the sensitivity analysis capability.

Johnson (1968) developed the first LP model for open-pit scheduling problem, which inspired Gershon (1987) to create a MILP model on the open-pit scheduling problem. Other than that, there are some noteworthy models that include, but are not limited to, Caccetta and Hill (1999), Dagdelen and Kawahata (2007), Askari-Nasab, Awuah-Offei, and Eivazy (2010), Eivazy and Askari-Nasab (2012) and others. However, due to the scale of the open-pit problem, the problem becomes computationally intractable. Hence, a variety of methods have been proposed to handle the large-scale problem such as the reduction of the number of binary integers as suggested by Ramazan and Dimitrakopoulos (2004) and the Lagrangian relaxation method proposed by Dagdelen and Johnson (1986). Ramazan and Dimitrakopoulos (2004) introduced a new MILP method that intended to reduce the number of binary variables by considering two aspects: (1) only positive value blocks defined as binary and (2) remaining variables are defined as linear. Additionally, Dagdelen and Johnson (1986) presented a Lagrangian relaxation method which uses the Lagrangian multipliers to decompose the complex problem into smaller problems, for instance, by solving the long-term open-pit optimization problem by decomposing the multi-period problem into multiple single-period problems.

2.2.2 Underground mining

In underground mine planning and optimization, the main components that are responsive to the optimization process are stope boundary optimization, development placement and production scheduling (Little 2012). This research project will focus on two components which are stope boundary and production scheduling. Defining an optimal stope layout is one of the important tasks in underground mine planning. Stope layout is known as a group of blocks that lies within an envelope and they are economical to be extracted as a whole. Meanwhile, stope layout optimization is a process to obtain the best combination of blocks to form stopes within the block model which generates the best project values and reserve utilization. As a result, the set of profitable stopes which has the highest return will form an underground mining layout. Numerous approaches have been presented to optimize the stope layout. Some of the noteworthy algorithms were developed by Alford (1995), Ataee-Pour (2004), and Grieco and Dimitrakopoulos (2007).

The FS algorithm is the most well-known stope layout algorithm presented by Alford (1995). The FS algorithm used a sophisticated, rectangular block as the minimum stope size that is floated through the block model. This algorithm has been improved and developed as the Vulcan Stope Optimizer (Maptek 2011). The Maximum Value Neighborhood (MVN) algorithm is another stope layout optimization method which was introduced by Ataee-Pour (2004). This algorithm is a heuristic-based approach which defines stope boundary by assessing the best neighbourhood for each block. With the possible combination of neighbourhood, the one with the maximum value is chosen. Grieco and Dimitrakopoulos (2007) presented a probabilistic mathematical programming model to solve the stope layout optimization. MILP model is developed to determine a stope size based on the number of blast rings being included in a stope. Additionally, there are other studies for underground stope layout design problem included, but are not limited to, the octree division algorithm (Cheimanoff, Deliac, and Mallet 1989), Stopesizor algorithm (Alford, Brazil, and Lee 2007), and the transformed stope boundary optimization (Topal and Sens 2010).

Apart from stope layout design, long-term underground mine production scheduling is important. Numerous approaches and algorithms are available in this respect. MILP and IP play a significant role in long-term underground production scheduling optimization. Over the years, many mathematical models have been developed to the optimize underground production scheduling problem. Those notable works include, but are not limited to Trout (1995), Topal (2008), Nehring et al. (2010), and Little and Topal (2011).

Trout (1995) developed a MILP model to obtain the optimal production sequence for a sublevel stoping method. The aim of the model is to maximize the NPV of the mining operation. This model was implemented on a copper operation and its efficiency proved. Nehring and Topal (2007) enhanced the MILP model by introducing a new formulation for limiting multiple exposure of fill masses. Following that, Topal (2008) introduced variable reduction strategies associated with MILP model which has increased the efficiency of the mathematical model. Two strategies have been introduced: which are defining (1) machine limitations and (2) introducing early and late start algorithm to narrow the observation period for the machine placement. Implementation has been demonstrated by using the Kiruna Mine dataset where significant reduction of variables resulted in increased computational efficiency. Besides, Nehring et al. (2010) presented a MILP model which integrates short-term and long-term production scheduling concurrently. The objective function consists of minimizing the deviation of mill feed grade in a short-term schedule, while maximising NPV in the long-term schedule and cash penalties for feed grade to ensure operational and recovery efficiency. Moreover, Little and

Topal (2011) proposed an IP model for stope layout and production scheduling optimization concurrently with the objective to maximize the NPV of the operation. The authors proposed two concepts to minimise the number of integer variables, such as combining blocks into a potential stope and removing negative value stopes.

Many approaches for both layout and production schedule optimization have been present for open-pit and underground mining method in the last decades. However, the optimal solution for realistic open-pit production scheduling optimization remains impossible. Hence, continuous efforts are required to improve the optimization process and to obtain the optimal schedule for open-pit within reasonable timeframes.

On the other hand, a mathematical model is advantageous for both underground mining layout optimization and production scheduling optimization. It guarantees the optimal solution and helps to ensure efficiency of the mine operations. However, the number of variables involved in a mathematical model are critical and it should be kept at the minimal level at all the time.

2.3 TRANSITION FROM OPEN-PIT TO UNDERGROUND - COMBINATION OF OPEN-PIT AND UNDERGROUND MINING STRATEGY

Apart from the conventional approach (Section 2-1), a few studies (Nilsson 1992; Camus 1992; Arnold 1996; Tulp 1998; Fuentes 2004; Brannon, Casten, and Johnson 2004; Fiscor 2010) share approaches and algorithms to solve the transition from open-pit to underground mining.

Soderberg and Rausch (1968) introduced a surface-to-underground stripping ratio approach that delineates mining cost, ore recovery and dilution which suggests that these are the controlling factors for the transition problem. It proposed a breakeven cost differential relationship in Equation (2-1) that accounts for open-pit mining cost per tonne of ore (m_{OP}), underground mining cost per tonne of ore (m_{UG}), and the open-pit waste stripping cost per tonne of waste (w_{OP}) and calculates the indicated stripping ratio (ISR). Accordingly, if the stripping ratio corresponding to a mining block is less than ISR in Equation (2-1), then open-pit mining would be economical, otherwise underground mining becomes economical.

$$ISR = \frac{m_{UG} - m_{OP}}{w_{OP}} \tag{2-1}$$

Nilsson (1982, 1992) proposed a cash flow analysis-based trial and error method that relies on the experience of a mine planning specialist. The author suggested the aspects which need to take into consideration and may influence the transition problem such as stripping ratio, interest rates and production costs. However, this approach is purely based on the knowledge of the mine planner. Hence, it does not describe by any optimization tactic. Abdollahisharif et al. (2008) modified the Nilsson (1982, 1992) method and applied an iterative approach that accounts for alternative crown pillar locations and selects the best among these feasible alternatives as the transition point.

Camus (1992) applied the Lerchs and Grossmann (1965) algorithm implemented on a modified economic block model. The block value is calculated by a modified economic block value (EV_m) accounting equation. The modified accounting equation integrates profit, open-pit cost and underground mining cost, as present in Equation (2-2). Hence, each block has to be able to pay both open-pit stripping cost and potential underground benefit if it needs to be mined through open-pit mining or, vice versa. For instance, if the profit and stripping cost for open-pit is \$50 and \$20 respectively, the block value for open-pit mining is \$30 ($Block\ Value_{open\ pit}$). For the same block, if the underground block value is \$20 ($Block\ Value_{underground}$), the modified block value is \$10 (EV_m). In this case, open-pit is the optimal mining method for the block. On the flip side, if EV_m is less than zero, underground mining method is the optimal mining method for the block. Camus (1992) claimed that the UPL generated using this modified economic block model provides the location (transition point) to switch from open-pit to underground operation.

$$EV_m = Block\ Value_{open\ pit} - Block\ Value_{underground} \tag{2-2}$$

Bakhtavar, Shahriar, and Oraee (2008) proposed a heuristic algorithm that maximizes the undiscounted value from both open-pit and underground mining. The approach keeps the first three levels in an open-pit operation and compares the value of the remaining levels for both open-pit and underground options. When the underground mining value is higher than open-pit mining, the last level of the pit is divided into sublevels and then the comparison is re-run. Transition from open-pit to underground happens when underground mining generates a higher value on the level than open-pit mining. The method can determine the open-pit layout, transition point, location of the crown pillar and a profile of underground levels.

Bakhtavar, Shahriar, and Mirhassani (2012) presented a two-dimensional IP based mathematical model to resolve the transition problem. This model aims to maximize the undiscounted value of the transition from open-pit to underground mining by catering for an objective function that includes both open-pit ($opbv_a$) and underground value ($ugbv_a$) of a block as shown in Equation (2-3). The constraints taken into consideration in the model include reserve restriction constraints, slope constraints, minimum stope width and height constraints, maximum stope width and height constraints, crown pillar constraints and level-based reserve restriction ('at most one method for each row') constraints. The proposed model has successfully demonstrated the complexities of the transition problem and guaranteed optimality. However, it is unable to be implemented in real applications due to the computational cost.

$$Objective\ function{:}\ \ Z = \max \sum opbv_a + ugbv_a \qquad (2\text{-}3)$$

Roberts et al. (2013) proposed an iterative process that applies the incremental value concept that ranks mining blocks to establish their potential for open-pit or underground mining. In this respect, the incremental value of a block (IV_a) is the difference between the discounted value per tonne of a block if mined by underground ($UGDV_a$) and the discounted value per tonne of block if mined by open-pit mining ($OPDV_a$), as presented in Equation (2-4). The underground discounted value per tonne of a block ($UGDV_a$) is calculated based on an equation by taking into account various important parameters such as net revenue of a block (r_a), processing cost for underground operation (p_{ug}), mining cost (m_{ug}) and cost associated with avoiding underground mining (v). On the other hand, the open-pit discounted value per tonne of a block ($OPDV_a$) is the maximum function of the discounted profit of open-pit mining which only includes net revenue of a block (r_a) and processing cost for open-pit operation (p_{op}). The equations for discounted value accounting for underground and open-pit mining are presented in Equation (2-5) and (2-6), respectively. Therefore, the positive and higher incremental value indicates the suitability of a block for underground mining and a negative incremental value which indicates the suitability of block for open-pit mining.

$$IV_a = UGDV_a - OPDV_a \tag{2-4}$$

$$UGDV_a = \frac{r_a - p_{ug} - m_{ug}}{(1+d)^{uy}} - \frac{v}{(1+d)^{oy}} \tag{2-5}$$

$$OPDV_a = \max\left(\frac{r_a - p_{op}}{(1+d)^{oy}}, 0\right) \tag{2-6}$$

where

d discount rate

oy year in which block is mined through open-pit

uy year in which block is mined through underground

Opoku and Musingwini (2013) introduced a structured methodology towards solving the transition problem. Initially, the procedure applies open-pit mining for the entire mineral resource, then it applies the option to switch from open-pit to underground mining and finally applies the option for underground mining for the entire mineral resource. Finally, it applies NPV, stripping ratio, average grade, refined metal as indicators to rank the three options and selects the option with highest rank.

Dagdelen and Traore (2014) applied a sequential procedure for the transition problem. The procedure creates a UPL through the Whittle commercial mine planning software, defines the underground stope layout using Studio 5 and EPS software and then finally applies OptiMine scheduler to define the production schedule over life of operation. However, this sequential procedure is prone to sub-optimality with issues around the location of a crown pillar.

Whittle et al. (2018) developed a UPL optimization algorithm while solving the transition problem at the same time. The authors framed a maximum graph closure problem which can define the optimal mine outline for the combination of open-pit and underground mining strategy. Digraph is used to tackle the problem which is similar to Whittle (1990). However, Whittle et al. (2018) included two more types of arcs to cover the underground mining option. The first type of arc is to bring underground opportunity cost into the optimization problem. Hence, the tail of the arc is in the open-pit vertices and the head is in the offset underground vertices (Z-elevation). The elevation offset is to accommodate the crown pillar requirement. Furthermore, the second type of arc is to satisfy the overall underground crown design requirement. A non-trivial strongly connected subgraph is introduced to achieve a prescribed shape.

In the past, very limited studies have been conducted to obtain the optimal transition period or the optimal production schedule for combination of open-pit and underground mining strategy due to the issues of complexity and scale of the problem. Newman, Yano, and Rubio (2013) successfully demonstrated how to solve the large longest-path problem by a series of small longest-path problems. The aim of the study is to maximize the NPV which takes the discounted profit from the mined strata (level) less the discounted underground infrastructure cost if strata is extracted through underground mining. From the study conducted by Newman, Yano, and Rubio (2013), a large network formulation that represents the transition problem has been presented in the first place. Due to the complexity of the problem, the authors suggested to decompose the large longest-path problem into a series of smaller networks which take advantages of the underlying composition of the problem. Besides, the authors also placed some rules during the construction of the simpler network to ensure that the series of networks are collective and compressed as much as possible. This approach is flexible as the user can choose to remove any impractical nodes or arcs which reduces the size of the problem. However, the drawback of this approach is the lack of practicality due to the utilization of a level-based concept.

Khaboushan, Osanloo, and Esfahanipour (2020) presented a heuristic-based process that optimizes the NPV of a mining project. The proposed process starts from UPL generation and then, generates a series of transition scenarios within the generated UPL. Underground mining method is assigned for the resource that lies below the UPL. Then, the authors generated a production schedule and maximized NPV for each of the transition scenarios. Finally, the authors compared the results and selected the best scenario which is the scenario which returns the highest NPV.

With the needs of solving the transition problem optimally, this research is committed to construct a robust model to solve the transition problem. The primary objective of this research project is to build an optimization tool that can solve the transition problem by satisfying both open-pit and underground mining constraints. These constraints include, but are not limited to reserve constraints and mining sequence constraints for open-pit mining, mine design restrictions and vertical access limitations for underground mining and crown pillar positioning. Also, the subsidiary outcome of this research is the ability to provide mine planning and optimization guidance for the combination of open-pit and underground mining strategy.

2.4 CLUSTERING TECHNIQUES

In the past, numerous algorithms have been introduced for block aggregation purposes in the mining industry. The clustering technique is an effective way to handle large-scale optimization problems. Clustering is a process in which a partition or aggregation of a set of entities is made into similar groups based on calculated or defined similarity index between each pair of data. The main idea of clustering is to decrease the size of the data which translates to reduction in size of the problem. Although the clustering algorithm generates sub-optimal result for the mine optimization problem, the drastic improvement on computational time for the mathematical model has been proved (Ren and Topal 2014).

The similarity index plays a significant part in the clustering process. It can be used on various properties. Besides, it offers flexibility which is able to be customized as requirements. However, the number of properties involved in the similarity index calculation may exponentially increase the difficulty and complexity of the index. In the mining industry, the most common settings of similarity index are block location, grade, rock type and inter-relationship between clusters (Askari-Nasab et al. 2010).

Hierarchical clustering is one of the most widely used clustering techniques. There are two methods to form cluster trees in a hierarchical clustering algorithm which are agglomerative and divisive (Askari-Nasab et al. 2010). The hierarchical agglomerative clustering algorithm considers each object as a cluster and it starts to aggregate them into a new group. The brief overview of agglomerative clustering algorithm is as below (Jain, Murty, and Flynn 1999; Johnson 1967):

- Step 1: Compute the proximity matrix and treat each entity as a class;
- Step 2: Seek for the most similar pair of the entities and group them into the same group and form the new cluster. Update the proximity matrix;
- Step 3: Stop, if only one cluster left. Otherwise, go to Step 2.

The divisive hierarchical clustering algorithm performs in a top-down fashion which considers the whole set of entities as a single cluster and splits the cluster to form a new group interactively. It stops when the desired number of clusters are reached (Jain, Murty, and Flynn 1999).

Askari-Nasab et al. (2010) presented a hierarchical clustering algorithm for open-pit mines with the aim of diminishing the number of variables of the MILP model. The MILP formulation utilized for production scheduling purposes consists of lower and upper bounds for the grade blending, mining and processing capacity, reserve constraints and precedence relationship rules. The objective function of the MILP model maximized the discounted value (Askari-Nasab et al. 2010; Askari-Nasab, Awuah-Offei, and Eivazy 2010). For the clustering algorithm, four attributes are proposed to be incorporated in the similarity index. The attributes are location, grade, rock type and beneath cluster as described below:

- Location: To avoid impractical block aggregation as opposed to location such as aggregation of blocks which are far apart;
- Grade: To prevent significant grade deviation among blocks within a cluster as uniform grade is considered in the production scheduling optimization;

- Rock type: To differentiate the ultimate destination of the materials; waste rock will be directed to the waste dump and ore will be directed to the processing plant or stockpile;
- Beneath cluster: To form clusters on top of each other which can avoid mining too much low-grade ore or waste material at any time in order to reach high-value clusters.

The proposed calculation of the similarity index for two blocks (block i and j) is as shown in Equation (2-7).

$$S_{ij} = \frac{R_{ij} \times C_{ij}}{\widetilde{D}_{ij}^{WD} \times \widetilde{G}_{ij}^{WG}} \tag{2-7}$$

where,

S_{ij}	Similarity index for blocks
R_{ij}	Rock type similarity factor
C_{ij}	Penalty factor for blocks which are below different clusters
$\widetilde{D}_{ij}^{WD}$	Normalised distance factor
$\widetilde{G}_{ij}^{WG}$	Normalised grade difference factor
WD	Weight of distance factor
WG	Weight of grade factor

The proposed algorithm assumed that each individual block within the pit as is treated as part of a cluster. The most similar and adjacent blocks merge together and form a cluster with a new calculated similarity index. Then, run the algorithm again by selecting the next 'perfect match' blocks. This process repeats until the defined number of clusters is attained. The authors used an adjacency matrix and updated the matrix to accelerate the running time of the algorithm. A case study was also presented by Askari-Nasab et al. (2010) to demonstate use of the algorithm. Although this method has successfully reduced the problem size, intensive processing time is required as it has to run the algorithm at every level in the pit.

Tabesh and Askari-Nasab (2011) introduced a two-stage clustering algorithm to address the optimization of open-pit production scheduling scale issue. The first stage is to adopt the hierarchical algorithm which was proposed by Askari-Nasab et al. (2010). The second stage is to utilize the Tabu Search method to re-evaluate the clusters formed by the first stage. The aim of the second stage is to reduce the binary constraints.

The concept of the second stage is to visit the clusters formed from the first stage and seek for opportunity to detach any of the clusters and attach to the neighbouring cluster. When exploring for opportunity, the main rule is not to break the bond of the current clusters if detachment is exercised. This second stage is advantageous, particularly for those blocks located at the border of the clusters.

Moreover, a 'state measure' is introduced which helps to achieve good and healthy relationship between similarity and arcs of clusters. The state measure is based on average intra-cluster similarity and number of arcs as it is presented in Equation (2-8) as follows:

$$State\ measure = \frac{Normalised\ average\ of\ all\ intracluster\ similarities}{Normalised\ number\ of\ arcs} \qquad (2\text{-}8)$$

The tabu search based clustering scheme is firstly to evaluate the result generated by the hierarchical clustering algorithm and the number of arcs for each entity. Then, it evaluates the relationship between clusters and the immediate clusters beneath it. The process runs iteratively to assess the most dependent block and seek for opportunity to attach to the neighbouring clusters. The case study presented by Askari-Nasab et al. (2010) was used to compare the MILP result. The proposed two-stage clustering method successfully improved the number of coefficient matrix size by 1% and non-zero elements number by 2%. However, it also degraded the MILP result.

Ramazan (2001), Ramazan, Dagdelen, and Johnson (2005) and Ramazan (2007) introduced the fundamental tree algorithm (FTA) to reduce the number of binary integers and constraints within the linear programming model. FTA is a linear programming-based model that aims to aggregate blocks. The conditional properties of FTA are:

- Positive economic value post-cluster;
- Ability to mine the post-cluster without violating slope requirements;
- Cluster cannot be detached after aggregation without compromising the above conditions.

The aggregated blocks are known as a 'fundamental tree'. Prior to FTA, a cone template which represents the wall slope angle requirement is needed in order to evaluate the deposit. The fundamental trees are formed within a pushback. Those trees with a negative value are treated as waste clusters. Besides, the precedence relationships among clusters are determined.

In the FTA, there are five steps to execute which are:

- First step: Seek for a cone value for each pushback or ultimate pit. The economic value of each block is known as a 'cone value' (CV) and is given by Equation (2-9).

$$CV_i = \begin{cases} \textit{Net revenue of i - Mining cost of i - Processing cost of block i; if ore} \\ \textit{- Mining cost of i; if waste} \end{cases} \tag{2-9}$$

- Second step: Assign a coefficient to each ore block to represent its ranking by bench. A ranking system is utilized to perform on-bench based ranking. If two cones with the same cone value exist, a random coefficient will be assigned; hence, no repeated coefficient will be assigned.
- Step three: Setting the mathematical formulation for the FTA and solve mathematical model to generate fundamental trees.
- Step four: If the number of trees generated is greater than the preceding solution, then run the process again. This process will be running iteratively until the number of trees generated is equal to the former result which will then be considered as optimal. The two-dimensional illustration for FTA is presented by Ramazan (2007) and Ramazan, Dagdelen, and Johnson (2005).
- Step five: Develop and solve the MILP prototype for open-pit production scheduling optimization.

Ramazan, Dagdelen, and Johnson (2005) presented a case study on a multi-element copper deposit. In the case study, the authors successfully decreased the number of binary variables by 85%. The result generated was compared to three traditional mine scheduling software. The undiscounted cashflows generated by two of the scheduling tools are higher than the proposed algorithm. However, the total NPV for the proposed algorithm returns the highest.

Mai (2017) and Mai, Topal, and Erten (2018) further developed the FTA into the TopCone Algorithm (TCA) which aggregates blocks into TopCones (TCs). The authors adopted the framework presented by Ramazan (2007) and advanced the algorithm by including the ability to maintain slope shape and able to control the number of TCs. In order to achieve those enhancements, the authors introduced four qualifications that TCs need to satisfy which are: (1) can be unearthed by not violating the slope restrictions, (2) return positive value of TC, (3) satisfy certain constraints such as minimum cone size and (4) TC cannot be fragmented into a smaller size without violating the forementioned qualifications (1-3). As the TCA is able to obey the minimum number of blocks per TC, the framework can be implemented to any real and large-scale problem. The authors implemented the TCA in a block model that contains 1.5 million blocks and compared the result with the Whittle Milawa NPV algorithm. The TCA returned a higher NPV by approximately 7%.

2.5 RESEARCH METHODOLOGY

Throughout the literature review, there is no evident track that any of the available presented methods can solve the transition problem optimally. The main reasons can be traced from the complexity of the transition problem, computational cost and scale of problem. Ultimately, a tool that can deal with both open-pit and underground mine planning and optimization concurrently is required to solve the transition problem.

The proposed research methodology to develop a solution to the problem is as follows:

1. Mathematical modelling method is proposed to solve the transition problem.

- Develop mathematical model to solve *transition point* and generate undiscounted project value for the mine operation – transition model.
- Enhance transition model to determine both *optimal transition point and optimal transition period.* The objective is to maximize the NPV of the project and minimize the capital investment cost for the transition – transition period model.

2. Structured approach is employed to handle the problem scale concern for underground mining.
   - Implement stope-based methodology introduced by Little and Topal (2011) to reduce the problem size.
3. Hierarchical clustering algorithm is selected for open-pit scale reduction purposes.
   - Aggregate open-pit blocks by evaluating the similarity of a group of blocks to reduce the open-pit decision variables.

2.6 SUMMARY

Literature has demonstrated the significance of solving the transition problem and many tactics have been presented to solve the transition problem. However, none of them can generate the optimal solution that fulfills the physical mining constraints in three-dimensional space. Hence, a robust tool that considers both open-pit and underground mining simultaneously is required to solve the transition problem. The tool should aim to define the optimal transition point and/or transition period as required.

According to the concept of optimization for combination of open-pit and underground mining strategy, the problem can be defined as a NP-hard problem due to the complication of the problem nature and the scale of the problem. Therefore, a research methodology is proposed to address the problem which aims to generate the optimal solution.

Chapter 3

MODEL FORMULATION AND VERIFICATION

The objective of this chapter is to demonstrate the mathematical models for the transition problem. It lists the assumptions made while conducting the research and then, the chapter will proceed with introducing the integration of the stope-based methodology involved in this research.

Section 3.3 presents the Transition Point model that defines an optimal transition point for the combination of open-pit and underground mining strategy;

Section 3.4 introduces the Transition Period model that identifies the optimal transition point, optimal transition period, optimal mining strategy and optimal mine schedule for the combination of open-pit and underground mining strategy;

Section 3.5 shows an implementation on a two-dimensional case study to show the validity of the proposed models.

3.1 ASSUMPTIONS

1. All mining blocks in the three-dimensional block model are regular, i.e., same size in x, y and z directions.
2. The economic block value of each mining block in the block model is known and constant.
3. The pit-wall slope requirement is to avoid the geotechnical risks such as pit-wall failure. A conventional 45 degree pit slope is considered in this research. Hence, to satisfy the wall slope requirement, five blocks are needed to be extracted in order to gain access of the underlying target block.
4. For the underground mining method, this research considers sublevel stoping method. The underground sublevel stoping mining method has been employed by many operations in Australia for its numerous advantages. The advantages of sublevel stoping method includes the high ore recovery rate, lower cost in a large-scale production, and high productivity.
5. The models allow for non-simultaneous open-pit and underground mining operations.
6. The NPV is calculated based on pre-tax and depreciation assumptions.
7. All values are in $AUD currency.
8. No ore stockpiling is included in the models.

3.2 DATA PREPARATION

3.2.1 Block model

The orebody block model is the basic geological input to the transition problem. A three-dimensional block model contains thousands of mining blocks. Each of the mining block consists of numerous attributes such as location, quality (grade) and quantity (tonnage). Density is another mining block attribute. This attribute is often used to differentiate the type of the rock and its grade. With the attributes in each block, the dimension of each block, tonnage value and metal grade of each block are distinct (Grobler 2015).

Furthermore, the geological block model is transformed into an economic block model by including economic parameters and operation-related parameters. The simplified formula used to determine the block value for each block in the economic block model is shown in Equation (3-1).

$$v = (p - r)gy - c - m \tag{3-1}$$

where
p = commodity price ($/unit of ore)
r = refining cost ($/unit of ore)
g = grade
y = recovery rate
c = processing cost ($/tonne of ore)
m = mining cost ($/tonne of ore)

3.2.2 Stope-based methodology for underground mining

Stope-based modelling for underground mining was introduced by Little and Topal (2011). It is an inventive way to reduce the number of binary variables in the mathematical model. Due to the number of binary variables (from both open-pit and underground) involved in the MILP model for the transition problem, stope-based modelling is adopted to decrease the number of binary variables for underground mining. The concept of the approach is to use 2x2x2 stope design, to combine eight blocks into one stope. Thus, only one binary variable is assigned for each stope instead of eight binary variables for each block. The naming convention for the stope is represented as X (the coordinate of the first and last block). By using the example in the Figure 3-1, the stope is referred to as X (1,1,1)/ (2,2,2).

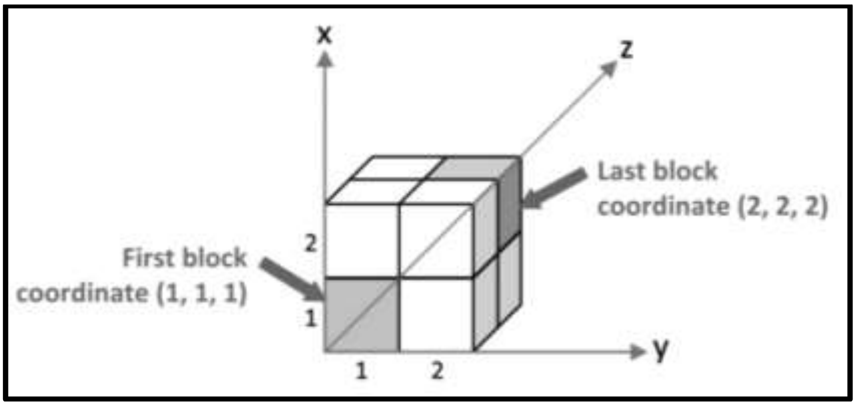

Figure 3-1: Stope based methodology naming convention (Little and Topal 2011)

Moreover, to further reduce the binary variables for underground mining, a pre-processing step is taken. The aim of this process is to predetermine the profitable stopes. First and foremost, an envelope of a stope such as 2x2x2 stope envelope, is used to go across the underground block model. This step is implemented to find all the potential stopes within the underground block model and determine its associated economic value. Then, only stopes with positive values are retained while those with negative values are removed. As a result, a list of profitable stopes is generated after the process completed. This approach aligns with the precedence concept presented by Ramazan and Dimitrakopoulos (2004), which considers the waste blocks as air blocks to obtain fewer binary variables. By employing this approach, Little, Knights, and Topal (2013) successfully improved the solution time of the problem.

3.3 TRANSITION POINT MODEL – OPTIMIZATION MODEL 1

Transition point model is developed to solve the transition problem by providing the optimal transition point by maximizing the undiscounted profit of the mine.

3.3.1 Notations and Variables

Indices

$i, \acute{i}$ = index for blocks in open-pit mining

$j, \acute{j}$ = index for stopes in underground mining

$k, \acute{k}$ = index for mining level

m = index for mining method; =1 for open-pit mining and 2 for underground mining

op = open-pit

ug = underground

Sets

K = set of levels in the orebody model

μ_i = set of overlying or precedence blocks for block i

B_j = set of all the stopes that share mutual blocks with stope j

$L_{op,k}$ = set of all open-pit blocks on level k

$L_{ug,k}$ = set of all underground stopes on level k

Parameters

C_i = the discounted profit to be generated by mining block i

S_j = the discounted profit to be generated by mining stope j

ℓ = number of rows that should remain as a crown pillar

A = total number of overlying blocks that need to be mined in order to extract ore block i

Decision variables

$$x_i = \begin{cases} 1, \text{if block i is mined by open} - \text{pit mining} \\ 0, \text{otherwise} \end{cases}$$

$$y_j = \begin{cases} 1, \text{if stope j is mined by underground mining} \\ 0, \text{otherwise} \end{cases}$$

$$T_{k,m} = \begin{cases} 1, \text{if level k is mined by mining method m} \\ 0, \text{otherwise} \end{cases}$$

$$H_k = \begin{cases} 1, \text{if level k is left as a crown pillar} \\ 0, \text{otherwise} \end{cases}$$

3.3.2 Transition Point Model Formulation

The mathematical formulation is as follows:

$$Max\ Z = \sum_{i \in M_i} C_i x_i + \sum_{j \in N_j} S_j y_j \tag{3-2}$$

Subject to

$$A \cdot x_i - \textstyle\sum_{\acute{i}} x_{\acute{i}} \leq 0 \qquad \forall\, i, \acute{i} \in \mu_i \tag{3-3}$$

$$y_j + y_{\acute{j}} \leq 1 \qquad \forall\, j, \acute{j} \in B_j \tag{3-4}$$

$$T_{k,1} - x_{\ddot{i}} \geq 0 \qquad \forall \ddot{i} \in L_{op,k} \tag{3-5}$$

$$T_{k,2} - y_j \geq 0 \qquad \forall \ddot{i} \in L_{ug,k} \tag{3-6}$$

$$\textstyle\sum_{m=1}^{2} T_{k,m} + H_k \leq 1 \qquad \forall\, k \tag{3-7}$$

$$\ell \cdot T_{k,1} + \textstyle\sum_{\acute{k}=0}^{k} H_{1+\acute{k}} \geq \ell \qquad \forall\, k \tag{3-8}$$

$$\ell \cdot T_{\ell+1,2} - \textstyle\sum_{\acute{k}=1}^{\ell} H_{\acute{k}} \leq 0 \tag{3-9}$$

$$x_i, y_j, T_{k,m}, H_k \in \{0, 1\} \qquad \forall\, ijkm \tag{3-10}$$

This model has an objective function to maximize the undiscounted value of the mine project from both open-pit and underground mine operations as shown in the Equation (3-2).

Constraint (3-3) is established to maintain a stable pit-wall slope for geotechnical safety purposes as well as to satisfy the precedence relationship. It makes sure that all the overlying blocks above a given block are removed prior to mining the block. The constant, A, is used to represent the number of blocks required to be extracted to gain access to a given block.

Constraint (3-4) makes sure that there are no overlapping stopes in the ultimate stope layout. Hence, with all the possible stope layouts which share one or more common blocks, only one of them can be removed and become part of the final underground mining layout. The aim is to hold the practicality of the mining strategy such that none of the stopes or blocks are being evaluated twice in the outcome.

Constraints (3-5) and (3-6) ensure that only one mining method can be selected to mine each level. The structure of these constraints is such that one stope or one block of a given level is mined through a selected mining method, the entire level is then considered to be mined by the same mining method. For instance, in Figure 3-2, Block 5 located at level 2 is extracted by open-pit mining, the whole level can only be mined through open-pit mining and vice versa. Additionally, the level can be left as a crown pillar if necessary.

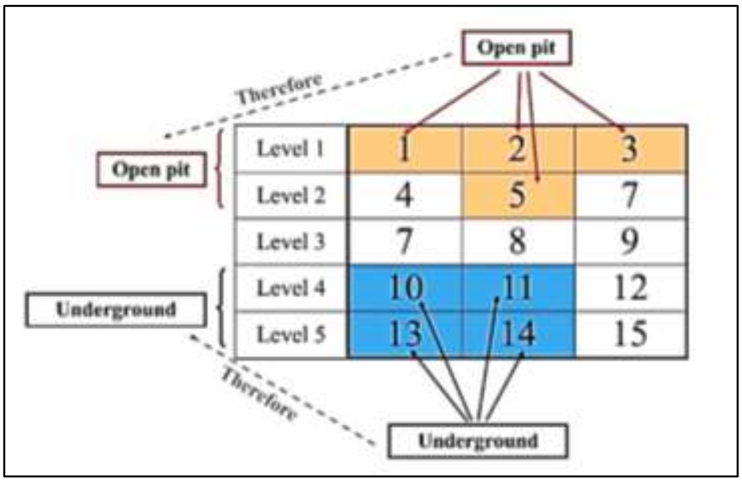

Figure 3-2: Equation 3.4 to Equation 3.6 - Only one mining method for each level

Constraint (3-7) and Constraint (3-8) ensure that a certain thickness of the strata is required to be retained as a pillar between open-pit and underground mining operations. It guarantees that the crown pillar is positioned between open-pit and underground working areas. The number of levels required to be retained is influenced by the geotechnical conditions and structures of the deposit. A crown pillar is important for the combination of open-pit and underground mining strategy. It is employed to control the interaction between the surface and underground mine operations. It also provides geotechnical stability and prevents some operational issues such as inrush of water into the underground working area. Thus, a crown pillar helps to reduce, if not eliminate, the geotechnical and operational problems. A thicker pillar is required for local rock with poor strength to avoid the subsidence of the surface.

Constraint (3-9) makes sure that if only underground mining method is the most profitable mining strategy for the project, the pillar requirement is still satisfied. This constraint has elevated the model by not only solving the transition problem, but also providing the guidance toward mining method selection process. The variables present in the model need to be non-negative and integer.

Constraint (3-10) ensures the nature of the decision variables required.

3.4 TRANSITION PERIOD MODEL – OPTIMIZATION MODEL 2

Transition Point model (Optimization Model 1) aims to generate the optimal point and mining layout for the combination of open-pit and underground mining strategy. However, the second variant model, which is called the Transition Period model, aims to generate the transition period and mine schedule along with optimal transition point and mining layout.

3.4.1 Notations and Variables

Indices

$i, \acute{i}$ = index for blocks in open-pit mining

$j, \acute{j}, \grave{j}, \ddot{j}, \dddot{j}, \tilde{j}$ = index for stopes in underground mining

$k, \acute{k}$ = index for mining level

$t, \acute{t}$ = index for time periods within scheduling horizon

m = index for mining methods, where $m = 1$ for open-pit mining and $m = 2$ for underground mining.

op = open-pit

ug = underground

Sets

I = set of blocks for open-pit mining

J = set of stopes for underground mining

K = set of levels in the orebody model

μ_i = set of overlying or precedence blocks for block i

α_j = set of stopes that share common blocks with stope j

β_j = set of horizontally adjacent stopes to stope j

γ_j = set of vertically stopes adjacent stopes to stope j

δ_j = set of overlying stopes over stope j

τ_j = set of stopes that do not share same extraction level with stope j

ξ_k = set of the levels that is located above level k

ν_k = set of the level that is located immediate above level k

T = set of scheduling periods

Parameters

A = total number of overlying blocks that need to be extracted to mine ore block i

B_i^t = the discounted value of block i in period or year t

S_j^t = the discounted value of stope j in period or year t

d^t = the discounted development cost in period t

tl = time lag between commencement of underground development and underground mining

H_k = total number of levels above level k

R = total number of levels in the orebody model

C = total number of levels to be retained as a crown pillar

g_i = grade or metal content of material in block i

$\bar{g}_j$ = average grade of material in stope j

q_i = quantity of material in block i

$\bar{q}_i$ = quantity of ore extracted from block i

$\underline{q}_j$ = quantity of ore extracted from stope j

N_i = tonnage of block i

N_j = tonnage of stope j

O_i, O_j = quantity of ore extracted from block i and stope j

M_{op}, M_{ug} = mining capacity for open-pit and underground operation per period, respectively

P_t = average processing capacity per period t over the planning horizon

D_t = average development capacity per period t for underground operation, i.e. if operation can develop 2 levels per period, then $D = 2$

$\overline{G}, \underline{G}$ = upper and lower bound on required head grade in the mine operation

Decision variables

$$x_i^t = \begin{cases} 1; & \text{if block } i \text{ is mined in period or year } t \text{ by OP mining} \\ 0; & \text{otherwise} \end{cases}; \quad \text{i.e. } x_i^t \in [0,1]$$

$$y_j^t = \begin{cases} 1; & \text{if stope } j \text{ is mined in period or year } t \text{ by UG mining} \\ 0; & \text{otherwise} \end{cases}; \quad \text{i.e. } y_j^t \in [0,1]$$

$$e_{km}^t = \begin{cases} 1; & \text{if level } k \text{ is mined with method } m \text{ in year } t; m = \begin{cases} 1 \text{ for OP mining} \\ 2 \text{ for UG mining} \end{cases}; \\ 0; & \text{otherwise} \end{cases}$$

$$L_k = \begin{cases} 1; & \text{if level } k \text{ is in crown pillar} \\ 0; & \text{otherwise} \end{cases}; \text{ i.e. } L_k \in [0,1]$$

$$a^t = \begin{cases} 1; & \text{if UG development commences in period } t \\ 0; & \text{otherwise} \end{cases}; \quad \text{i.e. } a^t \in [0,1]$$

3.4.2 Transition Period Model Formulation

The mathematical formulation to determine the optimal transition point, optimal transition point and optimal scheduling for combination of open-pit and underground mining strategy is as follows:

$$\max z = \sum_{t\in T}\left[\sum_{i\in I} B_i^t\, x_i^t + \sum_{j\in J} S_j^t\, y_j^t - D^t a^t\right] \tag{3-11}$$

subject to

$$A \cdot x_i^t - \sum_{i\in\mu_i}\sum_{t=1}^{t} x_i^t \le 0;\ \forall it \tag{3-12}$$

$$\sum_{t\in T} x_i^t \le 1;\ \forall i \tag{3-13}$$

$$\sum_t y_j^t + \sum_t y_{\acute{j}}^t \le 1;\ \forall j, \acute{j} \in \alpha_j, t \in T \tag{3-14}$$

$$y_j^t + \sum_{\ddot{j}} y_{\ddot{j}}^t \le 1;\ \forall jt, \ddot{j} \in \beta_j \tag{3-15}$$

$$y_j^t + \sum_{\dddot{j}} y_{\dddot{j}}^t \le 1;\ \forall jt, \dddot{j} \in \delta_j \tag{3-16}$$

$$\sum_t y_j^t + \sum_t \sum_{\grave{j}} y_{\grave{j}}^t \le 1;\ \forall j, \grave{j} \in \gamma_j\,, t \in T \tag{3-17}$$

$$\sum_t y_j^t + \sum_{\tilde{j}} y_{\tilde{j}}^t \le 1;\ \forall j, \tilde{j} \in \tau_j, t \in T \tag{3-18}$$

$$e_{k1}^t - x_i^t \ge 0;\ \forall ikt \tag{3-19}$$

$$e_{k1}^t - y_j^t \ge 0;\ \forall jkt \tag{3-20}$$

$$e_{k1}^t + e_{k2}^t + L_k \le 1;\ \forall kt \tag{3-21}$$

$$\sum_t C \cdot e_{k1}^t + L_{k-1}\ \ge C;\ \forall k, t \in T \tag{3-22}$$

$$\sum_k L_k \ge C;\ \forall\, k \in K \tag{3-23}$$

$$\sum_{\acute{k}} \sum_{\acute{t}=1}^{t} e_{\acute{k}1}^{\acute{t}} - \mathrm{H}_k t e_{k1}^t \ge 0;\ \forall tk, \acute{k} \in \xi_k \tag{3-24}$$

$$e_{k2}^t - e_{k2}^{t+1} \le 0;\ \forall kt \tag{3-25}$$

$$e^t_{\nu_k 2} - e^t_{k2} - L_k \leq 0;\ \forall kt \qquad (3\text{-}26)$$

$$\sum_{t=1}^{T-tl} a^t \leq 1;\ \forall t \qquad (3\text{-}27)$$

$$\sum_k e^{t+dl}_{k2} - \sum_{\dot{t}=1}^{t} D_t \dot{t} a^{\dot{t}} \leq 0;\ \forall t, k \in K \qquad (3\text{-}28)$$

$$\sum_i x^t_i N_i \leq M_{op};\ \forall t, i \in I \qquad (3\text{-}29)$$

$$\sum_{j \in J} y^t_j N_j \leq M_{ug};\ \forall t, j \in J \qquad (3\text{-}30)$$

$$\sum_{i \in I} x^t_i O_i \leq P_t;\ \forall t, i \in I \qquad (3\text{-}31)$$

$$\sum_{i \in I} x^t_i O_i \left(g_i - \overline{G}\right) + \sum_{j \in J} y^t_j O_j \left(\bar{g}_j - \overline{G}\right) \leq 0;\ \forall t, i \in I, j \in J \qquad (3\text{-}32)$$

$$\sum_{i \in I} x^t_i O_i \left(g_i - \underline{G}\right) + \sum_{j \in J} y^t_j O_j \left(\bar{g}_j - \underline{G}\right) \geq 0;\ \forall t, i \in I, j \in J \qquad (3\text{-}33)$$

$$x^t_i, y^t_j, e^t_{km}, L_k, a^t \in \{0, 1\};\ \forall t \qquad (3\text{-}34)$$

Objective function (3-11) aims to maximize the NPV of the mining project based on the three main components. In the objective function (3-11), the first two of the three components are the discounted block economic value for open-pit and underground mining, respectively. The last component included in the objective function is the decisive factor for pre-development capital investment. During the planning stage for the transition from open-pit to underground, pre-production capital investment is often ignored in the decision-making process. In fact, the capital investment may affect the decision of going underground. Thus, the proposed model includes the required pre-production capital investment as a lump sum dollar value. This will provide a more accurate and comprehensive result and guidance.

Constraint (3-12) satisfies the precedence pit slope requirements. The constraint is to ensure that, to mine an underlying block, all immediate predecessors are to be mined to successfully retain a required slope angle. In Figure 3-3, for example, all precedence blocks of Block 5 are extracted prior to or at the same time as Block 5 (T_5). This is to ensure the accessibility of Block 5 while it needs to be mined.

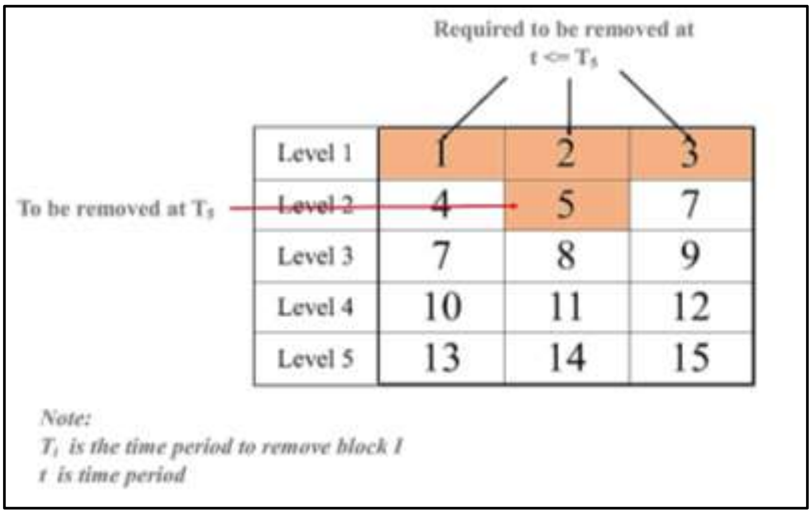

Figure 3-3: Precedence relationship

Constraint (3-13) - (3-14) enforce to oblige the reserve constraint for open-pit and underground mining. Constraint (3-13) guarantees that any blocks mined by open-pit mining can be mined only once in any period. Constraint (3-14) not only ensures that each stope can only be extracted only once in any time along the life-of-mine, but also prevents the overlapping stope formation in underground mining.

Constraint (3-15) satisfies the horizontal stope adjacency requirement. It ensures while mining a given stope, those adjacent stopes are not sequenced at the same period. Constraint (3-16) ensures the vertical adjacency constraints are satisfied, referring to the stopes located directly above or below a given stope are not mined at the same time. These settings are significant to prevent over-scheduling of mining activities (such as bogging, drilling, and firing stopes) in an area. Overly intensified mining schedule in an area within one schedule period may lead to massive voids and exhaustive interactions.

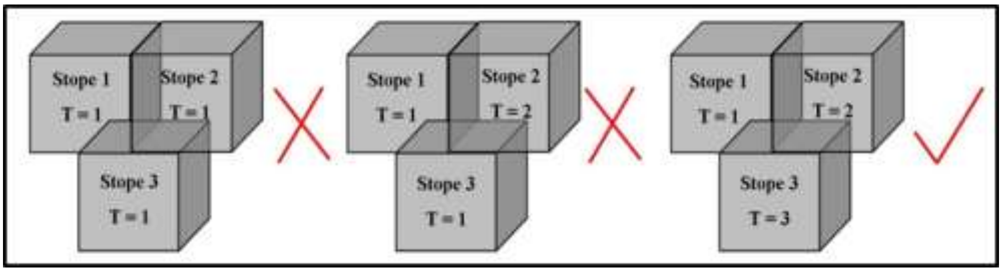

Figure 3-4: Non-concurrent adjacent stope production sequence example (Little and Topal 2011)

Constraint (3-17) structures the offsetting of the stopes vertically to eliminate the creation of plane of weakness. There are operational and geotechnical risks associated with stopes located directly above each other due to the plane of weakness as shown in Figure 3-5. Thus, it is important to consider the plane of weakness in the stope formation and optimization processes.

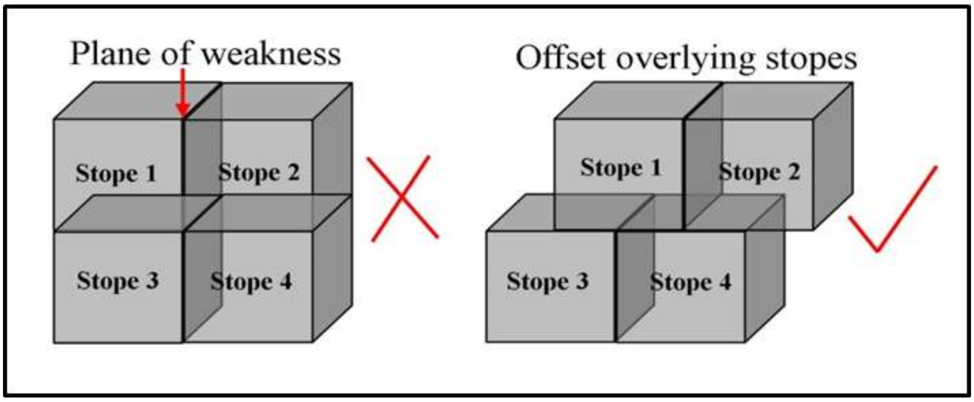

Figure 3-5: Plane of weakness occurrence (Little and Topal 2011)

Constraint (3-18) ensures the availability of an appropriate number of production levels. In underground mining, extraction level and production level are where most of the on-going capital development cost is incurred. Therefore, it is significant to optimize the extraction and production levels to minimize the cost and maximize practicality.

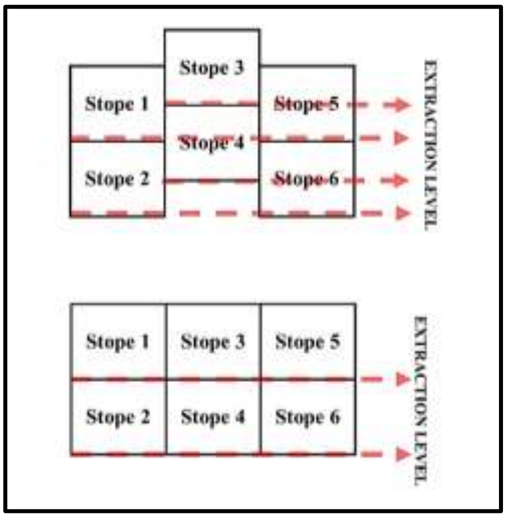

Figure 3-6: Same extraction levels concept (Little and Topal 2011)

Constraints (3-19) - (3-21) impose restrictions on mining strategy selection on each level. Hence, each level only can be mined by open-pit mining, underground mining or remains as a crown pillar. These constraints are sharing a common purposes as constraints (3-5) - (3-6).

Constraints (3-22) and (3-23) are formulated for crown pillar design requirements. Constraint (3-22) ensures that the crown pillar is located just beneath the final pit-shell. Constraint (3-23) maintains the required thickness of the crown pillar for the stability of the underground working area. When considering the thickness of crown pillar, it is important to take into account the factors that may affect the integrity of the pillar such as natural pillar deterioration, the water level in the pit bottom and ground support requirement for the development level closest to the pillar.

Constraint (3-24) enforces the level-based and top-down dependency for the open-pit. For example, if a block in a given level needs to be mined, the overlying levels have to be mined prior to or at the same time as the given level.

Constraint (3-25) satisfies the accessibility of a level in underground mining. In underground mining, once a level is accessible in a period of time, the entire level remains accessible until the end of LoM. Constraint (3-25) is designed to contain this nature. Constraint (3-26) is structured to flex the underground formation process of the production level as each level can be mined by underground mining or remained unmined.

Constraints (3-27) - (3-28) are designed for the sequence and dependency for underground development. Constraint (3-27) restricts underground development in which a^t is only initiate once along the LoM. Moreover, in underground mining, the development capacity will determine the number of additional accessible production levels in each period. For instance, with the available resources, two additional levels could be available to access per period. Hence, constraint (3-28) is used to handle the additional developed level per period.

Constraints (3-29) - (3-30) are formulated to maintain upper bound mining capacity of both open-pit and underground mining. Constraint (3-30) is known as underground ore handling capacity which is referred to as the capacity of underground production fleet. Due to the different nature of underground mine operations, underground development capacity is accounted separately from the production fleet.

Constraint (3-31) satisfies the mill capacity. In open-pit mining, the material movement constitutes ore and waste. The destination of the hauled material is either the processing plant or waste dump. On the other hand, in underground mining, only ore will be hauled from a production level to the surface. As a result, typically, underground mining capacity is equal to processing capacity.

Constraint (3-32) and (3-33) are structured to reduce grade fluctuations and to optimize plant operation efficiency. The consistency of head grade is important as it will directly impact the plant recovery; fluctuation in head grade will lead to a poor plant performance. Hence, upper and lower bounds of ore feed grade are used to maintain consistency in the head grade.

Constraint (3-34) retains the non-negativity and integrality of the variables as appropriate.

3.5 VERIFICATION – TWO-DIMENSIONAL CASE STUDY

The optimization models discussed above were programmed in Microsoft Visual Studio VB.NET (VB.NET 2015) and the mathematical models were solved using the IBM CPLEX Solver (IBM CPLEX 2013). To examine the functionality of the models, the models were tested by using a two-dimensional data set. This test helped to demonstrate how the models work. This section gives the details on the verification process which includes the introduction of the hypothetical data set, solution interpretation and the discussion of the result.

A two-dimensional hypothetical data set with 204 blocks was generated to demonstrate the validity of the models. The hypothetical deposit with a block size of 20 x 20m and a stope size of 2 x 2 were created. A crown pillar with a minimum thickness of 40m (two levels) is required to guarantee geotechnical stability. The block economic values of both open-pit and underground are calculated and shown in Figure 3-7 and Figure 3-8, respectively.

1	1	1	1	1	4	4	-1	2	4	4	4	-2	-2	-2	-2	-2
-3	-3	-3	1	2	1	0	3	5	5	0	-1	-2	-3	-3	-3	-3
-4	-3	-4	-4	-3	1	3	5	6	5	1	-1	-3	-3	-4	-4	-3
-4	-3	-4	-4	-3	-1	9	8	5	3	8	-2	5	-4	-4	-4	-4
-5	-5	-5	-5	-5	0	1	1	2	2	-1	8	-4	-5	-5	-5	-5
-5	0	-5	-5	0	11	7	5	3	5	6	-5	-5	-5	-5	-5	-5
-6	-2	-6	-6	-2	-1	2	2	2	4	4	-6	-6	-6	-6	-6	-6
-8	-4	-8	-8	1	1	1	2	2	5	4	-8	-8	-8	-8	-8	-8
-9	-6	-9	-9	-6	1	1	-9	-9	-9	-9	-9	-9	-9	-9	-9	-9
-9	-6	-9	-9	-6	-3	-6	-9	-9	-9	-9	-9	-9	-9	-9	-9	-9
-9	-6	-9	-9	-6	-3	-6	-9	-9	-9	-9	-9	-9	-9	-9	-9	-9
-9	-6	-9	-9	-6	-3	-6	-9	-9	-9	-9	-9	-9	-9	-9	-9	-9

Figure 3-7: Block economic model for open-pit mining

-1	-1	-1	-1	-1	-2	-1	1	1	2	0	0	-1	-1	-1	-1	-1
-1	-1	-1	-1	0	1	1	0	1	-1	-1	-1	-1	-1	-1	-1	-1
-2	-2	-2	-2	-2	-2	-1	1	0	2	1	-1	-2	-2	-2	-2	-2
-2	-2	-2	-2	-2	-1	2	1	0	-1	0	-1	-2	-2	-2	-2	-2
-2	-2	-2	-2	-2	-1	1	2	0	2	0	-1	-2	-2	-2	-2	-2
-2	-2	-2	-2	-2	-1	4	2	2	3	-1	-2	-2	-2	-2	-2	-2
-3	-3	-3	-3	1	-2	2	5	2	2	3	-3	-3	-3	-3	-3	-3
-3	-3	-3	-3	-2	-1	1	2	3	12	4	-3	-3	-3	-3	-3	-3
-3	-3	-3	-3	-2	-1	1	12	4	-3	-3	-3	-3	-3	-3	-3	-3
-3	-3	5	-3	5	5	2	12	7	-3	-3	-3	-3	-3	-3	-3	-3
-3	-3	-1	5	-3	5	5	6	7	-3	-3	-3	-3	-3	-3	-3	-3
-3	-3	-1	-3	5	5	2	6	1	-3	-3	-3	-3	-3	-3	-3	-3

Figure 3-8: Block economic model for underground mining

Numerous mining strategies have been considered to demonstrate the validity of the Transition Point model (Optimization Model 1) and Transition Period model (Optimization Model 2) proposed in this research. The strategies considered include: (i) open-pit mining only, (ii) underground mining only, (iii) conventional transition approach, (iv) proposed Transition Point model and (iv) proposed Transition Period model. Strategies (i) and (ii) are using the two proposed models in the research by relaxing the inputs and constraints in the model. For instance, using Transition Point Model for strategy (i), the underground constraints and inputs have been relaxed to considered only open-pit mining; vice versa. The information and result of each of the considered strategy is presented in the Table 3-1.

In comparison with the single mining method approach (open-pit or underground), the combination of open-pit and underground mining strategy approaches generated better value. The table has shown that the proposed Transition Point model generates the highest return which is $207 whereas open-pit mining and underground mining only generate $111 and $139, respectively. The conventional transition approach, however, returns a value of $152. The mining layouts for open-pit mining only and underground mining only are demonstrated in Figure 3-11.

As shown in Figure 3-9, the Transition Point model indicates that levels 1 to 4 should be mined using open-pit mining, retained the two levels below the pit as a crown pillar and extracted the remaining levels using underground mining. Therefore, the optimal transition point is 80m. However, the conventional transition approach only employs underground mining after the UPL is reached, as shown in Figure 3-12. The conventional transition approach only makes the transition to underground mining 200m below the surface (transition point is 160m). From the difference of the values generated by the proposed Transition Point model and conventional transition approach, it is fair to conclude that the conventional transition approach has an over-mined final pit which leads to the loss of value for the transition problem - combination of open-pit and underground mining strategy.

Table 3-1: The comparison of the result generated by each possible strategy

Scenario / Mining Strategy	Revenue
Open-pit mining only	$111
Underground mining only	$139
Conventional transition approach	$152
Transition Point Model (Optimization Model 1)	$207

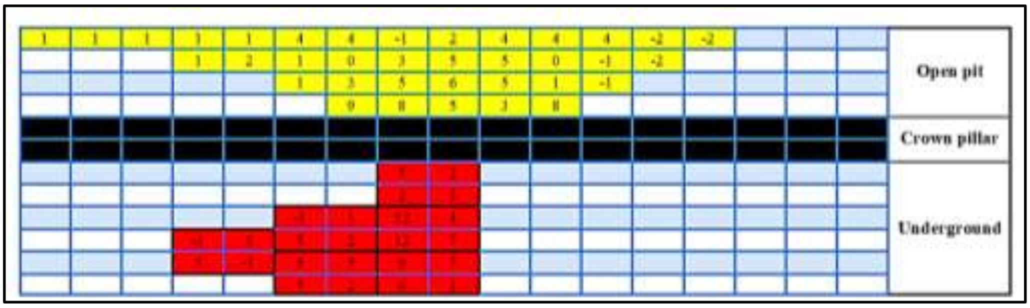

Figure 3-9: Mining layout generated by the Transition Point Model

Furthermore, the Transition Period model (Optimization Model 2) in this research aims to obtain the optimal transition point, optimal transition period and mine schedule simultaneously. Figure 3-10 demonstrates the mining layout for the proposed Transition Period model which generated an NPV of $169.8. As the Transition Period model considers elements such as mining capacities, grade profile, underground mining sequence practicality and underground development constraints, the final pit layout generated by the Transition Period model is smaller than the final pit layout of the Transition Point model. The main reason could be observed from the resource distribution along the schedule periods. The proposed Transition Period model is used to obtain the best resource allocation for the operation along the schedule period by applying a discount factor, whereas the Transition Point model only considered the economic block value of the block model. Therefore, if the resource can generate better value by making the transition to underground mining, the open-pit operation can be ceased. By comparing the results generated by Transition Point model and Transition Period model, optimal transition point, it is evident that the discount factor and mine schedule will directly impact the optimal transition point.

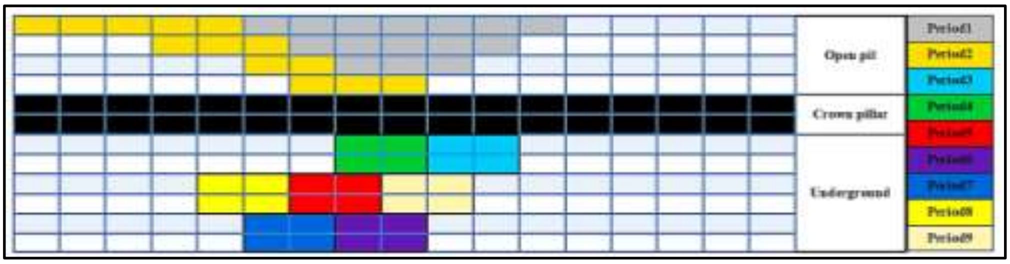

Figure 3-10: Mining layout generated by the Transition Period Model

Figure 3-11: Mining layout for open-pit only and underground only

Figure 3-12: Mining layout for the conventional transition approach

3.6 SUMMARY

In summary, mathematical models were developed to solve the transition problem. Two mathematical models are developed. The Transition Point model aims to search for the mining layout for the combination of open-pit and underground mining strategy. This model is formulated to achieve the maximized undiscounted project value while satisfying a range of restrictions. The constraints considered in the model are open-pit slope constraints, underground mine design constraints and reserve constraints. At the same time, crown pillar placement is also described in the model to ensure that a crown pillar is positioned at the level that returns the least value.

Moreover, the Transition Period model is established as a result of taking further views about 'when to make the transition' and the cost of development. Hence, the second mathematical model aims to obtain an optimal transition point and optimal transition period while the mine schedule is developed with the objective of maximizing the NPV and minimizing capital costs incurred for making the transition from open-pit to underground. The constraint settings of the models are included in the open-pit mining sequence, underground mining sequence, development rate, crown pillar and reserve constraints.

Lastly, a two-dimensional case study was presented to validate the legitimacy of the proposed models. The results were verified and showed that the constraints structured in the models are satisfied. Besides, the results indicated that the proposed models return higher values than any of the single mining method (open-pit or underground mining) and conventional transition approach.

Chapter 4

IMPLEMENTATION – EXACT METHODS

The objective of this chapter is to demonstrate the implementation of the proposed models in a three-dimensional case study. The chapter gives a brief overview of the case study and two sections as follows:

Section 4.2 demonstrates the implementation and results.

Section 4.3 discusses the performance of the models and challenges.

4.1 BACKGROUND

A gold deposit is used for implementation purposes. The block model was obtained from the training dataset used by Mining Education Australia (MEA). The original dataset consists of 83,025 blocks. Using this block model with the proposed transition models in order to solve the transition problem is beyond the computational capacity of a standard computer. Hence, the block model has been reblocked into a block size of 45m x 45m x 45m which consists of 7,200 blocks with an average gold (Au) grade of 2.25 g/t. Figure 4-1 demonstrates the grade distribution of the block model.

For underground mining, each stope is designed to be mined with the size of 2x2x2 blocks totaling of 8 blocks. The parameters used as inputs for the optimization models are shown in Table 4-1.

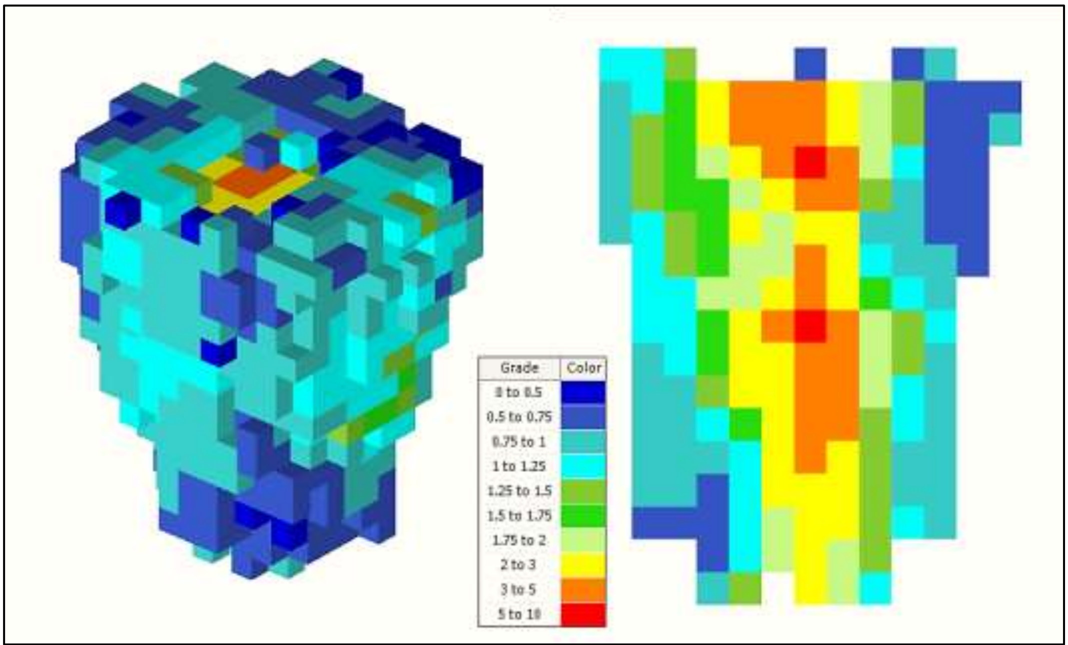

Figure 4-1: Resource distribution for the hypothetical gold deposit block model

Table 4-1: Economic and operational parameters

Open-pit cut-off grade	1 g/t
Underground cut-off grade	2 g/t
Recovery	90%
Discount Rate	12%
Crown pillar	2 Levels
Underground development rate	4 levels per period
Mining capacity for open-pit – Maximum	32,000,000 tonnes per schedule period
Mining capacity for underground – Maximum	7,424,000 tonnes per schedule period
Blocks per stope	8 blocks
Precedence blocks for open-pit	5 blocks
Time period	10
Slope angle	45 degrees

4.2 IMPLEMENTATION AND ANALYSIS

4.2.1 Pre-processing steps

Commonly, to solve the transition problem, an economic block model for both open-pit mining and underground mining (two block models) are imported into optimization process. As proposed in Section 3.2, only profitable stopes are qualified and included in the optimization process to reduce the problem size. The steps to generate the profitable stopes are as below:

1. The economic block model for the underground block model is generated. Then, blocks in the economic block model are aggregated using the stope profile of 2x2x2 blocks. As a result, all possible stopes are identified.
2. The next step is to determine the qualified stopes. Hence, within the possible stopes pool, the stopes with positive values are selected as the qualified stopes. All the negative value stopes are eliminated.

The data pre-processing took approximately 45 minutes to generate 326 qualified stopes obtained. As a result, the stope-based methodology successfully reduced the binary variables for underground mining to 326 stopes. Without the process of identifying profitable stopes, the model must consider 7,200 blocks which can create millions of combinations with a stope profile of 2x2x2. Hence, with the step of processing, the number of variables required for underground mining has reduced significantly. Then, the qualified stopes determined were substituted into the proposed mathematical models.

Moreover, the size of the problem for proposed Transition Period model increases exponentially as time is considered in the optimization model. Thus, to further reduce the problem size of Transition Period model, the UPL is determined prior to the optimization process. UPL is the largest pit where open-pit can mine and it generates the highest undiscounted profit returns to the mine operation. Therefore, while taking underground mining into account, the final pit of the combination of open-pit and underground mining strategy can be significantly smaller than the UPL (Fuentes 2004). Therefore, using the UPL for the Transition Period model will not violate the optimality of the model. As a result of the UPL definition, the variables for open-pit mining are reduced to 1,366 blocks. The generated UPL is shown in Figure 4-2. The UPL contains 1,394 blocks which generates the undiscounted cashflow of $2.362 billion.

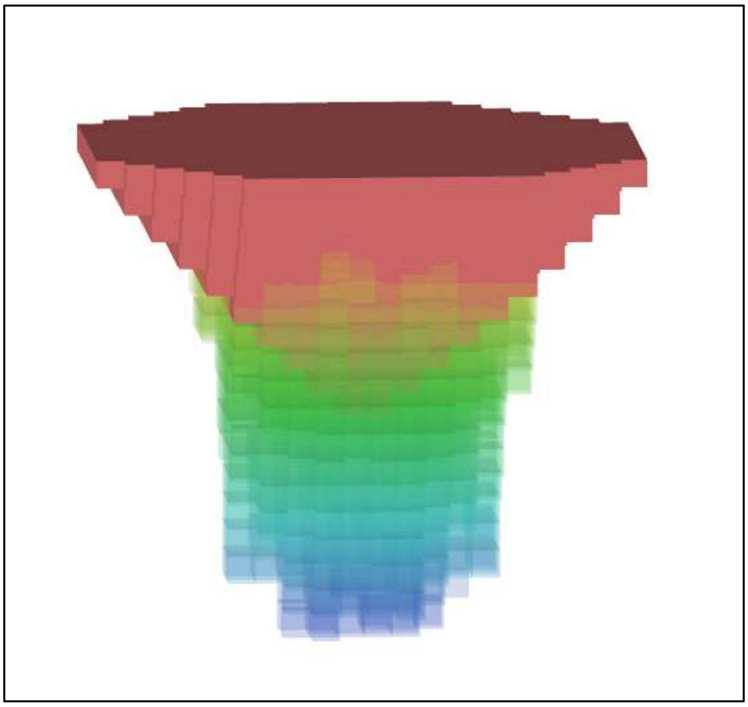

Figure 4-2: UPL generated to reduce problem size for the Transition Period model

The two proposed models resulted in two mathematical problems. Each mathematical problem involves thousands of variables. A standard computer with a specification of 2.8 GHz CPU and 16 GB RAM was used to solve the mathematical models. Based on experience, improvement in computational gap of

1% will take a long time. However, it may not see a significant improvement in the generated result in comparison to the optimal result. Thus, an appropriate gap can be utilized to shorten the computation time. In this research, a gap of 5% was used.

4.2.2 Models implementation

The result of applying the Transition Point model is presented in Figure 4-3. The optimal solution recommends the first six levels (240m) to be mined by open-pit mining. Then, the Level 7 and 8 are left as a crown pillar. Underground mining is recommended to extract the remaining ore underneath the crown pillar. The undiscounted value generated by Transition Point model is $3.740 billion which is higher than the UPL result ($2.362 billion). Hence, the optimal transition point is at level six with the transition depth of 240m.

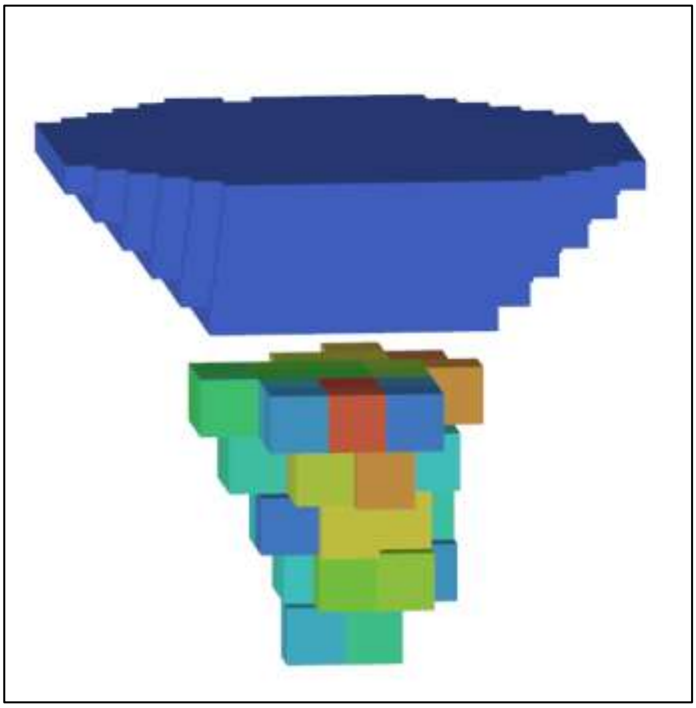

Figure 4-3: Optimal mining layout of the Transition Point model

For the Transition Period model implementation, two scenarios are presented. The first scenario considers no delay in the production during the transition from open-pit to underground, meanwhile, the second scenario considers two periods of delay in the production. In some cases, the mine operation is unable to make the smooth transition from open-pit to underground mining for various reasons such as the interaction between open-pit mining activities and underground portal development as well as time to arrange different mining equipment and personnel required for the transition to underground. Hence, second scenario demonstrated the possibilities of accommodating delays in the production during the transition from open-pit mining to underground mining by applying the Transition Period model.

For the first scenarios, with no delay in production during the transition from open-pit mining to underground mining, the result is shown in Figure 4-4. The optimized discounted value is $2.601 billion. The model suggested to extract the first eight levels (320m) by open-pit mining. A crown pillar is recommended to be placed at Level 9 and 10. Underground mining method will be adopted to extract the remaining reserve underneath the crown pillar. Hence, the optimal transition point and transition period is 320m and Period 3, respectively. Underground mining will start at Period 4.

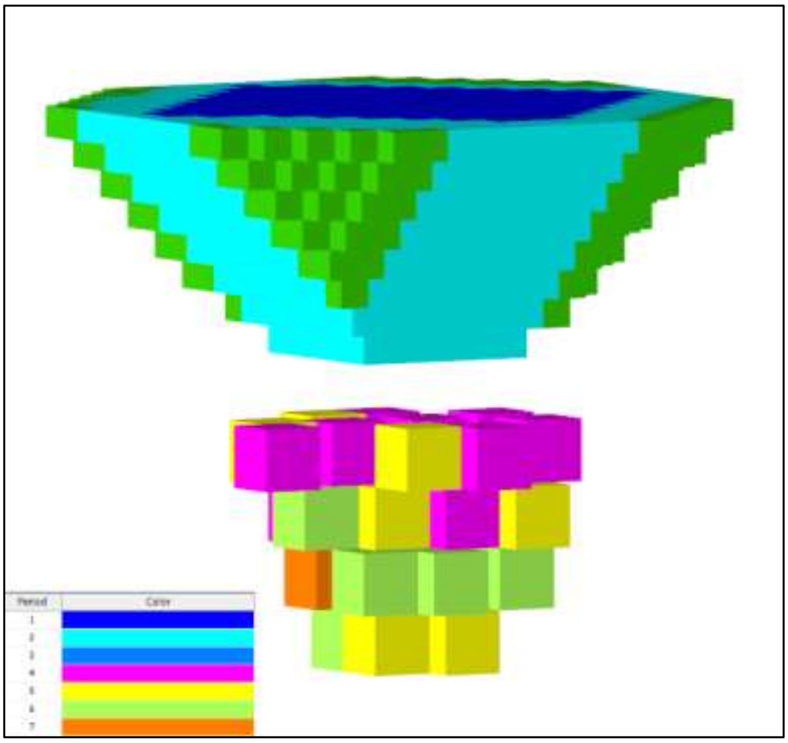

Figure 4-4: Scenario 1 Transition Period model result - with no delay

For the second scenario in which the Transition Period model is implemented with a two-period delay in production during the transition from open-pit to underground mining, the result is shown in Figure 4-5. The generated discounted value is $2.507 billion. The transition period and point are the same as the result of not having any delay period. The main difference between the two scenarios is that underground mining is scheduled to start at Period 6 instead of immediately after the open-pit mining ceased as in Scenario 2. In terms of the mine schedule result, there is no difference in quantity. The total material movement by period for both scenarios are plotted in the graph in Figure 4-6.

Table 4-2: Discounted profit generated by the Transition Period model – Scenarios 1 and 2

Implementations	Discounted profit
Transition Period model – scenario 1	$2.601 billion
Transition Period model – scenario 2	$2.507 billion

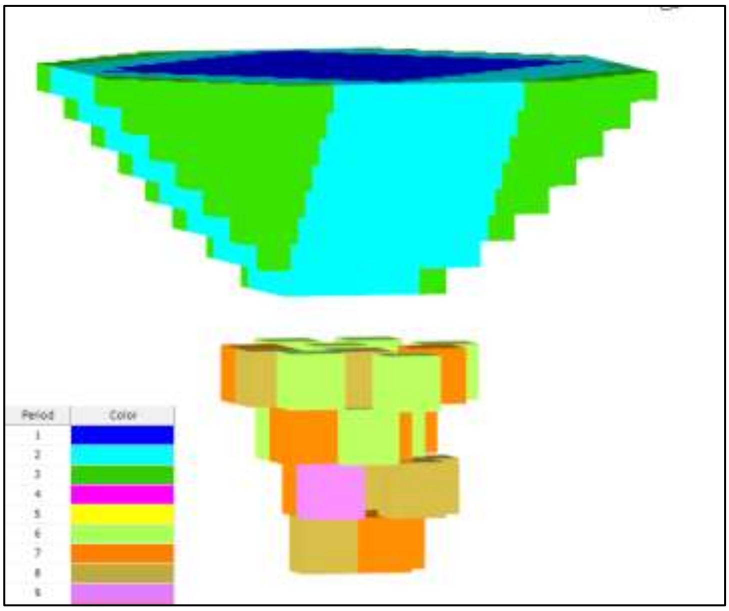

Figure 4-5: Scenario 2 Transition Period model result - with two-period delay

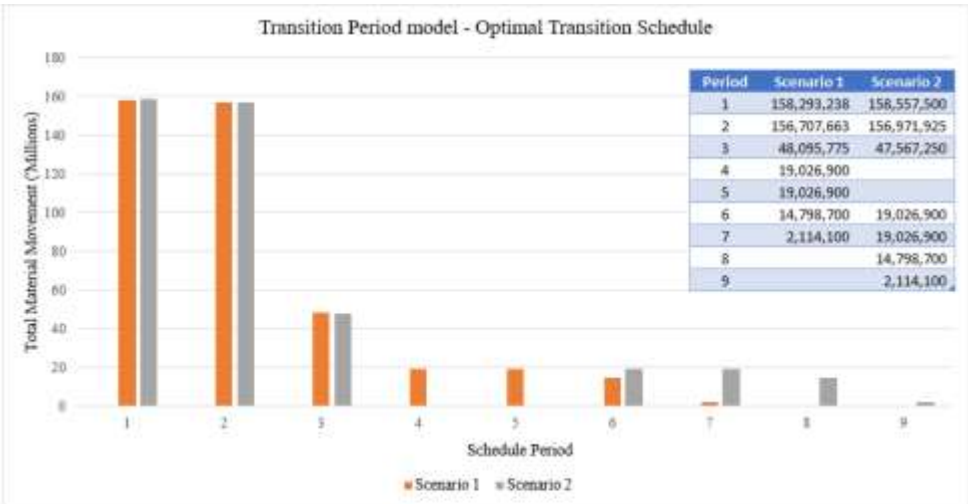

Figure 4-6: Graph and summary of the total material movement by period for Scenarios 1 and 2

Although the Transition Period model can solve the transition problem by providing an optimal transition point, optimal transition period, optimal mining layout and optimal mine schedule for the combination of open-pit and underground mining strategy, it is more computationally demanding in comparison to the Transition Point model. The number of variables, number of constraints and solution time are summarized as shown in Table 4. The table indicates that the growth of model complexity and increase of number of variables can affect the solution time exponentially. This is also one of the main reasons why in the mining industry, oftentimes, the pit optimization process and extraction period optimization process are treated independently. Despite that, the independency of those two processes are unable to produce optimal results. Hence, the Transition Point model creates value for the cases which are required to perform a brief evaluation for a combination of open-pit and underground mining strategy.

Table 4-3: Mathematical model size and solution time

Descriptions	Transition Point model	Transition Period model
Number of open-pit variables (blocks)	7,200	50,400
Number of underground variables (stopes)	326	2,282
Other variables	54	424
Total Constraints	30,898	193,998
Solution time (seconds)	10,213	44,679

4.3 CHALLENGES OF IMPLEMENTATION

From the implementation, it is evident that the mathematical models can solve the transition problem optimally. However, the foremost challenge of implementation of the models in a large/real dataset is the large-scale issue. The larger the scale of the problem, the higher the computation time. Hence, the large-scale problem is less likely to be solved within the reasonable time using a standard computer. To demonstrate the bottleneck of the model implementation, a range of instances of the problem have been solved and the result is shown in Table 5.

From the summary presented in Table 5, although, the practicality of the models proposed in this research project is valid and genuine and the problem of size reduction strategies have been adopted, it is still flawed. Thus, the large-scale issue for open-pit mining needs to be emphasized.

As mentioned in the literature (Section 2.2.1), open-pit mine planning and the optimization problem are computationally intractable due to their large-scale nature. Hence, reducing the number of binary integers in the LP model is the most important subject in this matter. The clustering or block aggregation technique is one of the available techniques to handle the large-scale problem. The clustering technique aims to aggregate blocks which have similar properties such as location, rock type, grade, and others. By aggregating blocks, a group of blocks can be represented as one entity, hence, reducing the number of binary variables in the LP model.

Table 4-4: Iterations summary

Transition Point Model: No. of Decision Variables	Transition Point Model: Solution Time (seconds)	Transition Period Model: No. of Variables	Transition Period Model: Solution Time (seconds)
962	443	9,638	616
3,059	867	30,616	1,298
5,396	4,218	53,992	5,113
10,046	4,694	100,500	7,455
14,896	6,660	149,006	33,090
25,092	9,081	250,974	45,575
30,640	10,020	306,458	78,913
42,760	10,812	427,664	> 4 days No solution

4.4 SUMMARY

This chapter has demonstrated the three-dimensional implementation for the two exact optimizations - Transition Point model and Transition Period model. From the series of results presented in this chapter, for the deposit which has potential to make the transition from open-pit to underground mining required underground mining to be considered as a viable mining strategy in the beginning of the planning and optimization processes. If the planning process is partial towards any mining method, it will impact the project value significantly. For instance, the stand-alone open-pit operation generates an undiscounted value of $2.362 billion. Meanwhile extending the optimization process to consider both open-pit and underground mining concurrently, the project generates a discounted value of $3.740 billion.

While taking account of time value and considering the mine schedule, the undiscounted cashflow of the case study without the delay period and with two-delay periods is \$2.601 and \$2.507 billion, respectively. Nevertheless, this research has proven the significance of studying both open-pit and underground mining concurrently in the mine planning and optimization processes for the combination of open-pit and underground mining strategy. However, the large-scale problem still remains as the main challenge for the implementation process. To overcome the large-scale challenge, the clustering technique is useful in reducing the LP model size by reducing the number of binary variables.

Chapter 5

HIERARCHICAL CLUSTERING ALGORITHM

This chapter discusses the hierarchical clustering algorithm. It gives a brief overview of the background of hierarchical clustering algorithm and two sections as following:

Section 5.2 discusses the hierarchical clustering algorithm proposed in this research.

Section 5.3 presents the two-dimensional implementation to ensure its validity.

5.1 BACKGROUND

The main challenge of the MILP models to solve the transition from open-pit to underground mining problem is their nature of having a huge number of binary decision variables. The increased number of variables leads to the exponential increase in computational time of the model. Moreover, the growing number of variables also increases the complexity of the model when considering the mining period - Transition Period model. During underground mining modelling process, a stope-based methodology is presented to aggregate the blocks into a mineable stope and it includes positive value stopes in the model implementation only in order to decrease the number of variables (Section 3.2). To further reduce the number of binary variables, the agglomerative hierarchical cluster analysis is employed to reduce the size of the problem of the open-pit model.

The agglomerative hierarchical cluster analysis is a bottom-up aggregation method. It considers each block or component as a cluster. The method starts with constructing a similarity matrix for all data points. Then, the aggregation starts from the most similar two clusters/blocks and make the way up to reach the desired number of clusters (Aggarwal 2014; Jain and Dubes 1988; Bailey 1975).

There are several agglomerative clustering methods such as single link, complete link, average, centroid, etc. Among these methods, the single link and complete link are the most commonly used methods. The single link hierarchical clustering method emphasizes mostly similar clusters. Hence, this method opts for the regions where clusters are closest. Due to this characteristic, this method is defined as local similarity-based method and capable to efficiently cluster different shapes of data objects such as non-elliptical and elongated shaped groups. On the other hand, the complete link hierarchical clustering method stresses on the dissimilarity of clusters. In other words, the cluster pairs with the least dissimilarity index will be merged. This behavior is considered as non-local similarity based method (Aggarwal 2014).

For mining applications, the most significant aspect for hierarchical clustering is the geospatial domain. The agglomerated clusters are required to be located in the same region due to practicality reasons. Hence, single link method is employed due to its suitability.

5.2 HIERARCHICAL CLUSTERING ALGORITHM

5.2.1 Notations and variables

Indices

i = index for block i

j = index for block j

Parameters

x_i = x-coordinate of block i

x_j = x-coordinate of block j

y_i = y-coordinate of block i

y_j = y-coordinate of block j

z_i =z-coordinate of block i

z_j = z-coordinate of block j

g_i = grade of block i

g_j = grade of block j

D_{ij} = Euclidean distance between block i and block j

N_{ij} = Neighborhood factor between block i and block j

G_{ij} = Grade factor between block i and block j

SF_{ij} = Slope factor between block i and block j

L_{ij} = Level factor between block i and block j

μ = A numerical factor assigned for localized neighborhoods

ω = A numerical penalty factor assigned for non-localized neighborhoods

δ = A numerical factor assigned to not maintaining the slope requirement

θ = A numerical penalty factor assigned to not maintaining the slope requirement

β = A numerical factor assigned for blocks that share the same level

τ = A numerical penalty factor for blocks that do not share the same level

5.2.2 Formulation

A similarity index is used to construct similarity matrix before block aggregation is exercised. In this research, the proposed similarity index calculation for cluster analysis is given by Equation 5-1:

$$S_{ij} = D_{ij} \times N_{ij} \times G_{ij} \times SF_{ij} \times L_{ij} \quad (5\text{-}1)$$

- Distance factor: This factor is to ensure that each cluster only consists of blocks that are close to each other. The distance factor for two blocks (block i and j) is calculated using the Euclidean distance method as shown in Equation (5-2).

$$D_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \quad (5\text{-}2)$$

- Neighborhood factor: The neighborhood factor is used to reinforce the localization of those aggregated blocks. A profile or an envelope needs to be defined as a boundary of a neighborhood. Then, all the blocks located inside the profile will be assigned a factor (μ) and those blocks sited outside the boundary will be penalized by a given penalty factor (ω). In the example shown in Figure 5-1, a profile of 3 by 3 blocks is employed. By using the neighborhood profile, the neighborhood members of Block 6 are $Block\ 6 = \{1, 2, 3, 5, 7, 9, 10, 11\}$. Hence, the neighborhood factor between Block 6 and each of the neighborhood members is μ.

$$N_{ij} = \begin{cases} \mu, \text{if block } j \text{ is within the neighbourhood profile of block } i \\ \omega, \text{otherwise} \end{cases} \tag{5-3}$$

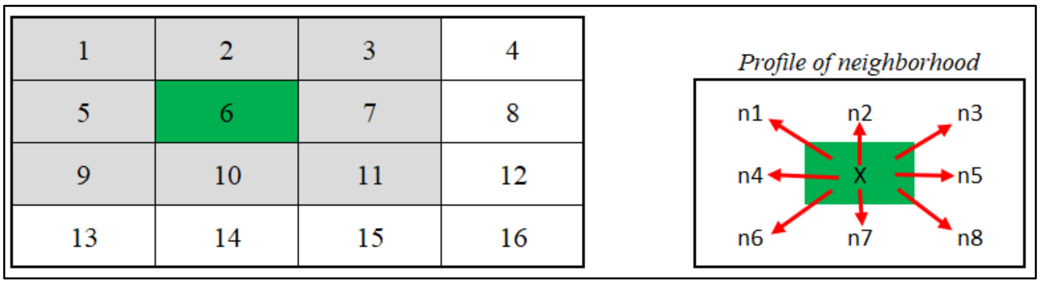

Figure 5-1: Neighborhood factor schematic

- Grade factor: This factor is to control the grade deviation of blocks within a cluster which is considered in production scheduling optimization as shown in Equation (5-4).

$$G_{ij} = (g_i - g_j)^2 \tag{5-4}$$

- Slope factor: This factor is assigned a penalty factor to those blocks that are located outside the slope requirement. δ is factor assigned to those blocks that are member of the blocks required to maintain the slope stability of a given block; δ should be a value greater than 0 and less than 1. A value of ω is assigned to those blocks which are not a member of the blocks required to maintain the slope requirement; θ should be a value greater than or equal to 1. For example, in Figure 5-2, the slope factor between Block 2 and Block 5 is δ; whereas, slope factor between Block 4 and Block 5 is θ as these are not members of required to maintain slope requirement between each other.

$$SF_{ij} = \begin{cases} \delta, \text{if block } j \text{ can help to maintain a slope stability of block } i \\ \theta, \text{otherwise} \end{cases} \tag{5-5}$$

Level 1	1	2	3
Level 2	4	5	6

Figure 5-2: Example for slope penalty factor

- Level factor: This factor is designed to penalize those blocks which are not located on the same level as a given block. β is assigned to those blocks that share the same level as a given block; β should be a value greater than 0 and less than 1; $\beta = (0,1)$. Whereas τ is allocated to those blocks not sharing the same level as a given block; τ should be a value greater than or equal to 1. For instance, using Figure 5-2 as an example, Blocks 4 and 5 are located on the same level (Level 2). Hence, L_{45} is assigned β. On the other hand, L_{25} is assigned τ as block 2 and 5 are located on level 1 and 2 respectively.

$$L_{ij} = \begin{cases} \beta, \text{if block } j \text{ is located on the same level as block } i \\ \tau, \text{otherwise} \end{cases} \tag{5-6}$$

Conceptually, if the combination of open-pit and underground mining strategy is employed, the final pit should be smaller than ultimate pit limit. Therefore, in the process of performing cluster analysis, only blocks within the ultimate pit are considered. Besides, the main reason for generating the ultimate pit limit before implementing the hierarchical clustering algorithm is to control any over-clustering. Over-clustering can happen when blocks in an area are very similar to each other. Waste blocks are good examples of this over-cluster effect. For instance, if there is no ultimate pit limit boundary for the clustering algorithm to run, waste blocks can cluster together as a big cluster group due to their similarity. Hence, this behavior can lead to an over-mined or under-mined clustering effect.

The process of performing cluster analysis is shown in Figure 5-3.

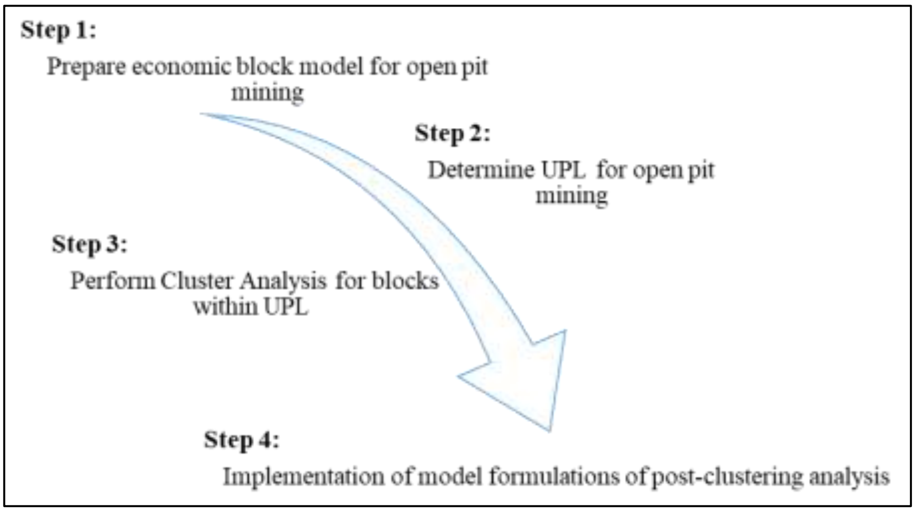

Figure 5-3: Process for Cluster Analysis

The third step presented in the cluster analysis process is as follows:

- Compute the similarity index of the blocks within the pit limit. The similarity index is presented in a matrix form and is shown in Figure 5-4.
- Run the hierarchical cluster analysis on Matlab (MATLAB R2015b) by using the similarity index formulation proposed in the research.

The process of clustering data has two most common types of methods which can be utilized to define the number of clusters. They are the natural division method and specifying arbitrary cluster method. The first method divides the dataset into discrete clusters using a ‘threshold’ value. This method allows the system to determine the natural partitions of the dataset. In this method, an inconsistency coefficient is normally used to verify the dissimilarity of clusters and inconsistent links between clusters. This method utilized the inconsistency coefficient function to create clusters. Hence, the inconsistent link plays a significant part in the process of determination of the natural division in a set of data. The second method which is the specifying arbitrary cluster method is relatively simple and straightforward. Basically, this method allows the user to determine the number of clusters and cluster data based on the height between two nodes in the cluster tree. This method is dependent on the user experience. Hence, it is difficult for a user to determine the correct number of clusters for the dataset. Whereas, due to the nature of the first methodology as explained, the clusters group created by this method is more appropriate for this research.

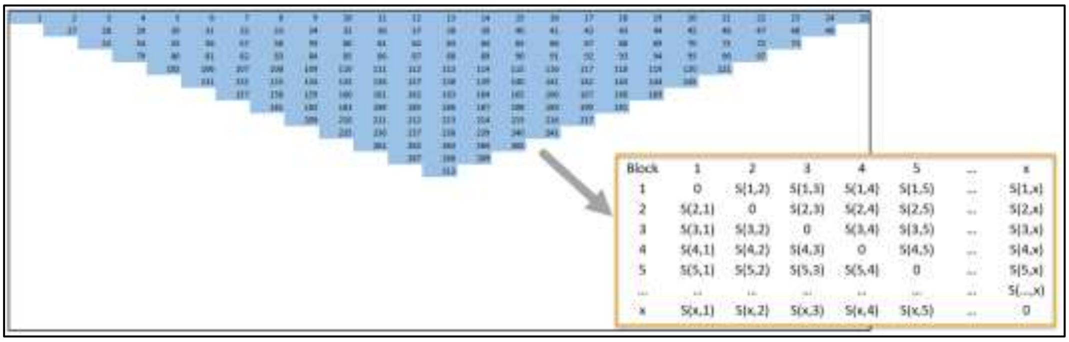

Figure 5-4: Sample of matrix generated by the similarity index computational process

5.3 VERIFICATION – TWO-DIMENSIONAL CASE STUDY

The two-dimensional case study shown in Section 3.5 was used for the verification process and the results were compared. As discussed earlier, the hierarchical cluster analysis is proposed to handle the computational complexity of the MILP model. The ultimate pit has been generated as shown in Figure 5-5. Then, the blocks within the ultimate pit are included in the cluster analysis process. All the details such as the location of the block, precedence blocks, block value and grade are tabulated into the cluster analysis model as shown in Table 6. These details are used to calculate the similarity index between blocks using the formula presented in Section 5.2. The computed similarity index is used to perform the hierarchical cluster analysis. The result generated is demonstrated in Figure 5-6. In Figure 5-6, each block has been assigned to a cluster attribute. Hence, blocks which share the same attributes belong to the same cluster. The cluster sets are substituted into the proposed model. The result generated by both the Transition Point Model and Transition Period Model is shown in Figure 5-7.

Table 5-1: Example of details or attributes required to perform cluster analysis

Block	Location	Grade	Block value	Precedence blocks
1	X1 , Y1	Grade (Block 1)	1	NIL
. . .	. . .	. . .	. . .	. . .
20	X2 , Y3	Grade (Block 20)	-3	Blocks{2,3,4}
. . .	. . .	. . .	. . .	. . .
128	X9 , Y8	Grade (Block 128)	2	Blocks{110,111,112}

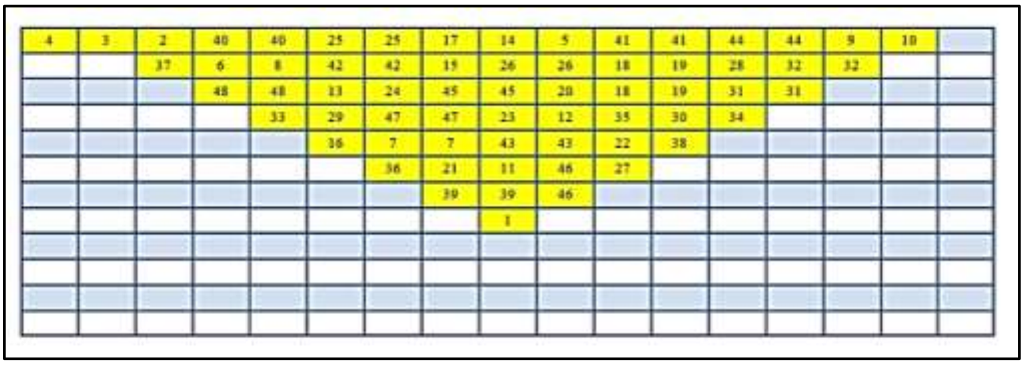

Figure 5-5: Cluster analysis result – cluster group

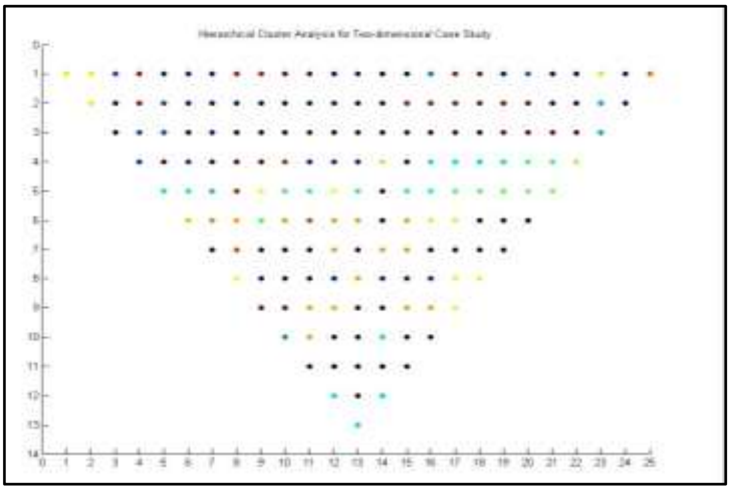

Figure 5-6: Result generated by the proposed hierarchical cluster algorithm

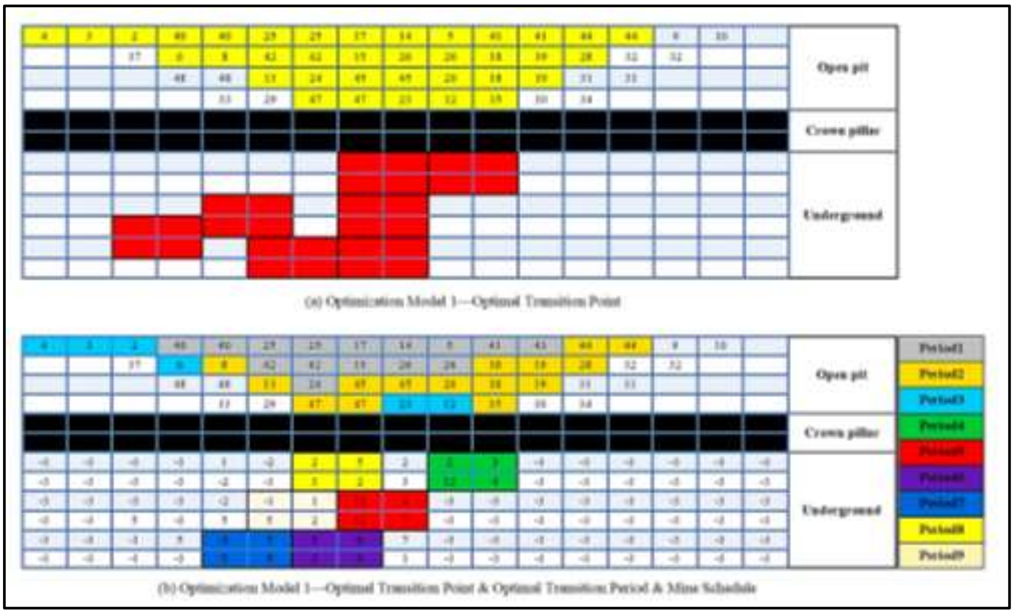

Figure 5-7: Mining layout generated post-cluster analysis

The results of Transition Point Model generated by pre-cluster analysis and post-cluster analysis are the same which is $207. However, the results of Transition Period Model generated from the pre-cluster and post-cluster analysis have shown differences. The main reason which leads to the difference is clustering effect'. Clustering effect occurred when several blocks are clustered and treated as a distinct element. As a result of cluster effect, in the open-pit sequencing process, the cluster which contains more than one block will have more precedence relationship. For example, in Figure 5-8, in order to mine cluster 20, cluster 18, 19 and 44 are required to be extracted due to clustering effect. Although the clustering effect in this two-dimensional case study has not affected the optimal transition point, it has impacted the transition period. The optimal transition period (generated pre-cluster analysis) is period 3, whereas the transition period post-cluster analysis is period 4. Due to the alteration in transition period post-cluster analysis, the mining layout and sequence for underground mining have been affected. Although there are some minor effects in post-cluster analysis in term of final pit layout, mining sequencing and transition period, the difference in NPV generated is minimal which is approximately 2%.

Table 5-2: Post-cluster Analysis results – two-dimensional case study

Scenario / Mining Strategy	Revenue
Transition Point Model	$207
Transition Period Model	$167

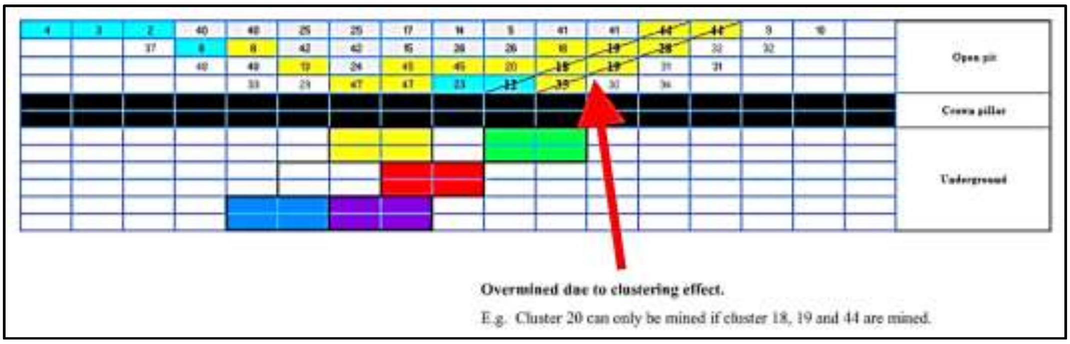

Figure 5-8: Over-mined due to clustering effect

5.4 SUMMARY

In summary, the main challenge of MILP model is the increase number of decision variables will boost the solution time exponentially. In this research, therefore, hierarchical clustering algorithm is employed to handle the computation complexity of the mine planning and optimization in open-pit mining. A new similarity index formulation is proposed to construct the similarity index for block aggregation purposes. The proposed similarity index is constructed by considering geospatial requirements along with grade factor and slope factor. The proposed hierarchical clustering algorithm is to be implemented within the ultimate pit limit. Once the block aggregation result is generated, it can be substituted into the proposed models in section 3 to generate the optimal transition point and period. A two-dimensional case study is presented to demonstrate the validity of the proposed methodology and comparisons of the results.

Chapter 6

LARGE-SCALE IMPLEMENTATION

The objective of this chapter is to present the large-scale implementation for the proposed hierarchical clustering algorithm along with the proposed optimization models.

Section 6.1 gives an overview of the case study;

Section 6.2 presents the results of the hierarchical clustering algorithm;

Section 6.3 presents the results generated from the proposed optimization models.

6.1 BACKGROUND INFORMATION

A gold deposit was used for the implementation of the combination of the hierarchical clustering algorithm and proposed transition models. The block model was obtained from Mining Education Australia (MEA) training material. A gold deposit is used for implementation of the combination of hierarchical clustering algorithm and the proposed transition models. The block model consists of 83,025 blocks with a block size 20mx20mx20m and average Au grade of 2.69 g/t.

The design size for the indicated stope size in an underground mining is eight stopes (2x2x2 blocks). The parameters used as inputs for the optimization models are shown in Table 8.

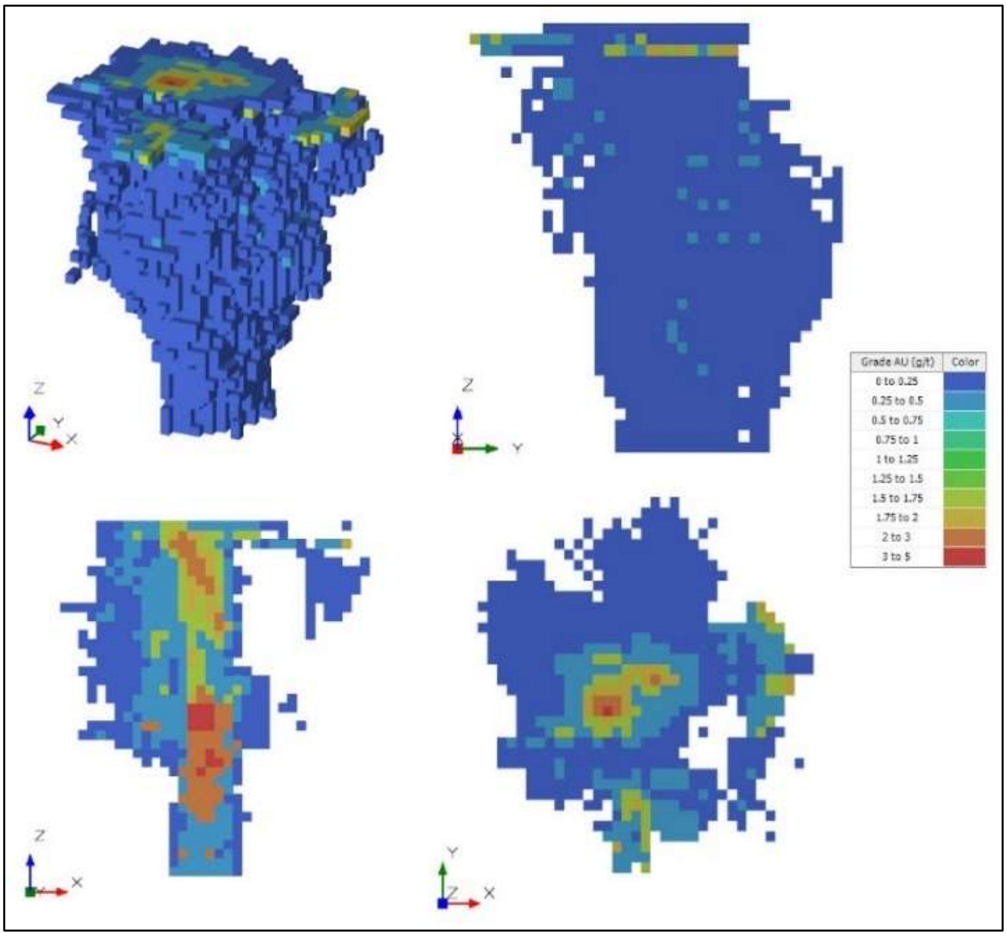

Figure 6-1: Resource distribution of gold grade (Au g/t)

Table 6-1: Scheduling parameters

Open-pit cut-off grade	1 g/t
Underground cut-off grade	2 g/t
Recovery	90%
Discount Rate	12%
Crown pillar	2 Levels
Underground development rate	8 levels per period
Mining capacity for open-pit – Maximum	150 million tonnes per scheduling period
Mining capacity for underground – Maximum	37 million tonnes per scheduling period
Blocks per stope	8 blocks
Precedence blocks for open-pit	5 blocks
Schedule period	7
Slope angle	45 degrees

6.2 HIERARCHICAL CLUSTERING ALGORITHM

As mentioned in the literature section, while considering underground mining as a viable option in the initial mine planning and optimization processes, the final pit is likely to be smaller than the UPL generated by a stand-alone open-pit case scenario. The ultimate pit limit consists of 13,555 blocks which are shown in Figure 6-2. The UPL ends at level 19 which is at 460m. Figure 6-2 displays the UPL generated for the hierarchical clustering algorithm.

Then, those blocks within the ultimate pit limit are substituted into the hierarchical clustering algorithm as shown in Section 5.2. The result from the hierarchical clustering algorithm successfully reduced the size of the open-pit mining from 13,555 blocks into 1,507 clusters which is approximately 85% of reduction in size of the problem. In the result generated by the hierarchical clustering algorithm, the maximum number of blocks in a cluster group is 39 blocks. Table 9 tabulates the count of the number of blocks within the each of the first 30 cluster groups. For example, in Table 9, each of Clusters 1 to 4 consists 3 blocks and Cluster 7 contains 6 blocks. Table 10 presents the example of the list of the blocks within the first seven set of the cluster group. In Table 10, the three blocks contained in Cluster 1 (as mentioned in Table 9) are X22Y22Z37, X22Y23Z37 and X22Y24Z37. Overall, the clustering algorithm used 8,936 seconds to solve the clustering problem.

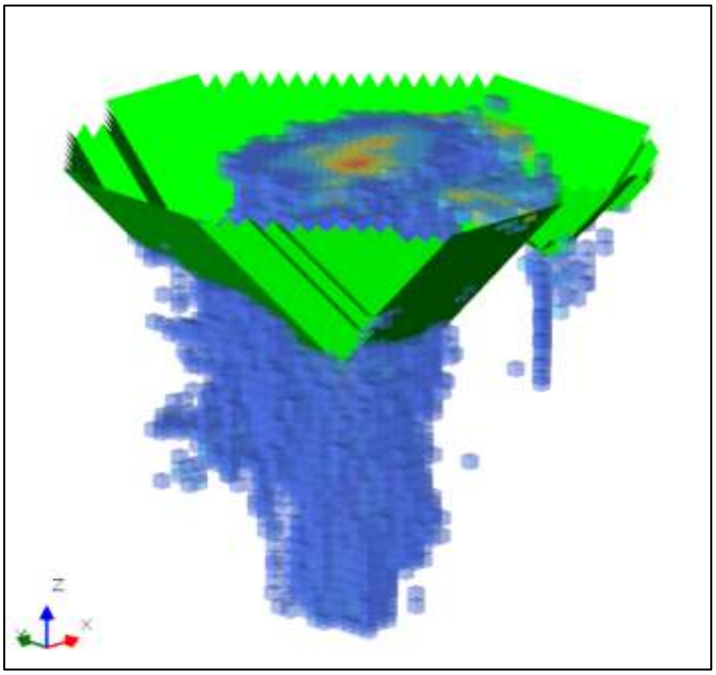

Figure 6-2: Ultimate pit limit pre-cluster

Table 6-2: Hierarchical clustering algorithm result by number of blocks

Clusters	Count of BLOCK_ID
1	3
2	3
3	3
4	3
5	4
6	4
7	6
8	6
9	7
10	6
11	3
12	3
13	6
14	1
15	6
16	3
17	3
18	3
19	5
20	7
21	4
22	4
23	8
24	8
25	9
26	6
27	6
28	7
29	7
30	8

Table 6-3: Hierarchical clustering algorithm result by number of clusters

Clusters	BLOCK_ID
1	X22Y22Z37
1	X22Y23Z37
1	X22Y24Z37
2	X24Y22Z37
2	X24Y23Z37
2	X24Y24Z37
3	X22Y22Z38
3	X22Y23Z38
3	X22Y24Z38
4	X24Y22Z38
4	X24Y23Z38
4	X24Y24Z38
5	X21Y21Z38
5	X21Y22Z38
5	X21Y23Z38
5	X21Y24Z38
6	X21Y25Z38
6	X22Y25Z38
6	X23Y25Z38
6	X24Y25Z38
7	X20Y20Z37
7	X20Y21Z37
7	X20Y22Z37
7	X20Y23Z37
7	X20Y24Z37
7	X20Y25Z37

6.3 OPTIMIZATION MODELS

The output from the clustering algorithm is substituted into the proposed optimization models as an open-pit mining input. Besides, the underground mining input has been proposed by using the stope-based methodology. Hence, the input for underground mining reduced to 2,136 positive stopes from 83,025 blocks. This process took 12 hours and 16 minutes to be completed. The larger the block model, the more the combination of stopes required, hence, it requires much longer time to process and define all the profitable stopes.

The number of variables and constraints required for both optimization models are shown in Table 11. For the Transition Point Model which only considers the optimal transition point, it consists of approximately 292,000 constraints and a total of approximate 4,000 binary variables. The Transition Point Model takes approximately 83 seconds to solve. On the other hand, the Transition Period Model which considers both optimal transition point and period require approximately 1.8 million and 32,000 as number of constraints and variables, respectively. As the scale of the problem increases drastically compared to the Transition Point Model, the Transition Period Model uses approximately 30 hours to generate a result with the gap of 4.9%. Both models were solved on a standard computer with a specification of 2.8 GHz CPU and 16 GB RAM.

Table 6-4: Number of variables and constraints for large-scale implementation

Descriptions	Transition Point Model	Transition Period Model
Number of open-pit variables (clusters)	1,507	15,049
Number of underground variables (stopes)	2,136	14,952
Other variables	123	1,435
Total Constraints	291,952	1,790,891
Solution time (seconds)	83	93,463

The result of the Transition Point Model is presented in both Figure 6-3 and Table 12. Figure 6-3 illustrates the optimal layout of the transition problem and Table 12 shows the mining method selection by level. The result for Transition Point Model recommends ceasing open-pit mining at level 24 which is at a depth of 360m and leave Levels 22-23 as a crown pillar. Underground mining method is suggested to start from Level 21 and below. From the result generated by Transition Point Model, 13,390 blocks are to be mined by open-pit mining and 112 stopes are to be mined by underground mining. The undiscounted cashflow generated by Model 1 was $191.02 million.

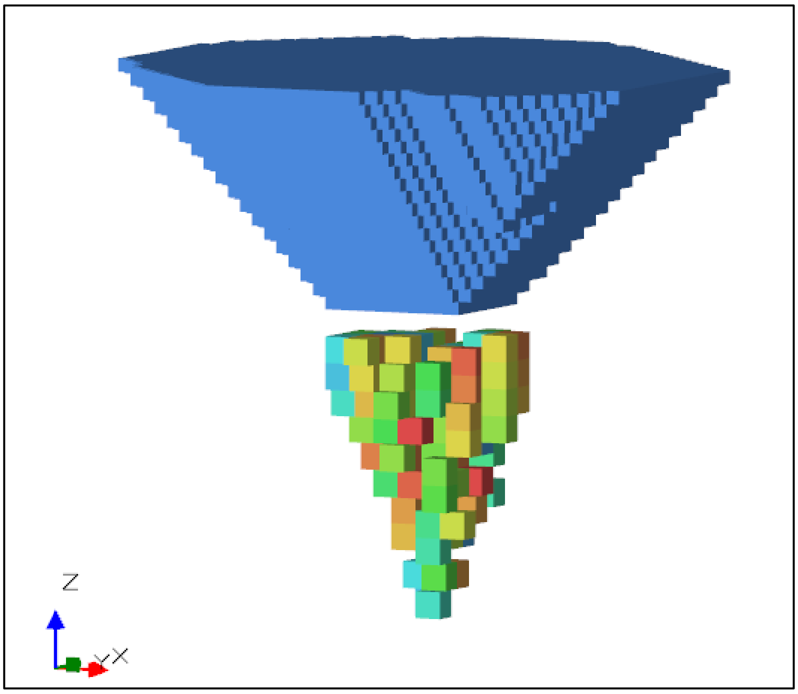

Figure 6-3: Transition Point Model result of large-scale implementation

The result of the Transition Period Model is shown in Figure 6-4 and Table 13. In Table 13, the result suggests mining the first 20 levels by open-pit mining which is up to 400m deep and mine Level 19 and below by underground mining. The two levels below the final pit are left as a crown pillar. The NPV generated by the Transition Period Model is $145.71 million. The material movement by period generates by the Transition Period Model is shown in Table 14. The total material movement for the entire schedule periods is approximately 309 million tonnes with the split of ore and waste of approximately 92 million tonnes and 217 million tonnes, respectively.

The results proved that, by using the hierarchical clustering algorithm, the proposed Transition Point Model and Transition Period Model are able to be implemented in a larger dataset. Therefore, the main challenge discussed in Section 4.3 which is the large-scale issue can be eased by using the proposed hierarchical clustering algorithm.

Table 6-5: Mining method selection by level – Transition Point Model

Z Level	TP_OP/UG	Z Level	TP_OP/UG
41	OP	21	UG
40	OP	20	UG
39	OP	19	UG
38	OP	18	UG
37	OP	17	UG
36	OP	16	UG
35	OP	15	UG
34	OP	14	UG
33	OP	13	UG
32	OP	12	UG
31	OP	11	UG
30	OP	10	UG
29	OP	9	UG
28	OP	8	UG
27	OP	7	UG
26	OP	6	UG
25	OP	5	UG
24	OP	4	UG
		3	UG
		2	UG
		1	UG

OP = Open Pit
UG = Underground

Table 6-6: Mining method selection by level – Transition Period Model

Z Level	OP/UG	Z Level	OP/UG
41	OP	19	UG
40	OP	18	UG
39	OP	17	UG
38	OP	16	UG
37	OP	15	UG
36	OP	14	UG
35	OP	13	UG
34	OP	12	UG
33	OP	11	UG
32	OP	10	UG
31	OP	9	UG
30	OP	8	UG
29	OP	7	UG
28	OP	6	UG
27	OP	5	UG
26	OP	4	UG
25	OP	3	UG
24	OP	2	UG
23	OP		
22	OP		

OP = Open Pit
UG = Uderground

Table 6-7: Mining tonnes by period

Transition Period	OP/UG	Ore Tonnes	Waste Tonnes	Total Tonnes	Au (g/t)
1	OP	50,877,600	96,906,400	147,784,000	1.80
2	OP	30,740,000	119,944,000	150,684,000	1.73
3	UG	2,784,000		2,784,000	2.73
4	UG	4,245,600	23,200	4,268,800	2.72
5	UG	2,969,600		2,969,600	2.71
6	UG	185,600		185,600	2.56
7	UG	185,600		185,600	2.62
Total		**91,988,000**	**216,873,600**	**308,861,600**	**2.01**

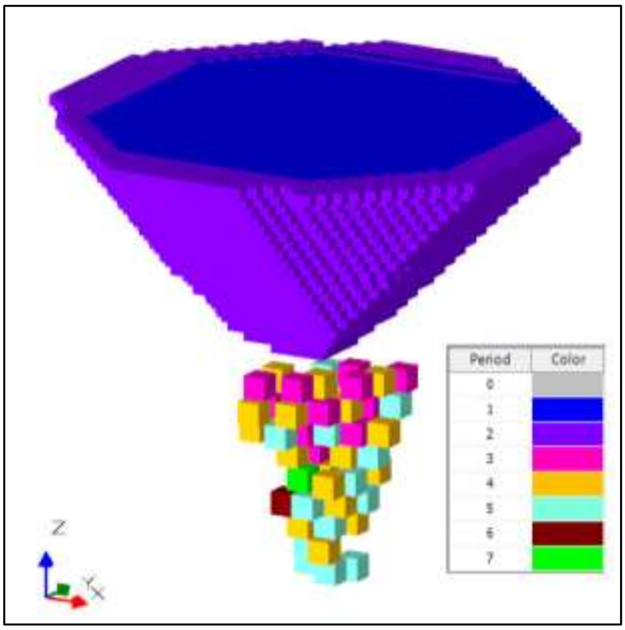

Figure 6-4: Model 2 result of large-scale implementation

6.4 SUMMARY

This chapter has demonstrated the implementation of the hierarchical clustering algorithm in association with the proposed mathematical models. The hierarchical clustering algorithm used nearly 2.5 hours to generate a result. It produced 1,507 clusters which reduced the decision variables by over 85%. The reduction of decision variables shortened the solution time significantly. The Transition Point Model and Transition Period Model are solved in approximately 80 seconds and 26 hours, respectively. The hierarchical clustering algorithm proved its ability to expand the problem scale in the transition models.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

This chapter is divided into two sections.

Section 7.1 summarizes the work completed for this thesis;

Section 7.2 outlines the recommendation for future studies.

7.1 THESIS SUMMARY AND CONTRIBUTION TO KNOWLEDGE

Traditionally, open-pit mining and underground mining have been studied as a separately project in cases where shallow deposits extend to a considerable depth. For shallow deposits, usually, open-pit mining acts as a primary mining method in the beginning of the project. When open-pit mining is approaching its end of mine life, the 'transition problem' becomes apparent. It is a decision-making process of whether to extend the pit or make the transition from open-pit mining to underground mining. If open-pit is no longer an economic mining method for the remaining mineral resource, the decision is to make the transition to underground mining. If the final decision is to make the transition to underground mining, a combination of open-pit and underground mining strategy is then adopted by the mining project.

In the practice of combination of open-pit and underground mining strategy, the inability to maximize project value (discounted or/and undiscounted) and resource utilization using open-pit mining and underground mining are studied independently. The main contributions for the failures are arbitrary crown pillar location and over-mined pit. Literature shows the significance of considering both open-pit and underground concurrently in the mine planning and optimization process.

This research used mathematical modelling techniques to approach the transition problem and proposed mathematical models for generating the transition point, transition period and crown pillar placement to solve the transition problem. The proposed models can provide a clear guidance on where an operation should consider making the transition from open-pit to underground mining.

Given the computation complexity of the problem, this research utilized data clustering techniques in open-pit mining and stope-based methodology for underground mining. The main reason was to decrease the size of the problem for both open-pit mining and underground mining. For open-pit mining, the data clustering technique was used to aggregate blocks with similar properties in order to decrease the number of entities. Meanwhile, the size reduction strategy for underground mining aims to retain the profitable stopes and eliminate the unprofitable stopes.

In this research, two mathematical models were developed and presented. The first model is Transition Point Model which is used to generate the optimal transition point by maximizing the undiscounted cashflow of the mine operation. Following that, the modelling process expands to achieve optimal transition point and period by maximizing the discounted cashflow of the mine operation by developing the Transition Period Model. The expansion considers the underground mining advancement, operation capacities, blending, capital expenditure for making the transition, underground mine design parameters, etc. The results generated by this expanded model provide a comprehensive transition plan to the engineers. These two models were tested with a simplified two-dimensional dataset for verification purposes. The results showed that all constraints were satisfied.

The two models were implemented on a synthetic block model which consists of 7,200 blocks. Upon solving the mathematical models, the optimal transition point, optimal crown pillar placement, and/or optimal transition period were generated. The Transition Point Model provides the optimal mining layout along with maximized undiscounted cashflow of \$3.74 billion. One the other hand, Transition Period Model provides the optimal mining layout along with maximized discounted cashflow of \$2.60 billion without considering any delay period when transitioning from open-pit to underground. In contrast, while considering a two-delay period, the discounted cashflow is \$2.51 billion. Although the implementations demonstrate the capability of the models, it is unable to be implemented in a real case study. This is due to the computational complexity of real cases. A series of case studies were presented to show the limitation of the proposed models.

The hierarchical clustering algorithm was proposed to handle the computational complexity and reduce the size of the problem for open-pit mining. The proposed hierarchical clustering utilized the similarity index to perform the clustering action. The similarity index is calculated based on coordinate, slope factor, grade factor, neighborhood factor and level factor. A two-dimensional case study was utilized to test the validity of the model.

The two models along with the hierarchical clustering algorithm were implemented on a larger scale dataset which consists of 83,025 blocks. The open-pit blocks were clustered into 1,507 cluster groups and underground mining generated 2,136 profitable stopes. The proposed clustering algorithm successfully reduced the size of the problem for open-pit mining by approximately 85%. The implementation of the hierarchical clustering algorithm along with the proposed transition models were successful. The solution time for the Transition Point Model and Transition Period Model reduced significantly.

In conclusion, a new methodology for solving the transition problem was developed, verified, and implemented. It can optimize the undiscounted cashflow and NPV of a mining project and can provide the optimal mining strategy for the project.

7.2 RECOMMENDATIONS

The first area of enhancement is a stochastic mining model which can generate a high confidence mining plan. Geological uncertainty has remained as a topical discussion within the mining industry and it will directly affect the mine planning and optimization process. As presented by Chung, Topal, and Erten (2015), geological uncertainty will affect the transition problem significantly. Hence, a ‘grey area’ will appear which is named as the ‘Transition envelope’. Future studies could consider transforming the model into a stochastic mining model where it can handle multiple inputs and generate a result with higher confidence levels.

Furthermore, future improvement by introducing flexibility for underground mining advancement can be considered. In the current model, the underground mining advancement is top-down approach. In the future studies, it can consider expanding the underground mining advancement into bottom-up and complete flexibility. Besides, the model has the potential to further improve to allow for simultaneous open-pit and underground mining operation/production.

Next, the aspect that should be concerned with in future studies is the underground development mining schedule. Underground development scheduling can directly affect the stope mining sequence and accessibility. Hence, it will help to generate a more detailed and practical mining plan if there are dependencies/links between development and stopes. Moreover, the sublevel stoping cycle should be involved in the future studies.

For instance, backfilling is one of the major concerns that can constrain the sequence and scheduling process.

Additionally, the clustering algorithm methodology can be expanded in future studies such as using meta-heuristic and hyper-heuristic approaches. Different clustering methodologies will provide different results and different purposes. Hence, it will help to provide flexibility to the flow of the process.

Last but not the least, a user-friendly interface is required to be developed. Hence, an external user can utilize this algorithm without the need to know about coding or replicating the work of this research. This aspect also can be expanded to make it as a stand-alone software which does not require other mining packages for visualization, etc.

REFERENCES AND BIBLIOGRAPHY

Abdollahisharif, J., E. Bakhtavar, K. Shahriar, J. Sobczyk, and J. Kicki. 2008. "Open-pit to underground mining-where is the optimum transition depth." *Proceedings of 21st WMC & Expo 2008*:189-196.

Aggarwal, Charu C. 2014. *Data classification: algorithms and applications*: CRC press.

Alford, C. 1995. "Optimization in underground mine design." Proceedings of 25th International APCOM Symposium, Brisbane, Australia.

Alford, Christopher, Marcus Brazil, and David H Lee. 2007. "Optimisation in underground mining." In *Handbook of Operations Research in Natural Resources*, 561-577. Springer.

Arnold, Timothy D. 1996. "Underground mining: a challenge to established open-pit operations." *Mining Engineering(Littleton, Colorado)* 48 (4):25-29.

Askari-Nasab, H, M Tabesh, MM Badiozamani, and H Eivazy. 2010. "Hierarchical clustering algorithm for block aggregation in open-pit mines." Mine planning and equipment selection (MPES), Fremantle, WA, Australia.

Askari-Nasab, Hooman. 2006. *Intelligent 3 D interactive open-pit mine planning and optimization*. Vol. 68.

Askari-Nasab, Hooman, and Kwame Awuah-Offei. 2009. "Open-pit optimisation using discounted economic block values." *Mining Technology* 118 (1):1-12.

Askari-Nasab, Hooman, Kwame Awuah-Offei, and Hesameddin Eivazy. 2010. "Large-scale open-pit production scheduling using mixed integer linear programming." *International Journal of Mining and Mineral Engineering* 2 (3):185-214.

Ataee-Pour, Majid. 2004. "Optimisation of stope limits using a heuristic approach." *Mining Technology* 113 (2):123-128.

Bailey, Kenneth D. 1975. "Cluster Analysis." *Sociological Methodology* 6:59-128.

Bakhtavar, E., K. Shahriar, and A. Mirhassani. 2012. "Optimization of the transition from open-pit to underground operation in combined mining using (0-1) integer programming." *J. South. Afr. Inst. Min. and Metall.* 112 (12):1059-1064.

Bakhtavar, E., K. Shahriar, and K. Oraee. 2008. "A model for determining optimal transition depth over from open-pit to underground mining." Proceedings of the 5th International Conference on Mass Mining.

Brannon, C., T. Casten, and M. Johnson. 2004. "Design of the Grasberg block cave mine." *Proceedings of Massmin*:623-628.

Caccetta, Louis. 2007. "Application of optimisation techniques in open-pit mining." In *Handbook of operations research in natural resources*, 547-559. Springer.

Caccetta, Louis, and SP Hill. 1999. "Optimization techniques for open-pit mine scheduling." International congress on modelling and simulation.

Camus, J.P. 1992. "Open-pit optimization considering an underground alternative." Proceedings of 23th International APCOM Symposium, Tucson.

Carlson, Thomas R, John D Erickson, DT O'Brain, and Milton T Pana. 1966. "Computer techniques in mine planning." *Mining Engineering* 18 (5):53-56.

Cheimanoff, NM, EP Deliac, and JL Mallet. 1989. "GEOCAD: an alternative CAD and artificial intelligence tool that helps moving from geological resources to mineable reserves." 21st APCOM.

Chung, J., M. Asad, Topal E., and A.G. Ghosh. 2016. "Determination of the Transition Point from Open-pit to Underground Mining." Open-pit Operators' Conference 2016, Kalgoorlie.

Chung, J., E. Topal, and O. Erten. 2015. "Transition from open-pit to underground - using Integer Programming considering grade uncertainty." The 17th annual conference of the International Association for Mathematical Geosciences, Freiberg, Germany, 5-13 September 2015.

Chung, J., E. Topal, and A.G. Ghosh. 2016. "Where to make the transition from open-pit to underground? Using integer programming." *Journal of the Southern African Institute of Mining and Metallurgy* 116 (8):801-808.

Dagdelen, K, and T.B Johnson. 1986. "Optimum open-pit mine production scheduling by Lagrangian parameterization." *Proc. of the 19th APCOM*:127-142.

Dagdelen, K., and I. Traore. 2014. "Open-pit transition depth determination through global analysis of open-pit and underground mine scheduling." Orebody Modelling and Strategic Mine Planning, Perth, Australia, 24-26 November 2014.

Dagdelen, Kadri, and Kazuhiro Kawahata. 2007. "Opportunities in multi-mine planning through large scale mixed integer linear programming optimization." Proc. 33rd International Symposium on Computer Application in the Minerals Industry (APCOM).

Denby, B, and D Schofield. 1994. "Open-pit design and scheduling by use of genetic algorithms." *Transactions of the Institution of Mining and Metallurgy. Section A. Mining Industry* 103.

Eivazy, H, and H Askari-Nasab. 2012. "A mixed integer linear programming model for short-term open-pit mine production scheduling." *Mining Technology* 121 (2):97-108.

Elevli, Birol. 1995. "Open-pit mine design and extraction sequencing by use of OR and AI concepts." *International Journal of Surface Mining and Reclamation* 9 (4):149-453.

Fiscor, S. 2010. "Major Open-Pit Copper Mines Move Underground." *Engineering and Mining Journal* 211 (5).

Flores, German, and Alex Catalan. 2019. "A transition from a large open pit into a novel “macroblock variant” block caving geometry at Chuquicamata mine, Codelco Chile." *Journal of Rock Mechanics and Geotechnical Engineering* 11 (3):549-561.

Fuentes, S.S. 2004. "Going to an underground (UG) mining method." MassMin Conference, Santiago, Chile.

Gershon, Mark. 1987. "Heuristic approaches for mine planning and production scheduling." *International Journal of Mining and Geological Engineering* 5 (1):1-13.

Grieco, N, and R Dimitrakopoulos. 2007. "Managing grade risk in stope design optimisation: probabilistic mathematical programming model and application in sublevel stoping." *Mining technology* 116 (2):49-57.

Grobler, F. 2015. "Optimsed Decision-Making under Grade Uncertatinty in Surface Mining." Doctor of Philosophy, Department of Mathematics & Statistics, Curtin University of Technology.

Hochbaum, Dorit S. 2001. "A new—old algorithm for minimum‐cut and maximum‐flow in closure graphs." *Networks: An International Journal* 37 (4):171-193.

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

IBM CPLEX. 2013. IBM CPLEX Solver.

Jain, Anil K, and Richard C Dubes. 1988. *Algorithms for clustering data*: Prentice-Hall, Inc.

Jain, Anil K, M Narasimha Murty, and Patrick J Flynn. 1999. "Data clustering: a review." *ACM computing surveys (CSUR)* 31 (3):264-323.

Johnson, S.C. 1967. "Hierarchical clustering schemes." *Psychometrika* 32 (3):241-254.

Johnson, T.B. 1968. Optimum open-pit mine production scheduling. Berkeley: DTIC Document.

Khaboushan, Ali Soltani, Morteza Osanloo, and Akbar Esfahanipour. 2020. "Optimization of open-pit to underground transition depth: An idea for reducing waste rock contamination while maximizing economic benefits." *Journal of Cleaner Production* 277:123530.

Lerchs, H., and F.I. Grossman. 1964. "Optimum design of open-pit mines." Operations Research.

Little, J, P. Knights, and E. Topal. 2013. "Integrated optimization of underground mine design and scheduling." *J. South. Afr. Inst. Min. and Metall.* 113 (10):775-785.

Little, J, and E Topal. 2011. "Strategies to assist in obtaining an optimal solution for an underground mine planning problem using Mixed Integer Programming." *Int. J. of Mining and Mineral Engineering* 3 (2):152-172.

Little, Jade. 2012. "Simultaneous optimisation of stope layouts and production schedules for long-term underground mine planning." Doctor of Philosophy, School of Mechanical and Mining Engineering, The University of Queesland.

Mai, N. L., E. Topal, and O. Erten. 2018. "A new open-pit mine planning optimization method using block aggregation and integer programming." *Journal of the Southern African Institute of Mining and Metallurgy* 118 (7). doi: 10.17159/2411-9717/2018/v118n7a4.

Mai, Ngoc Luan. 2017. "A New Mine Planning Methodology using Topcone Algorithm and Mathematical Programming." Thesis (Ph.D.)--Curtin University., Curtin University, Western Australia School of Mines.

Maptek. 2011. Vulcan 8.1 Stope Optimizer.

Nehring, M, and E Topal. 2007. "Production schedule optimisation in underground hard rock mining using mixed integer programming." *Proceedings-Project Evaluation. The Australasian Institute of Mining and Metallurgy, Melbourne*:169-175.

Nehring, M, E Topal, M Kizil, and P Knights. 2010. "An investigation to integrate optimum long-term planning with short planning in underground mine production scheduling." Proc. Mine Planning and Equipment Selection (MPES) Conf., Fremantle, Australia.

Newman, A.M., C. A. Yano, and E. Rubio. 2013. "Mining above and below ground: Timing the transition." *IIE Transactions* 45 (8):865-882.

Nilsson, D. 1992. "Surface Vs. Underground Methods." In *SME Mining Engineering Handbook*, edited by H.L Hartman, 2058-2069. Colorado: Society of Mining, Metallurgy, and Exploration, Inc.

Opoku, S., and C. Musingwini. 2013. "Stochastic modelling of the open-pit to underground transition interface for gold mines." *Int. J. Min. Reclam. and Environ.* 27 (6):407-424.

Ramazan, S, K Dagdelen, and TB Johnson. 2005. "Fundamental tree algorithm in optimising production scheduling for open-pit mine design." *Mining Technology* 114 (1):45-54.

Ramazan, S, and R Dimitrakopoulos. 2004. "Recent applications of operations research and efficient MIP formulations in open-pit mining." *Society of Mining, Metallurgy, and Exploration, Inc. Transactions* 316.

Ramazan, Salih. 2001. "Open-pit mine scheduling based on fundamental tree algorithm."

Ramazan, Salih. 2007. "The new fundamental tree algorithm for production scheduling of open-pit mines." *European Journal of Operational Research* 177 (2):1153-1166.

Ren, H., and E. Topal. 2014. "Using Clustering Methods for Open-pit Mine Production Planning." *MINING EDUCATION AUSTRALIA*:45.

Soderberg, A., and D.O. Rausch. 1968. "Pit Planning and Layout." In *Surface Mining*, edited by E.P Pfleider, 142-143. New York: The American Institute of Mining, Metallurgical, and Petroleum Engineers, Inc.

Sulistyo, Rafiq, Alibasjah Soedjarno, and Togar M Simatupang. 2015. "Operations Strategy Formulation: Shifting From Grasberg Open-Pit Surface Mining To Underground Mining." *INTERNATIONAL RESEARCH JOURNAL OF BUSINESS STUDIES* 4 (1).

Tabesh, M, and H Askari-Nasab. 2011. "Two-stage clustering algorithm for block aggregation in open-pit mines." *Mining Technology* 120 (3):158-169.

Tolwinski, Boleslaw, and Robert Underwood. 1992. "An algorithm to estimate the optimal evolution of an open-pit mine." The 23rd International Symposium on the Application of Computers and Operations Research in the Mineral Industry (APCOM) Proceedings. SME Publications, Littleton, Co.

Topal, E. 2008. "Early start and late start algorithms to improve the solution time for long-term underground mine production scheduling." *J. South. Afr. Inst. Min. and Metall.* 108 (2):99-107.

Topal, E., and J Sens. 2010. "A new algorithm for stope boundary optimization." *Journal of Coal Science and Engineering (China)* 16 (2):113-119. doi: 10.1007/s12404-010-0201-y.

Trout, LP. 1995. "Underground mine production scheduling using mixed integer programming." 25th International APCOM Symposium Proceedings.

Tulp, T. 1998. "Open-pit to underground mining." In *Mining Planning and Equipment Selection 1998*, edited by R.K Singhal, 7-12. Netherlands: A.A. Balkema.

VB.NET. 2015. Visual Studio. https://docs.microsoft.com/en-us/dotnet/visual-basic/.

Whittle, D., M. Brazil, P. A. Grossman, J. H. Rubinstein, and D. A. Thomas. 2018. "Combined optimisation of an open-pit mine outline and the transition depth to underground mining." *European journal of operational research* 268 (2):624-634. doi: 10.1016/j.ejor.2018.02.005.

Whittle, J. 1990. "Open-pit Optimization." In *Surface Mining*, edited by B Kennedy, 470-475. Colorado: Society of Mining, Metallurgy, and Exploration.

APPENDIX A: DATA PREPARATIONS

<pre>Dim AppXL As Excel.Application Dim wbXL As Excel.Workbook Dim wbsXL As Excel.Workbooks Dim shXL_1C, shXL_1 As Excel.Worksheet AppXL = CreateObject("excel.application") wbsXL = AppXL.Workbooks wbXL = wbsXL.Open(TextBox32.Text) shXL_1C = wbXL.Worksheets("Sheet1_C") shXL_1 = wbXL.Worksheets("Data") Dim count As Double = 0 Dim BinariesCount As Double = 0 Dim n, m As Double Dim X As Double = Double.Parse(TextBox36.Text) Dim Y As Double = Double.Parse(TextBox37.Text) Dim Z As Double = Double.Parse(TextBox38.Text) Dim COG_OP As Double = Double.Parse(TextBox43.Text) Dim Mcost_OP_lvlStart As Double = Double.Parse(TextBox60.Text) Dim Met_P As Double = Double.Parse(TextBox55.Text) Dim ton As Double = Double.Parse(TextBox39.Text) * Double.Parse(TextBox40.Text) * Double.Parse(TextBox41.Text) * Double.Parse(TextBox42.Text) Dim MCost_OP As Double = Double.Parse(TextBox57.Text) Dim MCost_OP_Waste As Double = Double.Parse(TextBox58.Text) Dim Rev As Double = Double.Parse(TextBox62.Text) Dim MCostAdd_OP As Double = Double.Parse(TextBox59.Text) Dim PCost As Double = Double.Parse(TextBox56.Text) Dim Mcost_OP_lvlStart1 As Double = Z - Mcost_OP_lvlStart - 1 Dim ExcelRow As Double = X * Y * Z With shXL_1C .Application.ScreenUpdating = False n = 1 For X1 = 1 To X * Y * Z .Cells(n + 1, 10) = "X" & .Cells(n + 1, 5).value & "Y" & .Cells(n + 1, 6).value & "Z" & .Cells(n + 1, 7).value n = n + 1 Next For i = 1 To ExcelRow Dim a As Double = Math.Round(.Cells(i + 1, 11).value, 2) Dim e1 As Double = .Cells(i + 1, 7).value If a >= COG_OP Then If e1 <= Mcost_OP_lvlStart Then Dim b As Double = (a) * Rev * Met_P * ton / 31.103</pre>	Summarizing and writing data in Excel for open-pit mining

```
            Dim c As Double = CLng(ton) * (MCost_OP +
            (MCostAdd_OP * (Z - e1 - Mcost_OP_lvlStart1)))
            Dim d As Double = CLng(ton) * PCost
            .Cells(i + 1, 9) = (b - c - d)
              Else
                 Dim b As Double = (a) * Rev * Met_P * ton / 31.103
                 Dim c As Double = CLng(ton) * (MCost_OP)
                 Dim d As Double = CLng(ton) * PCost
                 .Cells(i + 1, 9) = (b - c - d)
              End If
           Else
              If e1 <= Mcost_OP_lvlStart Then
             Dim c As Double = ton * (MCost_OP_Waste +
             (MCostAdd_OP * (Z - e1 - Mcost_OP_lvlStart1)))
              Dim d As Double = 0
             .Cells(i + 1, 9) = (-c - d)
            Else
                 Dim c As Double = ton * (MCost_OP_Waste)
                 Dim d As Double = 0
                 .Cells(i + 1, 9) = (-c - d)
              End If
           End If
        Next
     End With

     With shXL_1
        .Cells(6, 1) = "Metal Price"
        .Cells(6, 2) = Met_P

        .Cells(7, 1) = "OP Mining Cost"
        .Cells(7, 2) = MCost_OP

        .Cells(8, 1) = "Processing Cost"
        .Cells(8, 2) = PCost

        .Cells(9, 1) = "OP Ton/block"
        .Cells(9, 2) = ton

        .Cells(10, 1) = "Recovery"
        .Cells(10, 2) = Rev

     End With


wbXL.Application.ActiveWorkbook.Close(SaveChanges:=True)
     wbXL = Nothing
     AppXL.Quit()
     AppXL = Nothing
```

```
Dim AppXL As Excel.Application
Dim wbXL As Excel.Workbook
Dim wbsXL As Excel.Workbooks
Dim shXL_UG1, shXL_UG2, shXL_UG3, shXL_UG4, shXL_UG5, shXL_1 As
Excel.Worksheet
AppXL = CreateObject("excel.application")
wbsXL = AppXL.Workbooks
wbXL = wbsXL.Open(TextBox32.Text)
shXL_UG1 = wbXL.Worksheets("UG1")
shXL_UG2 = wbXL.Worksheets("UG2")
shXL_UG3 = wbXL.Worksheets("UG3")
shXL_UG4 = wbXL.Worksheets("UG4")
shXL_UG5 = wbXL.Worksheets("UG5")
shXL_1 = wbXL.Worksheets("Data")
shXL_UG2.Cells.Clear()
shXL_UG3.Cells.Clear()
shXL_UG4.Cells.Clear()
shXL_UG5.Cells.Clear()
Dim sLine, sLine1, sLine2 As String
Dim count As Double = 0
Dim BinariesCount As Double = 0
Dim SAG As Double
Dim n, m As Double
Dim X As Double = Double.Parse(TextBox36.Text)
Dim Y As Double = Double.Parse(TextBox37.Text)
Dim Z As Double = Double.Parse(TextBox38.Text)
Dim Stope_Size As Double = Double.Parse(TextBox64.Text) *
Double.Parse(TextBox65.Text) * Double.Parse(TextBox66.Text)
Dim Excel_LookUp As Double = X * Y * Z + 1
Dim COG_UG As Double = Double.Parse(TextBox44.Text)
Dim Met_P As Double = Double.Parse(TextBox55.Text)
Dim ton As Double = Double.Parse(TextBox39.Text) *
Double.Parse(TextBox40.Text) * Double.Parse(TextBox41.Text) *
Double.Parse(TextBox42.Text)
Dim Rev As Double = Double.Parse(TextBox62.Text)
Dim MCost_UG As Double = Double.Parse(TextBox61.Text)
Dim PCost As Double = Double.Parse(TextBox56.Text)
Dim ExcelRow As Double = X * Y * Z

With shXL_UG1
     .Application.ScreenUpdating = False
     n = 1
     For X1 = 1 To X
         For Y1 = 1 To Y
             For Z1 = 1 To Z
                 If .Cells(n + 1, 8).value < 0 Then
                    .Cells(n + 1, 8).value = 0
                 End If
                  .Cells(n + 1, 12) = .Cells(n + 1, 8).value
                 .Cells(n + 1, 9) = "X" & .Cells(n + 1, 5).value
                  & "Y" & .Cells(n + 1, 6).value & "Z" &
                  .Cells(n + 1, 7).value
                   .Cells(n + 1, 11) = "X" & .Cells(n + 1,
                  5).value & "Y" & .Cells(n + 1, 6).value & "Z"
                  & .Cells(n + 1, 7).value
                   n = n + 1
             Next
         Next
     Next
End With

Dim n1 As Double = 0
```

Summarizing and writing data in Excel for underground mining

```
n = 0
   For i = 1 To ExcelRow
       If shXL_UG1.Cells(i + 1, 8).value > 0 Then
          If shXL_UG1.Cells(i + 1, 5).value < X Then
            If shXL_UG1.Cells(i + 1, 6).value < Y Then
               If shXL_UG1.Cells(i + 1, 7).value < Z Then
                  n = n + 1
                  m = 0
                  For Z2 = 0 To 1
                      For X2 = 0 To 1
                          For Y2 = 0 To 1
                             shXL_UG2.Cells(n, 1).offset(0, m) =
                             "X" & shXL_UG1.Cells(i + 1,
                             5).value + X2 & "Y" &
                             shXL_UG1.Cells(i + 1, 6).value + Y2
                             & "Z" & shXL_UG1.Cells(i + 1,
                             7).value + Z2

                             shXL_UG3.Cells(n, 1).offset(0, m) =
                             "X" & shXL_UG1.Cells(i + 1,
                             5).value + X2 & "Y" &
                             shXL_UG1.Cells(i + 1, 6).value + Y2
                             & "Z" & shXL_UG1.Cells(i + 1,
                             7).value + Z2
                          m = m + 1
                          Next
                      Next
                 Next
                 shXL_UG2.Cells(n, 11) = shXL_UG1.Cells(i + 1,
                 7).value
                      End If
                End If
          End If
     End If
     Next

   For i = 1 To n
       For j = 1 To Stope_Size
           shXL_UG5.Cells(i, j) =
           shXL_UG5.Application.WorksheetFunction.Match(shXL_UG3
           .Cells(i, j), shXL_UG1.Range("$K$2:K" &
           Excel_LookUp), 0)
           Dim M1 As Double = shXL_UG5.Cells(i, j).value
           shXL_UG2.Cells(i, j) =
           shXL_UG2.Application.WorksheetFunction.Index(shXL_UG1
           .Range("$L$2:L" & Excel_LookUp), M1, 1)

        Next
  Next

  For i = 1 To n
      shXL_UG2.Cells(i, 10) =
      shXL_UG2.Application.WorksheetFunction.Sum(shXL_UG2.Range(
      shXL_UG2.Cells(i, 1), shXL_UG2.Cells(i, 8)))
      shXL_UG2.Cells(i, 10) = shXL_UG2.Cells(i, 10).value / 8

  Next

 For i = 1 To n
     If shXL_UG2.Cells(i, 10).value >= COG_UG Then
```

```
        shXL_UG2.Cells(i, 12).value = (shXL_UG2.Cells(i,
        10).value) * Stope_Size * CLng(ton / 31.103) * Rev *
        Met_P - (MCost_UG + PCost) * CLng(ton) * Stope_Size
        shXL_UG2.Cells(i, 13).value = shXL_UG3.Cells(i, 1).value
        shXL_UG2.Cells(i, 14).value = shXL_UG3.Cells(i, 8).value
    Else
        shXL_UG2.Cells(i, 12).value = -(MCost_UG) * CLng(ton) *
        Stope_Size - (PCost) * CLng(ton) * Stope_Size
        shXL_UG2.Cells(i, 13).value = shXL_UG3.Cells(i, 1).value
        shXL_UG2.Cells(i, 14).value = shXL_UG3.Cells(i, 8).value
      End If
 Next

 For i = 1 To n
     If shXL_UG2.Cells(i, 12).value > 0 Then
        shXL_UG4.Cells(n1 + 1, 1).value = shXL_UG2.Cells(i,
        12).value
        shXL_UG4.Cells(n1 + 1, 2).value = shXL_UG2.Cells(i,
        13).value
        shXL_UG4.Cells(n1 + 1, 3).value = shXL_UG2.Cells(i,
        14).value
        shXL_UG4.Cells(n1 + 1, 4).value = shXL_UG2.Cells(i,
        10).value
        shXL_UG4.Cells(n1 + 1, 5).value = shXL_UG2.Cells(i,
        11).value
         n1 = n1 + 1
     End If
 Next

 With shXL_1
      .Cells(11, 1) = "UG Mining Cost"
      .Cells(11, 2) = MCost_UG
      .Cells(12, 1) = "Stope Size"
      .Cells(12, 2) = Stope_Size
      .Cells(13, 1) = "Profitable stopes"
      .Cells(13, 2) = n
End With

wbXL.Application.ActiveWorkbook.Close(SaveChanges:=True)
wbXL = Nothing
AppXL.Quit()
AppXL = Nothing
```

APPENDIX B: TRANSITION POINT MODEL

``Dim FileName As String = TextBox33.Text & TextBox70.Text & ".lp" Dim Writer As StreamWriter = New StreamWriter(FileName, False) Dim AppXL As Excel.Application Dim wbXL As Excel.Workbook Dim wbsXL As Excel.Workbooks Dim shXL_1C, shXL_UG4 As Excel.Worksheet AppXL = CreateObject("excel.application") wbsXL = AppXL.Workbooks wbXL = wbsXL.Open(TextBox32.Text) shXL_1C = wbXL.Worksheets("Sheet1_C") shXL_UG4 = wbXL.Worksheets("UG4") Dim sLine, sLine1, sLine2 As String Dim count As Double = 0 Dim BinariesCount As Double = 0 Dim X As Double = Double.Parse(TextBox36.Text) Dim Y As Double = Double.Parse(TextBox37.Text) Dim Z As Double = Double.Parse(TextBox38.Text) Dim XYZ As Double = X * Y * Z Dim Adj_OP As Double = Double.Parse(TextBox63.Text) Dim CP As Double = Double.Parse(TextBox49.Text) With Writer .WriteLine("Maximize") For i = 1 To XYZ sLine = "" Dim a As Double = Math.Round(shXL_1C.Cells(i + 1, 9).value, 3) If a >= 0 Then sLine = sLine & "+" & a & shXL_1C.Cells(i + 1, 10).value Else sLine = sLine & a & shXL_1C.Cells(i + 1, 10).value End If .WriteLine(sLine) Next Dim n As Double = 1 While shXL_UG4.Cells(n, 5).value > 0 n = n + 1 End While For i = 1 To n - 1 sLine = "" sLine = sLine & "+" & Math.Round(shXL_UG4.Cells(i, 1).value, 3)```	Constructing MILP model to the LP File

```
    sLine = sLine & shXL_UG4.Cells(i, 2).value & ","
    sLine = sLine & shXL_UG4.Cells(i, 3).value
    n = n + 1
    .WriteLine(sLine)
Next

.WriteLine("Subject to")
For X1 = 2 To X - 1
    For Y1 = 2 To Y - 1
        For Z1 = Z - 1 To 1 Step -1
            count = count + 1
            sLine = ""
            sLine = sLine & "C" & count & ": "
            sLine = sLine & "+" & Adj_OP & "X" & X1 & "Y" &
             Y1 & "Z" & Z1
            sLine = sLine & "-" & "X" & X1 & "Y" & Y1 - 1 & "Z"
             & Z1 + 1
            sLine = sLine & "-" & "X" & X1 & "Y" & Y1 & "Z" &
             Z1 + 1
            sLine = sLine & "-" & "X" & X1 & "Y" & Y1 + 1 & "Z"
             & Z1 + 1
            sLine = sLine & "-" & "X" & X1 - 1 & "Y" & Y1 & "Z"
             & Z1 + 1
            sLine = sLine & "-" & "X" & X1 + 1 & "Y" & Y1 & "Z"
             & Z1 + 1
            sLine = sLine & "<=0"
            .WriteLine(sLine)
        Next
    Next
Next

For X1 = 1 To X - 2
    For Y1 = 1 To Y - 2
        For Z1 = Z - 2 To 1 Step -1
            count = count + 1
            sLine = ""
            sLine = sLine & "C" & count & ": "
            For k = 0 To 1
                For m = 0 To 1
                    For n = 0 To 1
                        sLine = sLine & "+X" & X1 + k & "Y" & Y1 +
                         m & "Z" & Z1 + n & "," & "X" & X1 + k + 1 &
                         "Y" & Y1 + m + 1 & "Z" & Z1 + n + 1
                    Next
                Next
            Next
    sLine = sLine & "<=1"
    .WriteLine(sLine)
        Next
    Next
```

```
Next

For Z1 = 1 To Z
   For X1 = 1 To X
      For Y1 = 1 To Y
         sLine = ""
         count = count + 1
         sLine = sLine & "C" & count & ": "
         sLine = sLine & "+H" & Z1 & "K1"
         sLine = sLine & "-X" & X1 & "Y" & Y1 & "Z" & Z1
         sLine = sLine & ">=0"
         .WriteLine(sLine)
      Next
   Next
Next
For Z1 = 1 To Z - 1
   For X1 = 1 To X - 1
      For Y1 = 1 To Y - 1
         sLine = ""
         count = count + 1
         sLine = sLine & "C" & count & ": "
         sLine = sLine & "+H" & Z1 & "K2"
          sLine = sLine & "-X" & X1 & "Y" & Y1 & "Z" & Z1 &
          "," & "X" & X1 + 1 & "Y" & Y1 + 1 & "Z" & Z1 + 1
         sLine = sLine & ">=0"
         .WriteLine(sLine)
      Next
   Next
Next

For Z1 = 1 To Z - 1
   For X1 = 1 To X
      For Y1 = 1 To Y
         sLine = ""
         count = count + 1
         sLine = sLine & "C" & count & ": "
         sLine = sLine & "+H" & Z1 + 1 & "K2"
         sLine = sLine & "-X" & X1 & "Y" & Y1 & "Z" & Z1 &
        "," & "X" & X1 + 1 & "Y" & Y1 + 1 & "Z" & Z1 + 1
         sLine = sLine & ">=0"
         .WriteLine(sLine)
      Next
   Next
Next

For Z1 = 1 To Z
   sLine = ""
   count = count + 1
   sLine = sLine & "C" & count & ": "
   sLine = sLine & "+H" & Z1 & "K1"
```

```
    sLine = sLine & "+H" & Z1 & "K2"
    sLine = sLine & "+H" & Z1
    sLine = sLine & "<=1"
    .WriteLine(sLine)
Next

For Z1 = Z To 2 Step -1
    sLine = ""
    count = count + 1
    sLine = sLine & "C" & count & ": "
    sLine = sLine & "+H" & Z1 & "K2"
    sLine = sLine & "-H" & Z1 - 1 & "K2"
    sLine = sLine & "<=0"
    .WriteLine(sLine)
Next

sLine1 = ""
For Z1 = Z To CP Step -1
    sLine = ""
    count = count + 1
    sLine = sLine & "C" & count & ": "
    If Z1 = Z Then
        sLine = sLine & "+" & CP & "H" & Z1 & "K1"
        For k = Z To Z1 - (CP - 1) Step -1
            sLine1 = sLine1 & "+H" & k
        Next
        sLine = sLine & sLine1 & ">=" & CP
        .WriteLine(sLine)
    Else
        sLine = sLine & "+" & CP & "H" & Z1 & "K1"
        sLine1 = sLine1 & "+H" & Z1 - (CP - 1)
        sLine = sLine & sLine1 & ">=" & CP
        .WriteLine(sLine)
    End If
Next

For Z1 = Z To CP Step -1
    sLine = ""
    sLine1 = ""
    count = count + 1
    sLine = sLine & "C" & count & ": "
    For k = Z To Z1 - (CP - 1) Step -1
        sLine1 = sLine1 & "+H" & k
    Next
    sLine = sLine & sLine1 & "<=" & CP
    .WriteLine(sLine)
Next

For Z1 = 1 To Z - 1
    For Y1 = 1 To Y
```

```
        count = count + 1
        sLine = ""
        sLine = sLine & "C" & count & ": "
        sLine = sLine & "+X" & 1 & "Y" & Y1 & "Z" & Z1
        sLine = sLine & "=0"
        .WriteLine(sLine)
    Next
Next

For Z1 = 1 To Z - 1
    For Y1 = 1 To Y
        count = count + 1
        sLine = ""
        sLine = sLine & "C" & count & ": "
        sLine = sLine & "+X" & X & "Y" & Y1 & "Z" & Z1
        sLine = sLine & "=0"
        .WriteLine(sLine)
    Next
Next

For Z1 = 1 To Z - 1
    For X1 = 2 To X - 1
        count = count + 1
        sLine = ""
        sLine = sLine & "C" & count & ": "
        sLine = sLine & "+X" & X1 & "Y" & 1 & "Z" & Z1
        sLine = sLine & "=0"
        .WriteLine(sLine)
    Next
Next

For Z1 = 1 To Z - 1
    For X1 = 2 To X - 1
        count = count + 1
        sLine = ""
        sLine = sLine & "C" & count & ": "
        sLine = sLine & "+X" & X1 & "Y" & Y & "Z" & Z1
        sLine = sLine & "=0"
        .WriteLine(sLine)
    Next
Next

.WriteLine("Binaries")
For X1 = 1 To X
    For Y1 = 1 To Y
        For Z1 = 1 To Z
            BinariesCount = BinariesCount + 1
            sLine = ""
            sLine = sLine & "X" & X1 & "Y" & Y1 & "Z" & Z1
            .WriteLine(sLine)
```

<pre> Next Next Next For X1 = 1 To X - 1 For Y1 = 1 To Y - 1 For Z1 = 1 To Z - 1 BinariesCount = BinariesCount + 1 sLine = "" sLine = sLine & "X" & X1 & "Y" & Y1 & "Z" & Z1 & "," & "X" & X1 + 1 & "Y" & Y1 + 1 & "Z" & Z1 + 1 .WriteLine(sLine) Next Next Next For Z1 = 1 To Z BinariesCount = BinariesCount + 1 sLine = "" sLine = sLine & "H" & Z1 .WriteLine(sLine) Next For k = 1 To 2 For Z1 = 1 To Z BinariesCount = BinariesCount + 1 sLine = "" sLine = sLine & "H" & Z1 & "K" & k .WriteLine(sLine) Next Next .WriteLine("End") End With Writer.Close() wbXL.Application.ActiveWorkbook.Close(SaveChanges:=True) wbXL = Nothing AppXL.Quit() AppXL = Nothing</pre>	

APPENDIX C: TRANSITION PERIOD MODEL

```
Dim FileName As String = TextBox33.Text & TextBox68.Text &
"_V4.lp"
Dim Writer As StreamWriter = New StreamWriter(FileName, False)
Dim AppXL As Excel.Application
Dim wbXL As Excel.Workbook
Dim wbsXL As Excel.Workbooks
Dim shXL_UG2, shXL1, shXL_UG4, shXL4 As Excel.Worksheet
AppXL = CreateObject("excel.application")
wbsXL = AppXL.Workbooks
wbXL = wbsXL.Open(TextBox32.Text)
shXL1 = wbXL.Worksheets("Sheet1")
shXL4 = wbXL.Worksheets("Sheet4")
shXL_UG4 = wbXL.Worksheets("UG4")
shXL_UG2 = wbXL.Worksheets("UG2")
Dim sLine, sLine1, sLine2 As String
Dim count As Double = 0
Dim BinariesCount As Double = 0
Dim X As Double = Double.Parse(TextBox36.Text)
Dim Y As Double = Double.Parse(TextBox37.Text)
Dim Z As Double = Double.Parse(TextBox38.Text)
Dim No_Blocks As Double = Double.Parse(TextBox77.Text)
Dim Adj_OP As Double = Double.Parse(TextBox63.Text)
Dim Prd As Double = Double.Parse(TextBox34.Text)
Dim D_R As Double = Double.Parse(TextBox35.Text) / 100
Dim Delay As Double = Double.Parse(TextBox54.Text)
Dim MCapMax_UG As Double = Double.Parse(TextBox46.Text)
Dim AL As Double = Double.Parse(TextBox48.Text)
Dim DevCost As Double = Double.Parse(TextBox78.Text) * 10 ^ 6
Dim MCapMax_OP As Double = Double.Parse(TextBox45.Text)
Dim PCapMax As Double = Double.Parse(TextBox47.Text)
Dim CP As Double = Double.Parse(TextBox49.Text)
Dim COG_OP As Double = Double.Parse(TextBox43.Text)
Dim ton As Double = Double.Parse(TextBox39.Text) *
Double.Parse(TextBox40.Text) * Double.Parse(TextBox41.Text) *
Double.Parse(TextBox42.Text)
Dim Stope_Size As Double = Double.Parse(TextBox64.Text) *
Double.Parse(TextBox65.Text) * Double.Parse(TextBox66.Text)
Dim Gmin_OP As Double = Double.Parse(TextBox51.Text)
Dim Gmax_OP As Double = Double.Parse(TextBox50.Text)
Dim COG_UG As Double = Double.Parse(TextBox44.Text)
Dim Gmin_UG As Double = Double.Parse(TextBox52.Text)
Dim Gmax_UG As Double = Double.Parse(TextBox53.Text)
```

With Writer .WriteLine("Maximize") For i = 1 To No_Blocks For T = 1 To Prd sLine = "" Dim a As Double = Math.Round(shXL1.Cells(i + 1, 9).value / ((1 + D_R) ^ (T - 1)), 3) If a >= 0 Then sLine = sLine & "+" & a & shXL1.Cells(i + 1, 10).value & "T" & T Else sLine = sLine & a & shXL1.Cells(i + 1, 10).value & "T" & T End If .WriteLine(sLine) Next Next	Constructing MILP model to the LP File Objective Function for open-pit mining
Dim n As Double = shXL_UG4.Application.WorksheetFunction.Count(shXL_UG4.Range("A:A")) For i = 1 To n - 1 For T = 1 To Prd sLine = "" sLine = sLine & "+" & Math.Round(shXL_UG4.Cells(i, 1).value / ((1 + D_R) ^ (T - 1)), 3) sLine = sLine & shXL_UG4.Cells(i, 2).value & "," sLine = sLine & shXL_UG4.Cells(i, 3).value & "T" & T .WriteLine(sLine) Next Next	Objective Function for underground mining
For T = 1 To Prd - Delay sLine = "" sLine = sLine & "-" & Math.Round(DevCost / ((1 + D_R) ^ (T + Delay - 1)), 3) & "D" & T WriteLine(sLine) Next .WriteLine("Subject to") For T = 1 To Prd For i = 2 To No_Blocks + 1 For j = 2 To 6 If String.IsNullOrEmpty(shXL4.Cells(i, j).value) Then Else count = count + 1 sLine = ""	Objective Function for development cost

```
                    sLine = sLine & "C_Slope" & count & ": "
                    sLine = sLine & "+" & shXL4.Cells(i, 1).value & "T" &
T
                    For T1 = 1 To T
                          sLine = sLine & "-" & shXL4.Cells(i, j).value &
"T" & T1
                    Next
                          sLine = sLine & "<=0"
                          .WriteLine(sLine)
                   End If
             Next
      Next
Next


If Prd > 1 Then
    For i = 1 To No_Blocks
          count = count + 1
          sLine = ""
          sLine = sLine & "C_OP_OneTimeProd" & count & ": "
          For T = 1 To Prd
                sLine = sLine & "+" & "X" & shXL1.Cells(i + 1, 5).value
               & "Y" & shXL1.Cells(i + 1, 6).value & "Z" &
               shXL1.Cells(i + 1, 7).value & "T" & T
          Next
          sLine = sLine & "<=1"
          WriteLine(sLine)
     Next
End If

n =
shXL_UG4.Application.WorksheetFunction.Count(shXL_UG4.Rang
e("A:A"))

For i = 1 To n
      sLine = ""
      count = count + 1
      sLine = sLine & "C_UG_OneTimeProd_DisStp" & count & ": "
       For T = 1 To Prd
            sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value & "Y"
            & shXL_UG4.Cells(i, 10).value & "Z" & shXL_UG4.Cells(i,
            11).value & "," & "X" & shXL_UG4.Cells(i, 9).value + 1 &
            "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" &
            shXL_UG4.Cells(i, 11).value + 1 & "T" & T

            sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value + 1 &
            "Y" & shXL_UG4.Cells(i, 10).value - 1 & "Z" &
            shXL_UG4.Cells(i, 11).value & "," & "X" &
            shXL_UG4.Cells(i, 9).value + 2 & "Y" & shXL_UG4.Cells(i,
```

10).value & "Z" & shXL_UG4.Cells(i, 11).value + 1 & "T" & T sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value - 1 & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value & "," & "X" & shXL_UG4.Cells(i, 9).value & "Y" & shXL_UG4.Cells(i, 10).value + 2 & "Z" & shXL_UG4.Cells(i, 11).value + 1 & "T" & T sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value & "," & "X" & shXL_UG4.Cells(i, 9).value + 1 & "Y" & shXL_UG4.Cells(i, 10).value + 2 & "Z" & shXL_UG4.Cells(i, 11).value + 1 & "T" & T sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value + 1 & "," & "X" & shXL_UG4.Cells(i, 9).value + 1 & "Y" & shXL_UG4.Cells(i, 10).value + 2 & "Z" & shXL_UG4.Cells(i, 11).value + 2 & "T" & T sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value & "Y" & shXL_UG4.Cells(i, 10).value & "Z" & shXL_UG4.Cells(i, 11).value + 1 & "," & "X" & shXL_UG4.Cells(i, 9).value + 1 & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value + 2 & "T" & T sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value + 1 & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value + 1 & "," & "X" & shXL_UG4.Cells(i, 9).value + 2 & "Y" & shXL_UG4.Cells(i, 10).value + 2 & "Z" & shXL_UG4.Cells(i, 11).value + 2 & "T" & T sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value + 1 & "Y" & shXL_UG4.Cells(i, 10).value & "Z" & shXL_UG4.Cells(i, 11).value + 1 & "," & "X" & shXL_UG4.Cells(i, 9).value + 2 & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value + 2 & "T" & T sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value + 1 & "Y" & shXL_UG4.Cells(i, 10).value & "Z" & shXL_UG4.Cells(i, 11).value & "," & "X" & shXL_UG4.Cells(i, 9).value + 2 & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value + 1 & "T" & T	

```
        sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value + 1 &
        "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" &
        shXL_UG4.Cells(i, 11).value & "," & "X" &
        shXL_UG4.Cells(i, 9).value + 2 & "Y" & shXL_UG4.Cells(i,
        10).value + 2 & "Z" & shXL_UG4.Cells(i, 11).value + 1 &
        "T" & T

        sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value & "Y"
        & shXL_UG4.Cells(i, 10).value + 1 & "Z" &
        shXL_UG4.Cells(i, 11).value + 1 & "," & "X" &
        shXL_UG4.Cells(i, 9).value + 1 & "Y" & shXL_UG4.Cells(i,
        10).value + 2 & "Z" & shXL_UG4.Cells(i, 11).value + 1 &
        "T" & T

        sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value + 1 &
        "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" &
        shXL_UG4.Cells(i, 11).value + 1 & "," & "X" &
        shXL_UG4.Cells(i, 9).value + 2 & "Y" & shXL_UG4.Cells(i,
        10).value + 2 & "Z" & shXL_UG4.Cells(i, 11).value + 1 &
        "T" & T

        sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value & "Y"
        & shXL_UG4.Cells(i, 10).value & "Z" & shXL_UG4.Cells(i,
        11).value + 1 & "," & "X" & shXL_UG4.Cells(i, 9).value + 1
        & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" &
        shXL_UG4.Cells(i, 11).value + 1 & "T" & T

        sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value + 1 &
        "Y" & shXL_UG4.Cells(i, 10).value & "Z" &
        shXL_UG4.Cells(i, 11).value + 1 & "," & "X" &
        shXL_UG4.Cells(i, 9).value + 2 & "Y" & shXL_UG4.Cells(i,
        10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value + 1 &
        "T" & T
    Next
    sLine = sLine & "<=1"
     .WriteLine(sLine)
    Next

n =
shXL_UG4.Application.WorksheetFunction.Count(shXL_UG4.Rang
e("A:A"))
For i = 1 To n
      count = count + 1
      sLine = sLine & "C_UG_OneTimeProd_DisStp" & count & ": "
       sLine = ""
       For k = 0 To 1
              For m = 0 To 1
                     For n = 0 To 1
                            For T = 1 To Prd
```

```
                        sLine = sLine & "+X" & shXL_UG4.Cells(i,
                        9).value + k & "Y" & shXL_UG4.Cells(i,
                        10).value + m & "Z" & shXL_UG4.Cells(i,
                        11).value + n & "," & "X" & shXL_UG4.Cells(i,
                        9).value + k + 1 & "Y" & shXL_UG4.Cells(i,
                        10).value + m + 1 & "Z" & shXL_UG4.Cells(i,
                        11).value + n + 1 & "T" & T
                    Next
                Next
            Next
        Next
    sLine = sLine & "<=1"
     WriteLine(sLine)
    Next

n =
shXL_UG4.Application.WorksheetFunction.Count(shXL_UG4.Rang
e("A:A"))

For T = 1 To Prd
        For i = 1 To n
        count = count + 1
                sLine = ""
                sLine = sLine & "C_NonConAdjStp_X" & count & ": "
                     For j = 0 To 2 Step 2
                           sLine = sLine & "+X" & shXL_UG4.Cells(i,
                          9).value + j & "Y" & shXL_UG4.Cells(i, 10).value
                          & "Z" & shXL_UG4.Cells(i, 11).value & "," & "X"
                          & shXL_UG4.Cells(i, 9).value + 1 + j & "Y" &
                          shXL_UG4.Cells(i, 10).value + 1 & "Z" &
                          shXL_UG4.Cells(i, 11).value + 1 & "T" & T
              Next
         sLine = sLine & "<=1"
        .WriteLine(sLine)
        Next
Next

For T = 1 To Prd
       For i = 1 To n
             count = count + 1
              sLine = ""
               sLine = sLine & "C_NonConAdjStp_Y" & count & ": "
               For j = 0 To 2 Step 2
                      sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value
                      & "Y" & shXL_UG4.Cells(i, 10).value + j & "Z" &
                      shXL_UG4.Cells(i, 11).value & "," & "X" &
                      shXL_UG4.Cells(i, 9).value + 1 & "Y" &
                      shXL_UG4.Cells(i, 10).value + 1 + j & "Z" &
                      shXL_UG4.Cells(i, 11).value + 1 & "T" & T
                Next
```

```
            sLine = sLine & "<=1"
            .WriteLine(sLine)
        Next
Next

n =
shXL_UG4.Application.WorksheetFunction.Count(shXL_UG4.Rang
e("A:A"))

For i = 1 To n
    count = count + 1
    sLine = ""
    sLine = sLine & "C_StpOffset" & count & ": "
    For T = 1 To Prd
        For j = 0 To 2 Step 2
            sLine = sLine & "+X" & shXL_UG4.Cells(i, 9).value
            & "Y" & shXL_UG4.Cells(i, 10).value & "Z" &
            shXL_UG4.Cells(i, 11).value + j & "," & "X" &
            shXL_UG4.Cells(i, 9).value + 1 & "Y" &
            shXL_UG4.Cells(i, 10).value + 1 & "Z" &
            shXL_UG4.Cells(i, 11).value + 1 + j & "T" & T
        Next
    Next
    sLine = sLine & "<=1"
    .WriteLine(sLine)
Next

n =
shXL_UG4.Application.WorksheetFunction.Count(shXL_UG4.Rang
e("A:A"))
For T = 1 To Prd
    For i = 1 To n
        sLine = ""
        count = count + 1
        sLine = sLine & "C_NonConAdjVerticalStp" & count & ":
        "
        sLine = sLine & "+" & "X" & shXL_UG4.Cells(i, 9).value
        & "Y" & shXL_UG4.Cells(i, 10).value & "Z" &
        shXL_UG4.Cells(i, 11).value & "," & "X" &
        shXL_UG4.Cells(i, 9).value + 1 & "Y" &
        shXL_UG4.Cells(i, 10).value + 1 & "Z" &
        shXL_UG4.Cells(i, 11).value + 1 & "T" & T

        Dim j As Integer = 2
        sLine = sLine & "+" & "X" & shXL_UG4.Cells(i, 9).value
        + 1 & "Y" & shXL_UG4.Cells(i, 10).value & "Z" &
        shXL_UG4.Cells(i, 11).value + j & "," & "X" &
        shXL_UG4.Cells(i, 9).value + 2 & "Y" &
        shXL_UG4.Cells(i, 10).value + 1 & "Z" &
        shXL_UG4.Cells(i, 11).value + 1 + j & "T" & T
```

```
sLine = sLine & "+" & "X" & shXL_UG4.Cells(i, 9).value -
1 & "Y" & shXL_UG4.Cells(i, 10).value & "Z" &
shXL_UG4.Cells(i, 11).value + j & "," & "X" &
shXL_UG4.Cells(i, 9).value & "Y" & shXL_UG4.Cells(i,
10).value + 1 & "Z" & shXL_UG4.Cells(i, 11).value + 1 + j
& "T" & T

        sLine = sLine & "+" & "X" & shXL_UG4.Cells(i,
9).value + 1 & "Y" & shXL_UG4.Cells(i, 10).value + 1 &
"Z" & shXL_UG4.Cells(i, 11).value + j & "," & "X" &
shXL_UG4.Cells(i, 9).value + 2 & "Y" &
shXL_UG4.Cells(i, 10).value + 2 & "Z" &
shXL_UG4.Cells(i, 11).value + 1 + j & "T" & T

sLine = sLine & "+" & "X" & shXL_UG4.Cells(i, 9).value
+ 1 & "Y" & shXL_UG4.Cells(i, 10).value - 1 & "Z" &
shXL_UG4.Cells(i, 11).value + j & "," & "X" &
shXL_UG4.Cells(i, 9).value + 2 & "Y" &
shXL_UG4.Cells(i, 10).value & "Z" & shXL_UG4.Cells(i,
11).value + 1 + j & "T" & T

sLine = sLine & "+" & "X" & shXL_UG4.Cells(i, 9).value -
1 & "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" &
shXL_UG4.Cells(i, 11).value + j & "," & "X" &
shXL_UG4.Cells(i, 9).value & "Y" & shXL_UG4.Cells(i,
10).value + 2 & "Z" & shXL_UG4.Cells(i, 11).value + 1 + j
& "T" & T

 sLine = sLine & "+" & "X" & shXL_UG4.Cells(i, 9).value -
1 & "Y" & shXL_UG4.Cells(i, 10).value - 1 & "Z" &
shXL_UG4.Cells(i, 11).value + j & "," & "X" &
shXL_UG4.Cells(i, 9).value & "Y" & shXL_UG4.Cells(i,
10).value & "Z" & shXL_UG4.Cells(i, 11).value + 1 + j &
"T" & T

sLine = sLine & "+" & "X" & shXL_UG4.Cells(i, 9).value &
  "Y" & shXL_UG4.Cells(i, 10).value + 1 & "Z" &
  shXL_UG4.Cells(i, 11).value + j & "," & "X" &
  shXL_UG4.Cells(i, 9).value + 1 & "Y" &
  shXL_UG4.Cells(i, 10).value + 2 & "Z" &
  shXL_UG4.Cells(i, 11).value + 1 + j & "T" & T

sLine = sLine & "+" & "X" & shXL_UG4.Cells(i, 9).value &
  "Y" & shXL_UG4.Cells(i, 10).value - 1 & "Z" &
  shXL_UG4.Cells(i, 11).value + j & "," & "X" &
  shXL_UG4.Cells(i, 9).value + 1 & "Y" &
  shXL_UG4.Cells(i, 10).value & "Z" & shXL_UG4.Cells(i,
  11).value + 1 + j & "T" & T
```

```
          sLine = sLine & "<=1"
          .WriteLine(sLine)
          Next
Next

For i = 1 To n
     For j = 1 To n
           If shXL_UG4.Cells(j, 11).value = shXL_UG4.Cells(i,
            11).value + 1 Or shXL_UG4.Cells(j, 11).value =
            shXL_UG4.Cells(i, 11).value - 1 Then
            sLine = ""
            count = count + 1
            sLine = sLine & "C_SameLvlExtraction" & count & ": "
                  For T = 1 To Prd
                       sLine = sLine & "+" & "X" & shXL_UG4.Cells(i,
                        9).value & "Y" & shXL_UG4.Cells(i, 10).value &
                        "Z" & shXL_UG4.Cells(i, 11).value & "," & "X" &
                        shXL_UG4.Cells(i, 9).value + 1 & "Y" &
                        shXL_UG4.Cells(i, 10).value + 1 & "Z" &
                        shXL_UG4.Cells(i, 11).value + 1 & "T" & T
                  Next
                   For T1 = 1 To Prd
                         sLine = sLine & "+" & "X" & shXL_UG4.Cells(j,
                        9).value & "Y" & shXL_UG4.Cells(j, 10).value &
                        "Z" & shXL_UG4.Cells(j, 11).value & "," & "X" &
                        shXL_UG4.Cells(j, 9).value + 1 & "Y" &
                        shXL_UG4.Cells(j, 10).value + 1 & "Z" &
                        shXL_UG4.Cells(j, 11).value + 1 & "T" & T1
                   Next

            sLine = sLine & "<=" & 1
            .WriteLine(sLine)
    End If
     Next
Next
For Z1 = 1 To Z
     For T = 1 To Prd
           For X1 = 1 To X
                 For Y1 = 1 To Y
                       sLine = ""
                       count = count + 1
                       sLine = sLine & "C_OnlyOneMinMethod_OP" &
                        count & ": "
                       sLine = sLine & "+H" & Z1 & "K1" & "T" & T
                       sLine = sLine & "-X" & X1 & "Y" & Y1 & "Z" &
                        Z1 & "T" & T
                       sLine = sLine & ">=0"
                       .WriteLine(sLine)
                    Next
             Next
```

```
        Next
Next

For Z1 = 1 To Z - 1
        For T = 1 To Prd
                For X1 = 1 To X - 1
                        For Y1 = 1 To Y - 1
                                sLine = ""
                                count = count + 1
                                sLine = sLine & "C_OnlyOneMinMethod_UG" &
count & ": "
                                sLine = sLine & "+H" & Z1 & "K2" & "T" & T
                                sLine = sLine & "-X" & X1 & "Y" & Y1 & "Z" &
                                Z1 & "," & "X" & X1 + 1 & "Y" & Y1 + 1 & "Z"
                                & Z1 + 1 & "T" & T
                                sLine = sLine & ">=0"
                                .WriteLine(sLine)
                        Next
                Next
        Next
Next
For Z1 = 1 To Z - 1
        For T = 1 To Prd
                For X1 = 1 To X
                        For Y1 = 1 To Y
                                sLine = ""
                                count = count + 1
                                sLine = sLine & "C_OnlyOneMinMethod_UG" &
count & ": "
                                sLine = sLine & "+H" & Z1 + 1 & "K2" & "T" & T
                                sLine = sLine & "-X" & X1 & "Y" & Y1 & "Z" &
                                Z1 & "," & "X" & X1 + 1 & "Y" & Y1 + 1 & "Z"
                                & Z1 + 1 & "T" & T
                                sLine = sLine & ">=0"
                                .WriteLine(sLine)
                        Next
                Next
        Next
Next

For Z1 = 1 To Z
        For T = 1 To Prd
                        sLine = ""
                        count = count + 1
                        sLine = sLine & "C_OnlyOneMinMethod1" & count &
": "
                        sLine = sLine & "+M" & Z1 & "K1"
                        sLine = sLine & "-H" & Z1 & "K1" & "T" & T
                        sLine = sLine & ">=0"
                        .WriteLine(sLine)
```

```
        Next
Next

For Z1 = 1 To Z
      For T = 1 To Prd
                 sLine = ""
                 count = count + 1
                 sLine = sLine & "C_OnlyOneMinMethod2" & count &
": "
                 sLine = sLine & "+M" & Z1 & "K2"
                 sLine = sLine & "-H" & Z1 & "K2" & "T" & T
                 sLine = sLine & ">=0"
                 .WriteLine(sLine)
       Next
Next

For Z1 = 1 To Z
              sLine = ""
              count = count + 1
              sLine = sLine & "C_OnlyOneMinMethod3" & count & ": "
              sLine = sLine & "+M" & Z1 & "K1"
              sLine = sLine & "+M" & Z1 & "K2"
              sLine = sLine & "+H" & Z1
              sLine = sLine & "<=1"
              .WriteLine(sLine)

Next

          sLine1 = ""
          For Z1 = Z To 1 Step -1
             sLine = ""
             count = count + 1
             sLine = sLine & "C_CPOPContiguous" & count & ": "
             If Z1 = Z Then

                sLine = sLine & "+" & CP & "M" & Z1 & "K1"

                For k = Z To Z1 - CP + 1 Step -1
                   sLine1 = sLine1 & "+H" & k
                Next
                sLine = sLine & sLine1 & ">=" & CP
                .WriteLine(sLine)
                   Else

                sLine = sLine & "+" & CP & "M" & Z1 & "K1"

                If Z1 - CP + 1 > 0 Then
                   sLine1 = sLine1 & "+H" & Z1 - CP + 1
                Else
                   Exit For
```

```
            End If
            sLine = sLine & sLine1 & ">=" & CP
            .WriteLine(sLine)
        End If
    Next

    For Z1 = 1 To Z
        sLine = ""
        sLine1 = ""
        sLine2 = ""
        count = count + 1
        sLine = sLine & "C_CP" & count & ": "
        sLine = sLine & "+" & CP - 1 & "H" & Z1
        For Z2 = 1 To Z1 - 1
            sLine1 = sLine1 & "-H" & Z2
        Next
        For Z3 = Z1 + 1 To Z1 + (CP - 1)
            sLine2 = sLine2 & "-H" & Z3
        Next
        sLine = sLine & sLine1 & sLine2 & "<=0"
        .WriteLine(sLine)
    Next

    For T = 1 To Prd
        For Z1 = 1 To Z
            sLine = ""
            count = count + 1
            sLine = sLine & "C_OPContiguousProd" & count & ": "
            For T1 = 1 To T
                For Z2 = 0 To Z1 - 1
                    sLine = sLine & "+H" & Z - Z2 & "K1" & "T" &
T1
                Next
            Next

            If Z - Z1 > 0 Then
                sLine = sLine & "-" & T * (Z1) & "H" & Z - Z1 &
                "K1" & "T" & T
            Else
                Exit For
                Exit For
            End If

            sLine = sLine & ">=0"
            .WriteLine(sLine)
        Next
    Next

    For Z1 = 1 To Z
        For T = 1 To Prd - 1
```

```
            count = count + 1
            sLine = ""
            sLine = sLine & "C_UGContiguousProd1" & count & ":
"
            sLine = sLine & "+H" & Z1 & "K2" & "T" & T
            sLine = sLine & "-H" & Z1 & "K2" & "T" & T + 1
            sLine = sLine & "<=0"
            .WriteLine(sLine)
        Next
    Next

    For Z1 = 0 To Z - 2
        For T = 1 To Prd
            count = count + 1
            sLine = ""
            sLine = sLine & "C_UGOPCPContguous" & count & ":
"
            sLine = sLine & "+H" & Z - Z1 - 1 & "K2" & "T" & T
            sLine = sLine & "-H" & Z - Z1 & "K2" & "T" & T
            'sLine = sLine & "-H" & Z - Z1 & "K1" & "T" & T
            sLine = sLine & "-H" & Z - Z1
            sLine = sLine & "<=0"
            .WriteLine(sLine)
        Next
    Next

    count = count + 1
    sLine = ""
    sLine1 = ""
    sLine = sLine & "C_Dev1" & count & ": "
    For T = 1 To Prd - Delay
        sLine = sLine & "+D" & T
    Next
    sLine = sLine & "<=1"
    .WriteLine(sLine)

    For T = 1 + Delay To Prd
        count = count + 1
        sLine = ""
        sLine1 = ""
        sLine = sLine & "C_Dev2" & count & ": "
        For Z1 = 1 To Z
            sLine = sLine & "+H" & Z1 & "K2" & "T" & T
        Next
        For T1 = 1 To T - Delay
            If (AL * (T - Delay - T1 + 1)) <= Z - CP Then
                sLine1 = sLine1 & "-" & (AL * (T - Delay - T1 + 1))
                & "D" & T1
            Else
                sLine1 = sLine1 & "-" & Z - CP & "D" & T1
```

```
            End If
        Next

        sLine = sLine & sLine1 & "<=0"
        .WriteLine(sLine)
    Next

    For T = 1 To Prd
        count = count + 1
        sLine = ""
        sLine1 = ""
        sLine = sLine & "C_Dev3" & count & ": "
        For Z1 = 1 To Z
            sLine = sLine & "+H" & Z1 & "K1" & "T" & T
        Next
        sLine1 = sLine1 & "+" & Z & "D" & T
        sLine = sLine & sLine1 & "<=" & Z
        .WriteLine(sLine)
    Next

    For T = 1 To Prd
        For Z1 = 1 To Z
            count = count + 1
            sLine = ""
            sLine1 = ""
            sLine = sLine & "C_OneMinMethodEachPrd" & count
& ": "
            sLine = sLine & "+" & Z & "H" & Z1 & "K1" & "T" &
T
            For Z2 = 1 To Z
                sLine1 = sLine1 & "+H" & Z2 & "K2" & "T" & T
            Next
            sLine = sLine & sLine1 & "<=" & Z
            .WriteLine(sLine)
        Next
    Next
    For T = 1 To Prd
        count = count + 1
        sLine = ""
        sLine1 = ""
        sLine = sLine & "C_MinCapOP" & count & ": "
        For i = 1 To No_Blocks
            sLine1 = sLine1 & "+" & ton & shXL1.Cells(i + 1,
           10).value & "T" & T
        Next
        sLine = sLine & sLine1 & "<=" & MCapMax_OP
        .WriteLine(sLine)
Next
```

```
n =
shXL_UG4.Application.WorksheetFunction.Count(shXL_UG4.Rang
e("A:A"))
        For T = 1 To Prd
            sLine = ""
            For i = 1 To n - 1
                sLine = sLine & "+" & ton * Stope_Size
                sLine = sLine & shXL_UG4.Cells(i, 2).value & ","
                sLine = sLine & shXL_UG4.Cells(i, 3).value & "T" & T
            Next
            sLine = sLine & "<=" & MCapMax_UG
            .WriteLine(sLine)
        Next

        For T = 1 To Prd
            count = count + 1
            sLine = ""
            sLine = sLine & "C_ProCapOP" & count & ": "
            For n = 1 To No_Blocks
                If shXL1.Cells(n + 1, 8).value >= Gmin_OP Then
                    sLine = sLine & "+" & ton
                    sLine = sLine & shXL1.Cells(n + 1, 10).value & "T"
& T
                End If
            Next
            sLine = sLine & "<=" & PCapMax
            .WriteLine(sLine)
        Next

        Dim n1 As Double = 1
        While shXL_UG4.Cells(n1, 1).value > 0
            n1 = n1 + 1
        End While

        For T = 1 To Prd
            count = count + 1
            sLine = ""
            sLine = sLine & "C_GrdOPMin" & count & ": "
            For n = 1 To No_Blocks
                If shXL1.Cells(n + 1, 8).value >= Gmin_OP Then
                    Dim gradeB_diff1 As Double = ((shXL1.Cells(n + 1,
                    8).value) - Gmin_OP) * ton

                    If gradeB_diff1 >= 0 Then
                        sLine = sLine & "+" & gradeB_diff1
                        sLine = sLine & shXL1.Cells(n + 1, 10).value &
"T" & T
                    Else
                        sLine = sLine & gradeB_diff1
```

```
                    sLine = sLine & shXL1.Cells(n + 1, 10).value &
"T" & T
                End If
            End If
        Next

        For m = 1 To n1 - 1
            If shXL_UG4.Cells(m, 4).value >= Gmin_UG Then
                Dim gradeDiff3 As Double = ((shXL_UG4.Cells(m,
                4).value) - Gmin_UG) * ton * Stope_Size
                If gradeDiff3 >= 0 Then
                    sLine = sLine & "+" & gradeDiff3
                    sLine = sLine & shXL_UG4.Cells(m, 2).value & ","
                & shXL_UG4.Cells(m, 3).value & "T" & T
                Else
                    sLine = sLine & gradeDiff3
                    sLine = sLine & shXL_UG4.Cells(m, 2).value & ","
                & shXL_UG4.Cells(m, 3).value & "T" & T
                End If
            End If
        Next

        sLine = sLine & ">=0"
        .WriteLine(sLine)
    Next

    n1 = 1
    While shXL_UG4.Cells(n1, 1).value > 0
        n1 = n1 + 1
    End While

    For T = 1 To Prd
        count = count + 1
        sLine = ""
        sLine = sLine & "C_GrdOPMAX" & count & ": "
        For n = 1 To No_Blocks
            If shXL1.Cells(n + 1, 8).value >= Gmin_OP Then
                  Dim gradeB_diff1 As Double = ((shXL1.Cells(n + 1,
                  8).value) - Gmax_OP) * ton

                If gradeB_diff1 >= 0 Then
                    sLine = sLine & "+" & gradeB_diff1
                    sLine = sLine & shXL1.Cells(n + 1, 10).value &
"T" & T
                Else
                    sLine = sLine & gradeB_diff1
                    sLine = sLine & shXL1.Cells(n + 1, 10).value &
"T" & T
                End If
            End If
```

```
            Next

            For m = 1 To n1 - 1
                If shXL_UG4.Cells(m, 4).value >= Gmin_UG Then
                    Dim gradeDiff3 As Double = ((shXL_UG4.Cells(m,
                    4).value) - Gmax_UG) * ton * Stope_Size
                    If gradeDiff3 >= 0 Then
                        sLine = sLine & "+" & gradeDiff3
                        sLine = sLine & shXL_UG4.Cells(m, 2).value & ","
                         & shXL_UG4.Cells(m, 3).value & "T" & T
                    Else
                        sLine = sLine & gradeDiff3
                        sLine = sLine & shXL_UG4.Cells(m, 2).value & ","
                         & shXL_UG4.Cells(m, 3).value & "T" & T
                    End If
                End If
            Next

            sLine = sLine & "<=0"
            .WriteLine(sLine)
        Next

.WriteLine("Binaries")

        For i = 1 To No_Blocks
            For T = 1 To Prd
                BinariesCount = BinariesCount + 1
                sLine = ""
                sLine = sLine & shXL1.Cells(i + 1, 10).value & "T" & T

                .WriteLine(sLine)
            Next
        Next

        For T = 1 To Prd
            For X1 = 1 To X - 1
                For Y1 = 1 To Y - 1
                    For Z1 = 1 To Z - 1
                        BinariesCount = BinariesCount + 1
                        sLine = ""
                        sLine = sLine & "X" & X1 & "Y" & Y1 & "Z" &
                      Z1 & "," & "X" & X1 + 1 & "Y" & Y1 + 1 & "Z" &
                      Z1 + 1 & "T" & T
                        .WriteLine(sLine)
                    Next
                Next
            Next
        Next

  For Z1 = 1 To Z
```

```
            BinariesCount = BinariesCount + 1
            sLine = ""
            sLine = sLine & "H" & Z1
            .WriteLine(sLine)
          Next

          For T = 1 To Prd
            For k = 1 To 2
              For Z1 = 1 To Z
                BinariesCount = BinariesCount + 1
                sLine = ""
                sLine = sLine & "H" & Z1 & "K" & k & "T" & T
                .WriteLine(sLine)
              Next
            Next
          Next

          For k = 1 To 2
            For Z1 = 1 To Z
              BinariesCount = BinariesCount + 1
              sLine = ""
              sLine = sLine & "M" & Z1 & "K" & k
              .WriteLine(sLine)
            Next
          Next

For T = 1 To Prd - Delay
            BinariesCount = BinariesCount + 1
            sLine = ""
            sLine = sLine & "D" & T
            .WriteLine(sLine)
          Next
          .WriteLine("END")
        End With

        Writer.Close()

wbXL.Application.ActiveWorkbook.Close(SaveChanges:=True)
        wbXL = Nothing
        AppXL.Quit()
        AppXL = Nothing
    End Sub
```

APPENDIX D: HIERARCHICAL CLUSTERING ALGORITHM IN MATLAB CODE

```
clc
clear
tic
format shortEng
format compact
File = load ('ULP_Cluster_20.txt');
set(0,'RecursionLimit',1000000)

rows =13555;
x = File(:,1);
y = File(:,2);
z = File(:,3);
x1 = File(:,4);
y1 = File(:,5);
z1 = File(:,6);
coordinate = [x y z];
block_ID = [File(:,4) File(:,5) File(:,6)];
sort_A = [coordinate block_ID];
Y = pdist(block_ID);
Y1 = squareform(Y);
grade = File(:,7);


for i = 1:rows
Slope_initial = [File(i,4) File(i,5) File(i,6)];
   connection_11 = [x1(i,1) y1(i,1) z1(i,1)+1];
   connection_12 = [x1(i,1)+1 y1(i,1) z1(i,1)+1];
   connection_13 = [x1(i,1)-1 y1(i,1) z1(i,1)+1];
   connection_14 = [x1(i,1) y1(i,1)+1 z1(i,1)+1];
   connection_15 = [x1(i,1) y1(i,1)-1 z1(i,1)+1];
   connection_2 = [x1(i,1)+1 y1(i,1) z1(i,1)];
   connection_3 = [x1(i,1)-1 y1(i,1) z1(i,1)];
   connection_4 = [x1(i,1) y1(i,1)+1 z1(i,1)];
   connection_5 = [x1(i,1) y1(i,1)-1 z1(i,1)];
   connection_6 = [x1(i,1)+1 y1(i,1)+1 z1(i,1)];
   connection_7 = [x1(i,1)+1 y1(i,1)-1 z1(i,1)];
   connection_8 = [x1(i,1)-11 y1(i,1)+1 z1(i,1)];
   connection_9 = [x1(i,1)-11 y1(i,1)-1 z1(i,1)];
   connection_10 = [z1(i,1)];

   for j = 1:rows
      block_array = [File(j,4) File(j,5) File(j,6)];
```

`` if i == j Slope1(i,j) = 0; Grade3(i,j) = 0; else if block_array == connection_2 Slope1(i,j) = 0.5; else if block_array == connection_3 Slope1(i,j) = 0.5; else if block_array == connection_4 Slope1 (i,j) = 0.5; else if block_array == connection_5 Slope1(i,j) = 0.5; else if block_array == connection_6 Slope1(i,j) = 0.5; else if block_array == connection_7 Slope1(i,j) = 0.5; else if block_array == connection_8 Slope1(i,j) = 0.5; else if block_array == connection_9 Slope1(i,j) = 0.5; else Slope1(i,j) = 1; end end end end end end end end if block_array == connection_11 Slope1(i,j) = Slope1(i,j)*0.75; else if block_array == connection_12 Slope1(i,j) = Slope1(i,j)*0.75; else if block_array == connection_13 Slope1(i,j) = Slope1(i,j)*0.75; else if block_array == connection_14 Slope1(i,j) = Slope1(i,j)*0.75; else ```	

```
            if block_array == connection_15
               Slope1(i,j) = Slope1(i,j)*0.75;
            else
               Slope1(i,j) = Slope1(i,j)*1;
            end
            end
            end
            end
            end

            if block_array == connection_10
              Slope1(i,j) = Slope1(i,j)*0.25;
            else
              Slope1(i,j) = Slope1(i,j)*10;
            end

            if grade(i,1) == grade(j,1)
               Grade3(i,j) = 0.01;
            else
Grade3(i,j) = (grade(i,1) - grade(j,1))^2;
            end

      end
   end
end

Cluster_Matrix = Grade3.* Y1.* Slope1;

for i = 1:rows
   for j = 1:rows
      if i == j
         Cluster_Matrix (i,j) = 0;
      end
   end
end


T5   = clusterdata(Cluster_Matrix,'Cutoff',0.75);

toc
```