Production Scheduling and Boundary Optimization in Block Caving Mines under Geologic and Material Flow Uncertainty

by

Roberto Noriega

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Mining Engineering

Department of Civil and Environmental Engineering
University of Alberta

ABSTRACT

Block caving methods have become desirable underground massive mining techniques as close to the surface, high/grade deposits are exhausted, and current open-pit mines reach their final mining limits. Block caving mining is the only method that can rival the economies and production capacity of open-pit exploitation, offering low operating costs as well as reduced environmental impacts in comparison. Some of the disadvantages, however, are high capital cost requirements, long development times required, and the operational challenge associated with caving mining practices. Block caving mining is based on the undercutting of the rock mass, inducing fragmentation on the overlying mass and extract it as it flows through a developed opening called drawpoints. The flow component poses a major operational challenge as the potential dilution and mixing introduce a large source of uncertainty in the economic forecasts for planning purposes. Moreover, geological uncertainty in relation with the grade and rock type estimation for the generation of numerical deposit models is also an issue.

This research presents a stochastic optimization model incorporates explicitly geological and material flow uncertainty to generate an optimal life of mine schedule for block caving mines at a block model scale. The optimization framework works over two steps: it initially aggregates the individual blocks into production units based on desired drawpoint spacing, representing the draw columns, and mining units based on the minimum draw rate, representing the slices that are commonly used in block caving mine planning. The mining units then become the basic scheduling unit for the stochastic integer programming scheduling model. Uncertainty is characterized by the development of multiple numerical deposit simulations. Geological simulations are developed using geostatistical simulation techniques, with sequential indicator simulation for rock types and sequential Gaussian

simulation for grades. Material flow uncertainty is integrated by the concept of a cone of movement. As each mining unit is extracted, it leaves a void that can be filled by any fraction of the material on its surroundings based on the flow properties of the broken rock mass. A cone, based on potential horizontal displacement and vertical slip angle of the broken rock mass is used to generate grade and tonnage mixing scenarios for each mining unit. The cone is placed at the bottom of each mining unit, and a random sample of the blocks from the deposit model that are contained within it, "filling" the mining unit, is used to update its grade and tonnage. This allows for scenarios where each mining unit material could potentially be part of fractions of adjacent units as well as waste blocks at the orebody accounting for dilution.

The stochastic mathematical model takes as an input the set of geological and material flow simulations to generate a single best schedule that maximizes the expected economic value from the uncertainty sources, while minimizes the deviations incurred in production and average grade targets due to the variability between the potential scenarios. The operational constraints considered in the model include mining capacity targets, average production grade, minimum and maximum heights of draw, minimum and maximum vertical draw rates, undercut development rate, maximum adjacent relative height of draw, mining precedence both horizontal and vertical, and mineral reserves.

The model was tested in a case study, for which a set of 20 geology simulations were obtained. A deterministic, stochastic with only geological uncertainty and stochastic with both geological and material flow uncertainty schedules were generated for comparison and evaluation purposes. Moreover, the deterministic mine sequence was evaluated over the uncertainty scenarios to quantify the impact of the uncertainty in the economics of the project. The models were used at different undercut elevations, to identify the most profitable one. The deterministic case yields the best undercut at 635m while both stochastic cases find it at 605m,

a significant difference. The deterministic schedule, when evaluated over the geological and geological and flow scenarios can lead to an expected NPV 8% to 13% lower than those reported. Although the stochastic schedules generate an expected NPV that is 3% to 11% lower than the reported NPV for the deterministic case, it is a more reliable estimate. Also, larger footprints are obtained through the stochastic schedule which could potentially unlock more value as more information is obtained.

Incorporating geological and material flow uncertainty through the approach presented in this research can aid decision-makers to make more informed and robust decisions to maximize the value of block caving projects at a prefeasibility stage, where little knowledge of the behavior of the rock mass is known.

This Thesis is Proudly Dedicated To:

My parents:

Roberto and Patricia,

and

My little sisters:

Andrea and Valeria

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Yashar Pourrahimian, for providing support, insightful comments and revisions throughout the different phases of the development of this research. I admire his hardworking and open nature, and have enjoyed my time in his research group. I look forward to keep working with him in the future.

I would also like to extend my thanks to the professors I've had the chance to meet and attend to their lectures at the University of Alberta: Dr. Hooman Askari-Nasab, Dr. Wei Victor Liu, Dr. John Doucette and Dr. Clayton Deutsch. I have learned a lot from them as well.

I would like to express my appreciation to my friends at the research group in the mining department at the University of Alberta, as well as my friends in the city of Edmonton. I would have not enjoyed my time here without them.

This research or any other goal I have achieved in my life would not have been possible without the support of my parents and sisters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Parameters

BHOD	Best Height of Draw
CoM	Cone of Movement
EPGAP	Relative MILP Gap Tolerance
EV	Economic Value
GP	Goal Programming
GSLIB	Geostatistical Software Library
HOI	Height of Interaction
IP	Integer Programming
LOM	Life-of-Mine
LP	Linear Programming
MILP	Mixed Integer Linear Programming
NPV	Net Present Value
OK	Ordinary Kriging
QP	Quadratic Programming
SGS	Sequential Gaussian Simulation
SIP	Stochastic Integer Programming
SIS	Sequential Indicator Simulation

LIST OF NOMENCLATURE

Set

O^b	Set containing all PUs that overlap with block b, with number of elements $N\left(O^b\right)$
C^c	Set containing the mining units that are within the production unit c. Each set has a total number of elements of $N\left(C^c\right)$
$V^{u,c}$	Single element set containing the mining unit directly below unit u of the production unit c. Used for vertical precedence constraints
$H^{u,c}$	Set containing the production units that have to be opened before the extraction of the mining unit u of the production unit c, based on the mining direction. Each set has a total number of elements of $N(H^{u,c})$
A^c	Set containing all adjacent production units to the unit c for cave back slope constraint. Each set has a total number of elements of $N(A^c)$

Indices

$c \in \{1,\ldots,C\}$	Index for all possible production units
$b \in \{1,\ldots,B\}$	Index for all possible blocks in a given undercut level that are part of a PU with non-zero metal content
$u \in \{1,\ldots,U\}$	Index for all mining units
$t \in \{1,\ldots,T\}$	Index for all periods
$c \in \{1,\ldots,C\}$	Index for all production units
$s \in \{1,\ldots,S\}$	Index for all geological and material flow realizations
u_c^1	Index for the first or lowest mining unit within the production unit c

Decision variables

$x_c \in \{0,1\}$	Binary variable. Takes the value of 1 if unit c is included in the layout and 0 if not.
$z_u^t \in \{0,1\}$	Binary variable controlling the decision to extract the mining unit u of production unit c in period t
$tonDev_{t,s}^{+} \in [0,\infty]$	Continuous variable representing the ore tonnage "positive deviation" (overproduction) incurred on period t by deposit simulation s
$tonDev_{t,s}^{-} \in [0,\infty]$	Continuous variable representing the ore tonnage "negative deviation" (shortage) incurred on period t by deposit simulation s
$gradeDev_{t,s}^{+} \in [0,\infty]$	Continuous variable representing the grade quality "positive deviation" (over upper bound) incurred on period t by deposit simulation s
$gradeDev_{t,s}^{-} \in [0,\infty]$	Continuous variable representing the grade quality "negative deviation" (under lower bound) incurred on period t by deposit simulation s

Parameters

M_c	Metal content of production unit c
$MetalPrice$	Metal price per ton of metal ($/t)
p^{+}	Cost for positive deviation (overproduction) of ore tonnage ($/t)
p^{-}	Cost for negative deviation (shortage) of ore tonnage ($/t)
q^{+}	Cost for positive deviation (over upper bound) of grade quality range ($/t)
q^{-}	Cost for negative deviation (under lower bound) of grade quality range ($/t)
Rec	Metallurgical recovery of the processing operations (%)
g_u	Grade of the mining unit u of production unit c (g/t)
Ton_u	Tonnage of mining unit u of production unit c (t)

$S_{u_c^1}$	Planar surface area of production unit c (m²), based on the first (lowest) MU of the PU
$Height_u$	Height of mining unit u (m)
MC	Mining cost per ton of ore in period t ($/t)
PC	Processing cost per ton of ore in period t ($/t)
$OpCost_c^t$	Development cost of production unit c, accounted for when extracting the first (lowest) MU, u_c^1, of production unit c ($)
i	Discount rate (%)
$\overline{M}^t$	Ore production target in period t (t)
$\overline{G}^t$	Average grade upper bound on period t
$\underline{G}^t$	Average grade lower bound on period t
$\underline{DR}$, $\overline{DR}$	Minimum and maximum Draw Rate of production units per period (t/period)
$MaxOpRate^t$	Maximum undercutting rate (m²/period)
$MaxAbsDiff_c^{c'}$	Maximum allowable absolute difference in height between production unit c and each member c' of its adjacent set A^c to maintain cave back slope constraints
$EPGAP$	Relative gap tolerance for the branch and bound MILP solution method

CHAPTER 1

INTRODUCTION

This chapter explains the background and structure of the study. The definition of the problem is presented followed by a summary of the relevant literature. The objectives are defined as well as the limitations and scope of the study, and the research methodology is elaborated.

1.1 Background

Underground mining methods are becoming more desirable as innovations in technology enable the exploration of deeper mineral resources, environmental and reclamation requirements play an increasingly major role in the decision-making process of mining projects, and current operating surface mines are reaching their projected final limits. Caving, and particularly block-caving, are massive underground production methods that present an important alternative to access deep and low-grade mineral resources, with the ability to attain high production rates and low operating costs, rivalling the economics of large open-pitopen-pit mines. The basic concept of caving mining is to undercut the rock mass at a certain elevation and breaking the overlying rock, then extracting it below the undercut through a series of openings called drawpoints, from which the flow of the broken ore has been shown to follow a cylindrical column-wise shape (D. Laubscher, 2000).

Life-of-mine (LOM) planning is a key step in the development of a mining project, from the evaluation to the later operative stages. This process is carried in two sequential steps, first determining the economic mining boundaries of the orebody, which define the final mining limits, and then the production scheduling within this envelope. This general workflow is adapted for both surface and underground mines considering the particular characteristics of the method, and different optimization algorithms in commercial software packages have been designed to solve the problem of transforming a mineral resource into a mineable reserve with a feasible production plan (Alford, 1995; Whittle, 1999). In the area of block caving, GEOVIA PCBC™ software package stands as the absolute industry standard (Diering, 2000).

The construction of a quantitative deposit model representing metal grades and other parameters incurs in major estimation uncertainties due to the very low volume of samples obtained from exploration programs in relation to the volume of the estimation domains. Mine plans are built assuming a single estimated model, which generate unreliable plans with a low probability of achieving its expected forecast (Dimitrakopoulos, et al., 2002). Novel algorithms have been developed to tackle this problem, integrating multiple simulated deposit models that capture uncertainty on a strategic mine plan (Dimitrakopoulos, 2011; Dimitrakopoulos & Ramazan, 2008; Koushavand, 2014). Additionally, block caving operations rely on the gravitational flow of the broken rock mass, a process which is very challenging to model accurately. This flow mechanism introduces another major source of uncertainty in the mine planning workflow for block caving (F Khodayari & Pourrahimian, 2019).

This thesis proposes a stochastic programming methodology for the definition of the economic mining limits and production scheduling of block caving mines that captures metal

grade and material flow uncertainty on a block-model scale, through the use of grade and flow simulations. The resulting framework can be used to evaluate the application of caving mining methods on mineral deposits at prefeasibility stages, as well as more detailed studies as more information becomes available, providing a robust mine plan and economic performance estimation.

1.1 Statement of the Problem

Traditional mine plans are built using a single estimated deposit model, in which the mineral deposit is discretized into blocks. Each block has assigned attributes such as grade, rock type, and density. estimated from samples obtained from drillholes. These values are computed usually through geostatistical techniques that account for spatial trends, with ordinary kriging (OK) being the most common method. Based on this quantitative model, an economic value (EV) can be assigned to each block from its metal content and estimated extraction and processing costs, given a market price assumption. The economic block model is then optimized to define the final mining limits and a production schedule with certain objectives such as maximizing the net present value (NPV) while accounting for different constraints representing the mining method and a set of production capacity assumptions (Hustrulid, et al., 2013). The output of this process, in terms of economic and production forecasts, is used for the decision making process during the different stages of the mine.

Uncertainty in the estimation of metal grades and rock type modeling that arises due to the low volume of samples obtained during exploration programs have a high impact in the reliability of the projected cash flows and mining rates, leading to unexpected shortages and poor economic performance of the project. Adopting a risk-oriented approach, in which multiple simulated orebodies are built to capture the whole range of scenarios based on the available data, and building LOM plans that make use of them provide more robust schedules that take advantage and quantify geologic uncertainty (Dimitrakopoulos, 2011; Godoy, 2016). Figure 1-1 shows a summary and comparison of both approaches to mine planning.

Moreover, block caving operations rely on the gravitational flow of broken rock as the ore is extracted through the drawpoints at a specific elevation. This flow mechanism is related to the geotechnical and geological properties of the rock mass as well as the primary fragmentation or breakage method, leading to the mixing of the original estimated block model grades, which also affects the reliability of the mine plan (Figure 1-2).

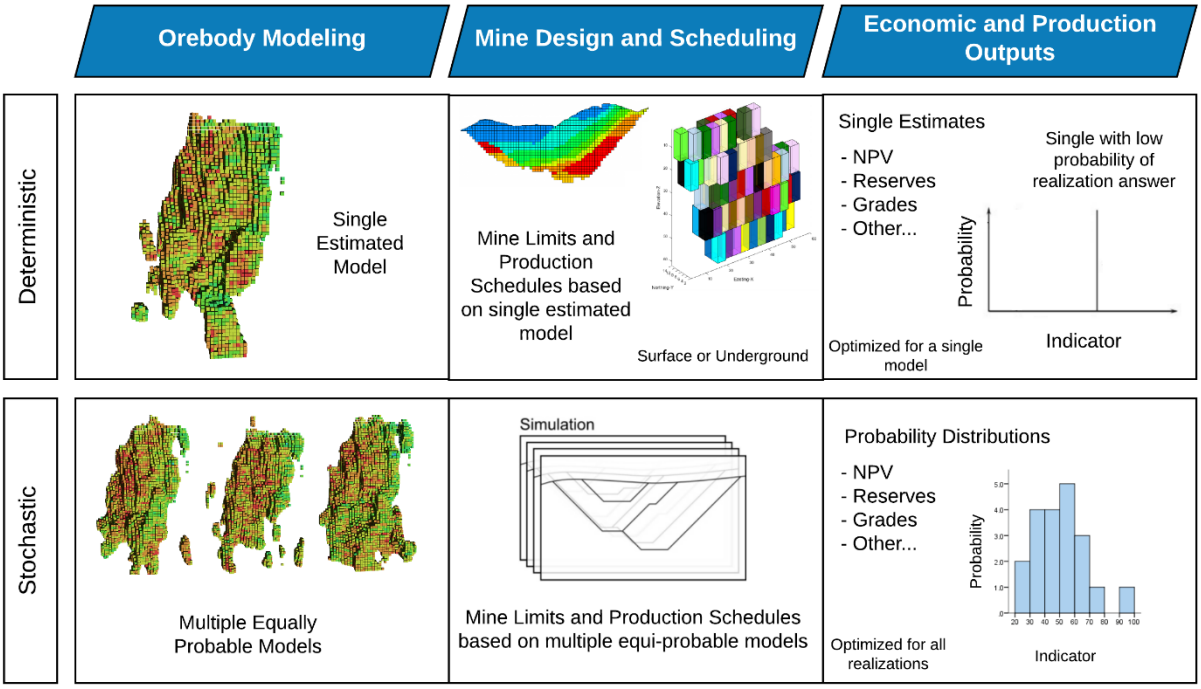


Figure 1-1. General traditional (deterministic) and risk-oriented (stochastic) approach to mine planning.

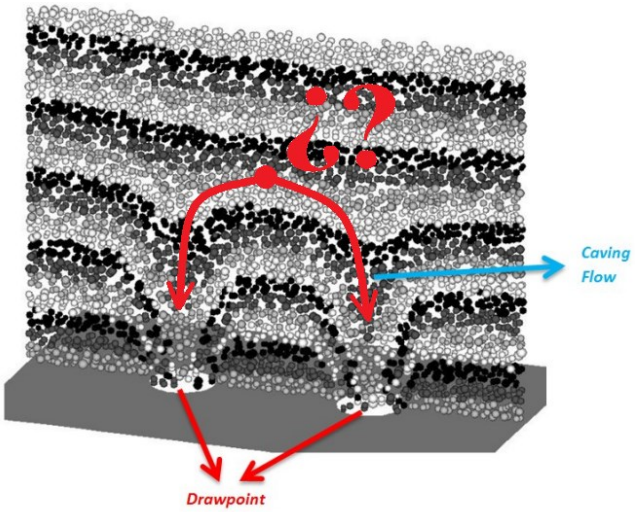


Figure 1-2. Schematic representation of material flow uncertainty in block caving (after Khodayari, 2018).

Literature shows a lot of emphases, and good results, on the development of algorithms and methods to model flow and predict diluted grades based on extraction parameters, however, little research has been carried out in the coupling of material flow with mine planning algorithms to produce a robust mine plan. Furthermore, these algorithms are built for operating mines requiring data and knowledge of the behaviour of the orebody under caving.

LOM plans developed at a prefeasibility or feasibility level currently include more basic flow algorithms, which fail to capture the major uncertainty inherent to this behaviour when limited knowledge of the deposit is available.

For the purpose of this research, geological uncertainty refers to the uncertainty of the grade and rock type estimates of the resource block model due to the scarce number of samples and the natural variability of these variables throughout it. To characterize the geological uncertainty in the orebody model a set of stochastic conditional simulations are built that are as an ensemble represent the variability of the estimation parameters.

Material flow uncertainty in this research refers to the impact on the grade of the production units defined (analogous to slices) due to the mixing of the broken rock as it flows throughout the draw column. When an unit is drawn, the estimated grade does not reflect the potential mixing of rocks, which is a challenging problem to model in detail at a prefeasibility stage. To deal with the mixing uncertainty, a set of scenarios are built where the grade and tonnage of each production unit is updated or recalculated based on the sampling of a set of blocks that fall within a cone of movement, that represents the potential horizontal and vertical movement of the broken rock as it is drawn from the column.

The standard workflow for block caving LOM planning starts with the selection of an undercut elevation, from which the mineable reserves lie in the overlying caved rock mass. The extension of the mining footprint is defined over this level, and the drawpoint excavation layout is designed, estimating the best height of draw (BHOD) which serves as the vertical extraction limit from each drawpoint. Based on the extraction starting point and mining advancement direction, a production plan is constructed for the depletion of each drawpoint up to its BHOD and the economic and other performance indicators of the project can be estimated (Rubio, 2002). This production schedule is built on a column and slice model which aggregates the individual blocks from the deposit resource model into vertical slices within cylindrical shapes from each drawpoint to represent the block caving extraction scheme (Diering, et al., 2010). This assumes knowledge of the mining footprint extensions and drawpoint location and layout, however some models can also work directly on a block model scale to provide estimates for evaluation at earlier stages of the project (Rodriguez, 2018; Villa, 2014). A summary of the block caving LOM planning workflow is presented in Figure 1-3.

This thesis focuses on step 2 of the general workflow for the LOM planning of block caving mines, in which it is of interest to build an initial schedule to define the optimum undercut level and caving envelope along with economic performance measures of the project,

developed on a block scale rather than column and slice models. However, these blocks are aggregated into units which are representative of the caving extraction method to produce more accurate results and improve computing times. The selection of the undercut level plays a significant role on the mineable reserves as the underlying rock mass is loss, and is not a flexible decision. This along with the defined caving envelope, based on the footprint extension and the BHOD, can have an enormous impact in the later stages of the planning workflow, as sections of the deposit are discarded, which could be taken advantage of to improve the decision making process. In this research, a mathematical programming model is developed to optimize this stage incorporating geologic and material flow uncertainty in order to provide a robust and near-optimal solution. Figure 1-4 presents a schematic of the problem statement for this research thesis.

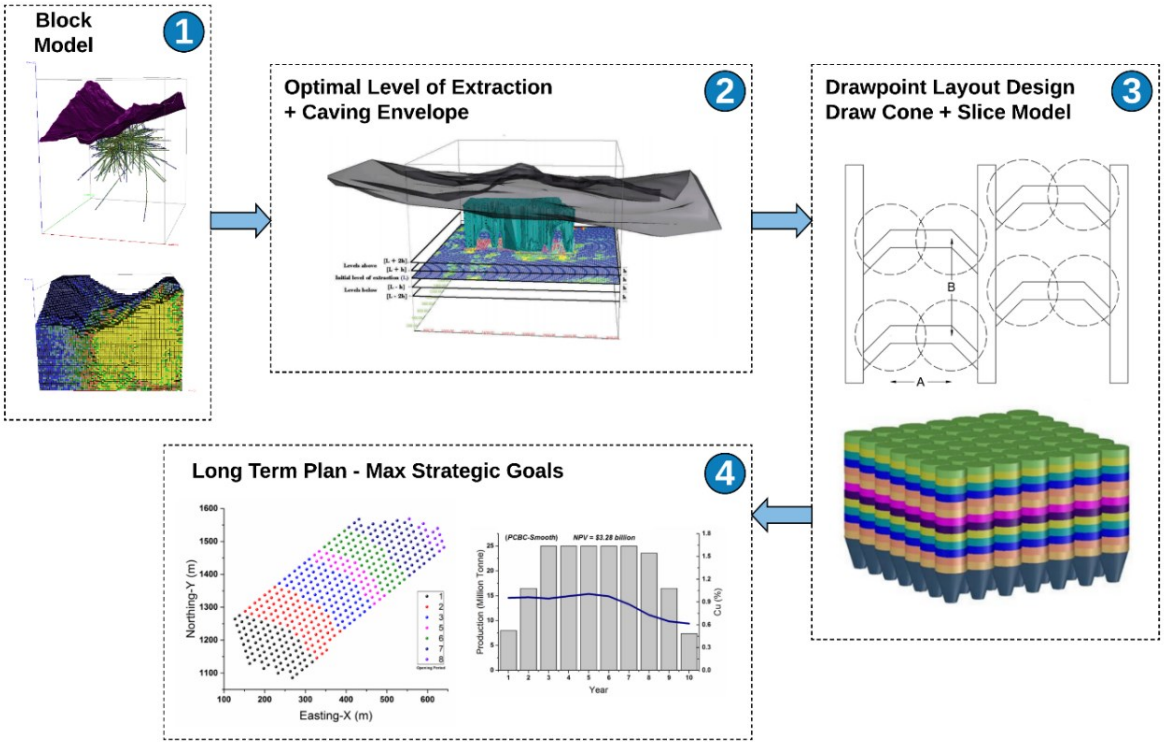


Figure 1-3. General block caving LOM planning workflow.

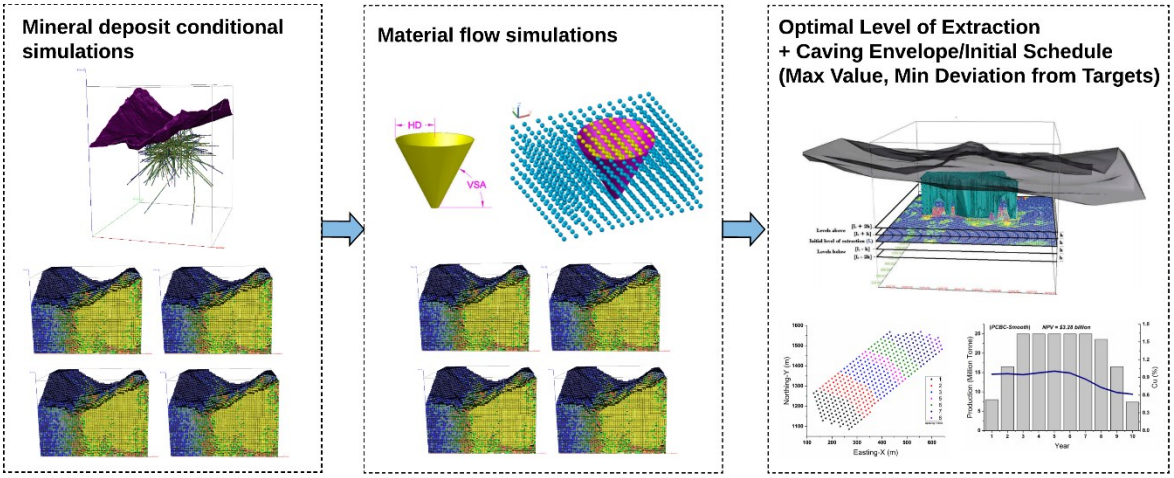


Figure 1-4. Schematic representation of the problem definition.

A mathematical programming framework is developed in this research that maximizes the economic value of the project while minimizes the deviations caused in ore production tonnage and grade quality due to the potential geological and material flow scenarios. The framework incorporates operative constraints including production and grade quality targets, undercut development rates, vertical mining rates (draw rate per draw column), mining precedence based on the advancement direction and adjacent differential height of draw to control the cave profile.

The following research question summarizes the problem and purpose of this dissertation.

> *Is it possible to incorporate geologic and material flow uncertainty into long-term mine planning for block caving mines considering adequate operational constraints, which improve on current methodologies while providing better control over potential production and quality target deviations and more robust economic forecasts?*

1.2 Summary of Literature Review

Current industry practice regarding the long-term planning of block caving mines is based on the methodology described by Diering (2000, 2010) in GEOVIA PCBC™ software, which is used by virtually almost every operating block caving mine or potential project. PCBC includes the Footprint Finder (FF) module to determine the optimal level of extraction as the first step in the designing and planning of a block caving mine. FF takes as an input the deposit block model including economic valuation of each block, commonly determined based on the estimated mineral grades, mining costs and revenue factors, accounting for dilution with a vertical mixing model integrated based on Laubscher model (1994). Each elevation is evaluated

independently generating vertical columns based on the block locations. The BHOD for each column is calculated as the height that yields the maximum economic benefit, limited by a defined maximum height of draw. A discounted economic value is obtained for each column independently by adding the economic values of each block starting from the selected undercut, discounted by a vertical mining rate usually expressed as meters per year, up to the determined BHOD.

Additionally, development costs can also be included to account for the opening of each column. The columns with a positive economic value are considered to be included in the mining footprint at the current elevation. The value and ore tonnage of each potential undercuts are calculated as the summation of the economic values and ore tonnages of all the columns included within the footprint respectively and are used as the main criteria to select the optimal level to start the extraction. The practical workflow makes it easy to examine multiple levels quickly and choose the best undercut elevation and an initial footprint; however, it does not consider the interaction between each column, as a certain cave back slope and relative extraction rates are required to maintain a desirable cave shape. Moreover, it does not explicitly account for the horizontal mining advancement direction within the undercut and the potential effect on the economic value from the discounting associated with the opening of columns at different periods. Recent versions have been developed to include a scheduling option within the FF module; however, the potential workflow becomes time-consuming as it would be required to iteratively formulate multiple scenarios for each undercut, while still not guaranteeing an exact optimal solution (Villa, 2014).

A methodology proposed by Elkington, et al. (2012) uses an integer programming (IP) formulation to determine a 3D cave outline over multiple cut-off grades for a deposit block model. The objective was built to maximize the metal content above a certain cut-off constrained to a minimum mining footprint area, a minimum and maximum column height, a maximum adjacent height of draw and a minimum horizontal pillar distance between caves, in order to represent the geometrical characteristics of a caving operation. The method allows defining alternative caving outlines or grade shells at different cut-offs to aid the planner in identifying high-grade areas, guiding the cave development and selecting the best extraction level placement. Some of the disadvantages of this method are that it does not maximize economic value explicitly, as it maximizes metal content within the different caving outlines. Furthermore, no time factor is considered in the model so the economic discounting effect due to the extraction sequences of a block caving operation logic is not accounted for.

An approach presented by Vargas, et al. (2014) introduced geological uncertainty into the process of selecting the elevation and defining the caving outline by building multiple realizations of the block model. These conditional simulations are evaluated in GEOVIA PCBC™ FF module to determine the best undercut elevation and afterwards a version of the final pit limit procedure is adapted to mimic the geometry of the caving outline and define an economic envelope. The blocks below the selected elevation are eliminated and the deposit block model is turned upside down, where the final pit limit algorithm to calculate a mining envelope is used with the added restriction of allowing for vertical walls up to a certain percentage of the height of column, then controlling the cave back slopes through the precedence constraints for the blocks on top of said percentage. A final step filters the result to take out individual columns with no neighbors included in the footprint and smooths the outline. The algorithm is repeated for the multiple simulations obtaining the undiscounted economic value and ore tonnage of the envelope. A value at risk evaluation is performed to quantify and summarize the variability of the economic values and tonnages associated with different risk levels to aid the decision maker. The methodology does not account for the discounting effect of the vertical, within columns, and horizontal, within the undercut, since the method used to calculate the envelope is a variation of the final pit limit optimization algorithm which maximizes undiscounted profit. This could lead to a significant difference between the envelope value and the actual expected NPV of the project, as well as the shape of the estimated caving outline. Furthermore, the methodology allows to quantify the impact of uncertainty but does not include it explicitly in the optimization procedure.

There is a clear gap in the literature regarding the incorporation of geologic uncertainty in the planning of block caving mines. The advantages of using stochastic orebody simulations and stochastic mine planning is well documented in open-pit mining (Dimitrakopoulos, 2011; Dimitrakopoulos & Ramazan, 2008; Godoy, 2016; Leite & Dimitrakopoulos, 2007), providing more robust plans in term of control of potential deviations, as well as adding value to the total NPV of the project and reserves. However, in block caving to date, the only documented stochastic planning method is the one presented by Dirkx, et al. (2018). In their methodology, a stochastic programming model is developed that accounts for multiple geologic simulations to minimize the deviations from production targets and incorporate hang-up uncertainties that affect the projected overtime during the mine life.

The flow mechanism of broken rock through drawpoints has been a major area of research and development in block caving, due to its large impact on a caving project operation

(Kvapil, 1965; Marano, 1980). The investigations in this area are usually done either on full-scale experiments, pilot tests or numerical models.

GEOVIA PCBC™ contains a numerical empirical code for the prediction of flow in caving based on the analysis of in situ tests in operating mines. The procedure defines draw cones over which empirical mixing rules are applied to account for the flow mechanism observed in practice, without modeling detail evolution of the draw profile. For better results, it is suggested that the inputs required to apply the flow mixing area calibrated to match the behavior of the rock mass (Diering, 2000).

One of the best known numerical models for flow prediction in caving is REBOP (Rapid Emulator Based on PFC). REBOP was initially developed by Cundall, et al. (2000) based on observations and tests of draw simulations in PFC3D, and studying in detail the mechanism of isolated draw offering a powerful tool for flow prediction. Furthermore, Pierce (2010) improved it by a detailed investigation of the material properties that govern the movements zones evolution, and embedded a procedure to predict stresses under draw within the cave. The calibration and validation process for the different material properties and other inputs is key in order to guarantee robust results.

Different stochastic approaches to material flow have been proposed (Alfaro & Saavedra, 2004; Deserable, 2006). As a general overview, the models discretize the material and estimate the probability of movement of each unit based on different parameters, from empirical coefficients to material properties. The different simulations can be generated based on these probabilities. Castro, et al. (2009) propose a flow simulator based on cellular automata to estimate dilution entry, mixing and ore recovery. The model estimates the probability of movements for the broken particles based on empirical coefficients and material properties, and show good results when implemented in real operating mines. The framework is used for flow prediction and draw control in operating caving mines.

The approaches discussed above provide great tools during the operation of block caving mines. However it is of interest to couple flow material mechanisms during the LOM planning stage, where knowledge generated from other operating mines can be used to generate a representative set of scenarios. Khodayari & Pourrahimian (2019) introduce one of the first approaches to incorporate material mixing explicitly within the mine planning workflow for block caving. Rather than modeling a detail flow mechanism, it is treated as a source of uncertainty reasoning that at this stage little is known about the response of the actual response of the rock mass, but multiple representative scenarios can be generated, and an optimization

approach over the expected value of this set of potential scenarios can increase the value of the project as well as providing a more reliable mine plan. The concept of a cone of movement is developed (CoM), based on Alvial (1992) observation on full-scale tests at El Teniente block caving mine. Also mentioned in (D. Laubscher, 2000), Alvial marker experiments in El Teniente showed that horizontal displacement of particles varies between 2m and 42m, averaging 14.5m following a vertical slip angle between 60° and 88°, averaging 80°. They then model the potential movement of particles based on these observations as a cone of movement defined by a horizontal displacement (HD) and vertical slip angle (VSA). When material is drawn from a drawpoint, the void generated can be filled by any of material contained within this CoM pertaining the particular drawpoint, and therefore grades and tonnages are updated following a random sample from this space. These serve as scenarios to optimize the expected value of the project rather than a single estimated model with a premixing algorithm, such as (D. Laubscher, 1994). A case study was used where the introduction of flow uncertainty generated mine plans with larger value than those generated with a single estimated model with premixing algorithms on PCBC.

This thesis adapts the concept presented in Khodayari & Pourrahimian (2019), coupling it with geologic orebody simulations to generate a mine planning workflow that can be used at prefeasibility stages on a block model scale, providing robust plans and more reliable production and economic forecasts.

1.3 Objectives of the Study

The objective of this research is to develop, implement, and verify a theoretical optimization framework for the definition of the economic mining limits and LOM production schedule for block caving mines, incorporating metal grade and material flow uncertainty. The goal of the optimization model is to obtain an optimal economic envelope and schedule that maximizes the NPV of the project, under a set of technical and economic parameters, and minimizes the ore production and grade quality deviations from defined targets, caused by the variability due to the uncertainty in grade estimation and material flow.

The methodology presented takes multiple conditionally simulated mineral deposit grade models based on geostatistical techniques as input, and developes flow scenarios using the concept of a cone of movement to generate a near-optimal production schedule.

As an initial step, the block model is aggregated into production units (PU) based on the extraction scheme of block caving mining, to reduce the number of variables needed to obtain the LOM plan while still being representative of the method. These units are built using a Binary

Integer Programming (BIP) method to optimize the metal content contained within the potential mining reserves, and serve as the scheduling basis for the planning algorithm. A Stochastic Integer Programming (SIP) model is then formulated to obtain the production schedule that maximizes the NPV under the following constraints: mining production capacity target, mining advancement direction and extraction precedence, suitable vertical mining rates (draw rates), undercut development capacity, cave draw slope control, and blending and grade quality target.

To achieve these goals, this research focuses on:

- Develop a methodology to generate strategic schedules for block caving projects that maximize its economic value while minimizing the deviations caused by the uncertainties in metal grade estimation and material flow.
- Develop techniques that allow the practical implementation of the proposed methodology in a reasonable CPU time, while still generating near-optimal solutions.
- Integrate the optimization framework into the standard workflow for block caving LOM planning at early stages, by evaluating its use in the selection of the best level of extraction and obtaining relevant KPI's pertaining a mining production schedule.
- Evaluate the solutions obtained from the optimization framework using a case study to highlight the effects of incorporating grade and flow uncertainty into the LOM planning workflow for block caving.

1.4 Scope and Limitations of the Study

The following assumptions are made in the development of the methodology:

- The modelled orebody is considered to be a stationary domain to generate multiple grade and rock type scenarios by the use of Sequential Gaussian Simulation (SGS).
- Material flow is incorporated in the model as multiple scenarios, not dynamically within the planning optimization algorithm. Material flow is treated as a mechanism that generates different grades scenarios based on a resource model and the common flow pattern observed in block caving.
- The aggregation of the individual blocks into production (PU) and mining units (MU) is based on the potential drawpoint spacing at the extraction level, and the minimum vertical extraction rate (draw rate) required. The block size of the input deposit model will limit the resolution of the aggregation algorithm and selection of its parameters.

- The scheduling SIP model defines binary decision variables to extract a mining unit on a given period or not, rather than a continuous one. The size and tonnage of the unit, based on the aggregation parameters, limits the resolution of the production schedule solution and selection of its parameters.
- To solve the SIP model it is converted into its equivalent deterministic MILP model, by the introduction of recourse variables and deviation penalty costs. The objective function is then reformulated to maximize the expected value of the profit contributions from the extraction of each mining unit amongst all the simulated models, which are also incorporated in the constraints to minimize deviations from required production targets.

The scheduling is carried over MUs, which are built as an aggregation of individual blocks from the deposit model with plan dimensions based on the potential drawpoint spacing and height based on the minimum draw rate required for caving the rock mass. This concept is aligned with the different aggregation methodologies that have been presented and validated for the scheduling of open-pit mines, but representing the extraction scheme in block caving mines. Binary decision variables are used since the mathematical modeling of the block caving operating constraints using continuous variables require the introduction of a large number of extra binary variables, which render the problem unpractical requiring excessive computing times. This requires the definition of parameters such as maximum draw rates, maximum heights of draw and adjacent relative draw rates to be in line with the block size and mining unit size. No detailed scheduling such as drilling and blasting or ventilation are considered, as the model addresses a LOM plan.

The solution approach to the SIP model is to express it in its equivalent deterministic MILP through recourse variables and penalty costs on deviation targets. These penalty costs allow the user to define a risk profile on the production schedule. The equivalent MILP is solved using IBM CPLEX, which uses a branch-and-cut solution strategy. A sliding time window heuristic is implemented to improve the computing time required while still providing near-optimal solutions.

1.5 Research Methodology

The main motivation for the development of this research is to improve block caving production scheduling methods by the incorporation of geological and material flow uncertainty into an optimization algorithm to maximize the project value, while minimizing deviations from production and material quality targets. The scheduling optimization

framework works on a block level scale and also makes the decision on the extension of the footprint at the undercut level and the BHOD and caving envelope. To reduce the number of variables, an initial step aggregates the individual blocks into MUs, representing the column-wise extraction scheme of block caving, based on a desired drawpoint spacing and planned dilution. The aggregation maximizes the metal content of the units, calculated in reference to an undercut elevation, in order to cover the deposit while not imposing a boundary within the orebody and leaving this decision to the later production scheduling step.

Geological and material flow uncertainty are characterized by generating multiple scenarios. The scheduling algorithm becomes a stochastic optimization model which is transformed into its equivalent MILP formulation by the introduction of deviations, or recourse, variables. These deviation variables are assigned a cost that gives the user control over the risk of over- or under-production from defined targets in both ore tonnage and grade quality over the LOM plan. The objective function is reformulated to maximize the expected value of the economic profit from the extraction of a unit while minimizing the total cost incurred due to deviations from targets over the multiple geological and flow scenarios. The following tasks are completed to build the optimization framework:

- Generate a set of block model realizations of the grade and rock type of a mineral deposit using SGS techniques.
- Develop a BIP model to aggregate the individual blocks into production and mining units to reduce the number of variables to schedule while being representative of the block caving extraction scheme. The aggregation maximizes the expected metal content of the layout considering the multiple geological realizations in order to give the boundary and sequencing decisions to the scheduling algorithm.
- From the set of geological block model realizations, generate material flow simulations using the CoM concept based on HD and VSA parameters to represent the geomechanical characteristics of the broken rock mass. This mechanism to simulate material flow and its incorporation in block caving planning is adapted from Khodayari (2018).
- Develop a SIP model to generate a production schedule and mining envelope from the previously generated mining units considering all the potential geological and flow scenarios. The SIP is solved by converting it into its MILP equivalent formulation defining deviation variables that the user can control to define a risk profile.

- Implement the formulation in a MATLAB environment using IBM CPLEX as the optimization engine.
- Test the formulation on a case study to assess the results in terms of the practical feasibility and validity of the mining plan.
- Compare a deterministic version of the model that uses a single estimated deposit model with the stochastic version. Quantify and compare the impact of incorporating grade and flow uncertainty with respect to the project NPV, production output, grade quality output, economic envelope and best elevation for the undercut level.

The research methodology is summarized in Figure 1-5. The initial step for the building of geologic simulations follows the standard workflow for geostatistical simulations widely described in the literature (Godoy, 2016; Goovaerts, 1997; Rossi & Deutsch, 2014), and developed using the geostatistical software library GSLIB (C. Deutsch & Journel, 1997).

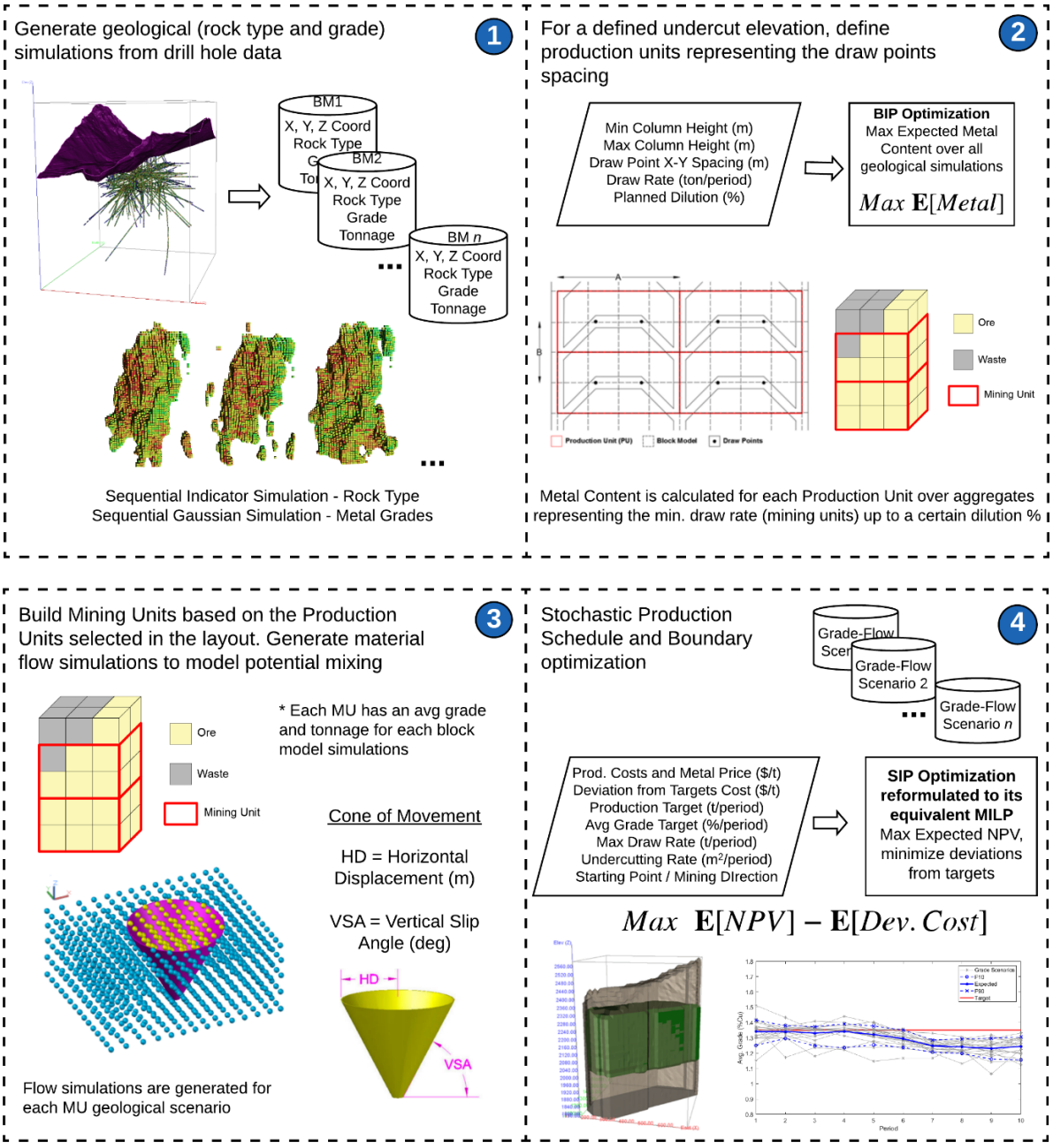


Figure 1-5. Summary of the research methodology for this research.

The general steps towards geologic simulation comprise:

1. Declustering of the drillhole data to obtain the representative distributions of both rock type and metal grade.
2. Descriptive and multivariate statistical analysis to explore data and relations between different variables, define stationary domains.
3. Determine principle direction of continuity for variogram modeling.

4. Normal score transformation of the data.
5. Variogram modeling.
6. Definition of simulation parameters including the number of grid nodes in each direction and distance between them.
7. Generate *n* realization using sequential indicator simulation (SIS) for rock type, and for each rock type simulation generate a metal grade realization using SGS.

Once the *n* geologic block model realizations have been generated, the next step is to aggregate the individual blocks into suitable MUs that represent the block caving extraction scheme and reduce the number of variables for the scheduling decisions.

The first decision is the selection of an undercut elevation, which is evaluated based on the total profit obtained by the extraction of the individual blocks at a particular elevation discounted based on a vertical mining rate (draw rate), going through a set of candidate elevations to select the one that yields the highest economic value. This thesis proposes the use of the presented optimization framework to provide a more reliable estimate of the profit for each undercut elevation, by evaluating a set of candidate undercuts with the incorporation of the discussed sources of uncertainty and more detailed operating parameters.

For a candidate undercut the individual blocks are aggregated into mining units based on a desired drawpoint spacing between drifts (A) and across the minor pillar (B), representing the column-wise area of extraction in block caving, with the vertical dimension set based on the minimum draw rate (vertical mining rate) required to sustain caving of the rock mass. Figure 1-6 shows a schematic of the aggregate units.

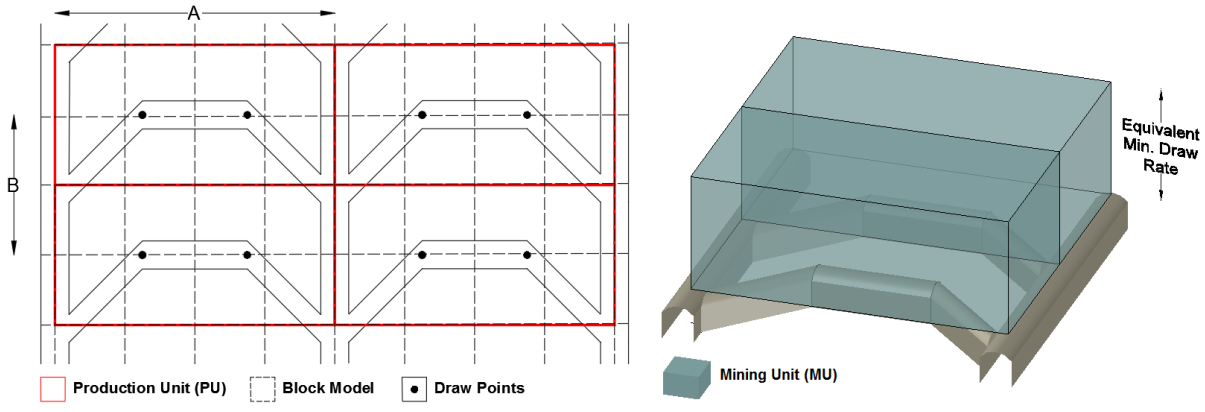


Figure 1-6. Aggregation of blocks into mining units for scheduling.

The aggregation first determines the best arrangement on the plan section of the undercut, representing the draw zones or production units, which maximize the expected metal content contained in the layout.

The expected metal content of each production unit is calculated as the summation of the average metal contained on the blocks amongst all the realizations, along the vertical column spanning from each particular production unit up to a certain dilution percentage. The calculation is performed on steps over vertical aggregates based on the draw rate, which later become mining units for scheduling purposes, and the dilution is calculated as presented in equation (1.1). The blocks within the mining unit with zero average metal content across all realizations are considered as waste, and once a unit dilution surpasses an established limit the column is terminated, with the calculation going up to a predefined maximum column height. This goes along the practice of allowing certain dilution when defining the column draw height limits (D. Laubscher, 2000).

$$\frac{Waste\ Ton}{Waste\ Ton + Ore\ Ton} \times 100\% \tag{1.1}$$

Metal content is selected rather than economic value at this stage. Due to the vertical extraction system in block caving, the economic value of each draw column is calculated up to its BHOD, which is the final height at which the column is closed. Traditionally, the selection of this height is done before any scheduling by evaluating each column individually and selecting the height at which the maximum cumulative economic value is reached (Rubio, 2002). However, since there is no consideration of the impact of the BHOD over the adjacent columns and due to draw control there could be value lost. The extraction of additional material from a particular column above its BHOD could allow the draw of more valuable ore at larger heights on adjacent columns. This framework leaves the BHOD selection for the cave boundaries as an output of the scheduling step, where it decides where to stop and close each column to yield the maximum NPV over the life of the project considering the whole resource. The boundaries of the cave are output of the production schedule.

The MUs have a tonnage and average grade for each block model realization, representing the geologic scenarios. For each scenario, a material flow simulation is generated. This is carried out by adapting the concept of a cone of movement as presented by (F. Khodayari, 2018; F Khodayari & Pourrahimian, 2019). When a portion of the broken ore is extracted through a drawpoint, an open space is generated along the draw column that can be filled by any mass within a certain neighborhood. The modeling of this neighborhood and calculation of

probabilities associated with the movement of material within it is a very challenging problem that requires a great amount of knowledge of the behavior of the rock mass. Full scale tests show that the movement can be thought of as a cone characterized by the horizontal displacement (HD) and the vertical slip angle (VSA) of the broken ore movement within it (D. Laubscher, 2000). Figure 1-7 presents an example of the concept. In this research, the concept is applied over mining units, as these are the smallest scheduling units.

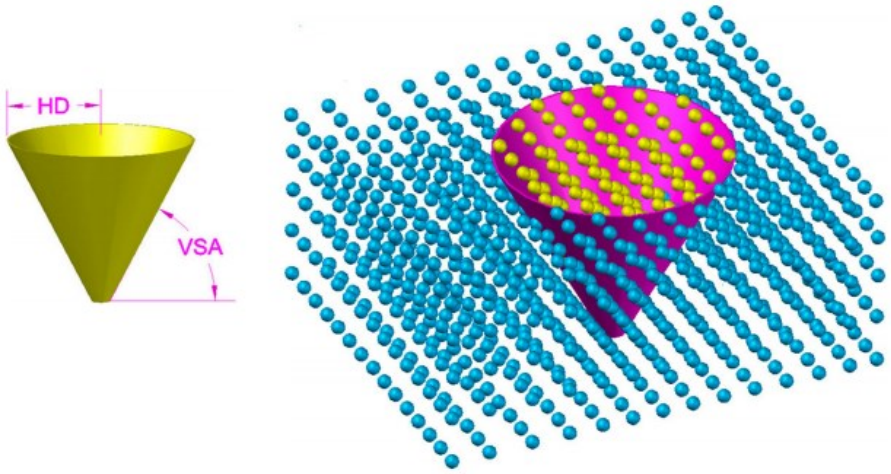


Figure 1-7. Cone of Movement (CoM) concept adapted for the material flow simulation as presented in Khodayari (2018).

When a mining unit is extracted, the void in the bottom of the drawpoint can be filled by any of the units within the neighborhood defined by the CoM. Random scenarios are generated to represent the uncertainty in this movement pattern, where each realization takes a random unit from the set that falls within the CoM to update the grade and tonnage model. A realization is performed for each geologic block model scenario in order to provide coupled geologic and flow uncertainty simulations into the scheduling step. The parameters defining the CoM (HD and VSA), should be established to be representative of the geomechanical and flow behavior of the broken ore.

With the flow simulations, now each mining unit has an updated coupled geologic and flow average grade and tonnage representing the different potential scenarios. A stochastic programming model is developed that takes all the geologic and flow simulations as input to provide a robust mine plan that maximizes the expected NPV of the project.

The objective function of the optimization model uses binary decision variables to select on which period each mining unit is extracted in order to maximize the total expected NPV. To reach this objective, the expected NPV from the extraction of each mining unit, u, at each

period, t, $E\{NPV_u^t\}$ is calculated considering all the generated geologic and flow scenarios. Additionally, the objective function minimizes the expected cost of deviations from ore tonnage and grade quality production targets. These deviations are caused due to the variability of the orebody amongst the different geologic scenarios, which traditional tools based on a single estimated model cannot control.

The following constraints are considered to model the block caving operating extraction scheme:

1. Ore tonnage production targets per period. This constraint is defined as a tonnage per period production target on the overall mining system. By defining it on a period basis, ramp-up and ramp-down periods can be estimated and incorporated.
2. Grade quality target per period. This constraint forces the model to follow as closely as possible an average grade target over the whole mining system on a period basis. This implicitly gives the scheduling algorithm the option to explore blending.
3. Mining precedence. This constraint ensures the sequence is feasible and coherent. It controls the vertical direction to limit the extraction of a particular unit only once the unit below it has been extracted, and also the opening of columns based on the mining advancement direction.
4. Draw rates and continuous extraction. These constraints ensure that the material removed from each production unit does not exceed a required maximum draw rate, which is usually defined based on operational and geotechnical parameters. The continuous extraction constraint guarantees that once a column is opened, at least one unit (representing the minimum draw rate) is extracted each period otherwise it is closed.
5. Undercut development rate. This constraint controls the area in the footprint that can be developed and therefore the number of new production units that can be opened on each period.
6. Adjacent relative height of draw. A maximum adjacent relative height of draw between columns and its adjacent ones. This constraint allows for a more even draw and smoother cave development along the LOM, as well as defining a maximum cave slope at the end of the project.

1.6 Scientific Contributions and Industrial Significance of the Research

The contribution of this research lies in the incorporation of coupled grade and material flow uncertainty in the LOM planning and scheduling for block caving mines, which can be used at early stages for project evaluation and decision making providing more robust and reliable mine plans. This expands current capabilities of software and mine planning algorithms for block caving. Proto-type software can be developed based on the algorithm presented in this research to transfer the generated knowledge to the industry. Chapter 1 of this thesis serves as an introduction to the background and structure of the study. The definition of the problem is presented followed by a summary of the relevant literature. The objectives are defined as well as the limitations and scope of the study and the research methodology is elaborated.

Chapter 2 contains the review of the current state of the literature pertaining block caving LOM production scheduling. The general workflow for block caving mine planning is presented in detail with current software alternatives and their limitations. Standard techniques to model geological uncertainty and material flow are discussed, as well as current methodologies to incorporate them into the planning process. A review on mathematical programming methods and their application in block caving and mine planning, in general, are presented. This chapter highlights some of the limitations of current methods and the rationale for this research.

Chapter 3 contains the theoretical framework for the proposed optimization framework. The workflow for the geostatistical simulation of the deposit metal grades is presented, as well as the details for the flow model used to develop the different simulated models to capture the uncertainty in the planning process. The optimization framework is detailed starting with the aggregation of the individual blocks from the deposit model into production and mining units maximizing the metal content within them. The BIP implementation is presented, detailing the building of the potential production units and the calculation of the coefficients of the objective function (metal content) subject to the desired operational constraints. The SIP formulation is presented, commenting also on the deterministic version. The calculation of objective function and constraints coefficients area detailed, as well as the matrix structure for computational implementation.

Chapter 4 discusses the application of the model to a copper deposit. Multiple geostatistical simulations of the deposit are generated, which are coupled with material flow simulations based on a set of assumed geotechnical parameters to test. Production schedules are built for the stochastic and deterministic (traditional workflow) versions of the model, and compared to highlight the performance of the incorporation of the sources of uncertainty.

Finally, summary, contribution of the research and suggestions for future work are discussed in chapter 5.

CHAPTER 2

LITERATURE REVIEW

This chapter presents an overview of the operating principles of block caving mines, the theory of gravity flow and its influence on different design parameters such as drawpoint spacing. A general review of previously developed mathematical programming models to solve the production scheduling problem in block caving mines is also provided. Finally, current methods that integrate different sources of uncertainty into the block caving mine planning process are discussed.

2.1 Block Caving

2.1.1 General Overview

Block caving is part of a larger set of underground mining methods in which the orebody caves naturally after the excavation of an undercut, with the caved rock mass extracted through a series of openings called drawpoints. The principle of undercutting the rock mass lies in that by the excavation of a horizontal layer at a certain elevation (undercut), the overlying rock mass loses it support and continuously caves by gravity as material is drawn and void space is created in the cave back (Brannon, et al. 2011; Laubscher, 2011).

The governing factors on whether a rock mass is suitable for caving methods are caveability and fragmentation (Rubio, et al., 2004). Caveability is a term coined to describe the minimum undercutting area needed to be opened to induce and sustain caving, and directly affects the feasibility of a caving project. A specialized rock mass rating system, MRMR, was developed for the purpose of assessing the caveability of a rock deposit (Jakubec & Laubscher, 2000; Laubscher, 1990). This system adapts the well-known RMR (Bieniawski, 1976), to introduce measures for weathering, orientation of the cave front, induced stress and blasting. Moreover, empirical charts have been developed that correlate the MRMR and the size of the undercut required to cave the rock mass, expressed as the hydraulic radius (area/perimeter)(Laubscher, 1994).

Fragmentation plays a key role insuccessful caving operation. It affects draw control, productivity and overall costs. There are two main types of fragmentation: primary and secondary. Primary fragmentation refers to the size distribution of the blocks that separate from the cave back as the undercut is developed, and it relies on the natural fragmentation and stress state of the rock mass. Annavarapu (2019) provided a comprehensive review on this topic as well as a practical method to estimate primary fragmentation for a block cave mine. Secondary fragmentation refers to the breakage of the blocks as they move down through the column to the drawpoints, with its size distribution directly impacting the productivity of each drawpoint. Liu (2016) provided a review on methods to characterize the secondary fragmentation observed at drawpoints in a block cave mine.

Block caving is executed usually in two different setups. An approach includes the division of the deposit into multiple large production blocks, or a single panel advancing forward through the mineral deposit (Fuentes, S. & Villegas, 2014).

The layout of block caving mines consists of multiple horizontal levels which serve different purposes. The undercut level, from which the broken rock flows through openings called draw

bells into the extraction level, where it is drawn and transported into a haulage level to transport it to the surface (Brannon et al., 2011). Additional levels such as drainage, crushing/reduction and other services are present depending on the conditions of the mine.

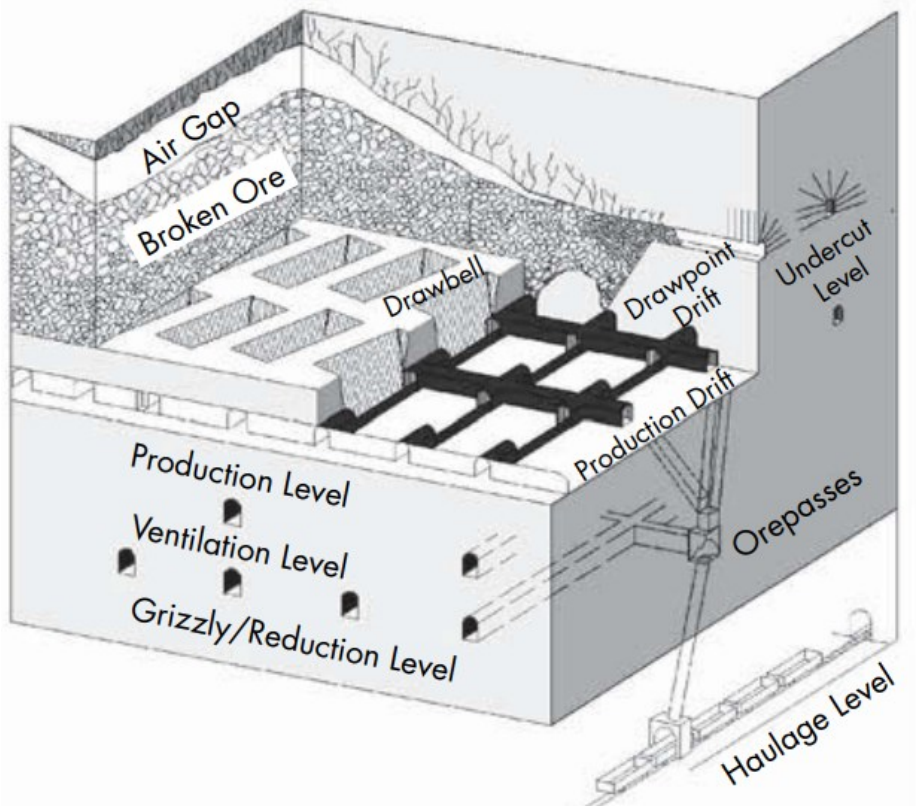


Figure 2-1. General overview of the layout in a block caving mine (Brannon et al., 2011)

The design and initiation of the undercut level is closely related to the design of the extraction level below, and the chosen method of undercutting. In a post-undercutting strategy, the underlying extraction level and drawbells are completed ahead of the blasting of the undercut level. The main advantage lies in the fact that the broken material is readily available for extraction as the extraction level infrastructure is already there. However, the extraction level is subjected to increased abutment stress with substantial damage to the major and minor apex reducing the effective life of the drawpoints and drifts (Brannon et al., 2011).

A pre-undercutting method consists of mining the undercut level ahead of the development of the extraction level. The extraction level is then excavated in a de-stressed environment, with reduced support required. However, since drawpoints are the only point at which stresses can be managed in a caving operation, it offers little flexibility to relieve potential stress situations that can be build up in complex geology environments (Laubscher, et al. 2017).

Finally, the advance undercut method is the most popular method as it reaches a compromise to solve the issues associated with both pre- and post-undercutting strategies. In an advance undercut system, the extraction level infrastructure (drawpoints and drawbells) is done only after the undercut has passed over by a distance no less than the inter-level spacing. Usually, the length between the undercut advance and the extraction level development is kept at around the length of one drawbell. The extraction level is then excavated in a de-stressed environment while offering flexibility to resolve potential geomechanical issues and quicker times for commencing production. However, this method requires high up-front capital expenditure and resources, and also a high level of scheduling for the activities within both the undercut and extraction level development.

The extraction level is composed of production drifts developed at regular spacing, from which crosscuts are excavated to give access to drawpoints and drawbells, which connect to the undercut level above. The design of the extraction level varies depending on different factors, most importantly the degree of fragmentation that is expected to be achieved, the undercutting strategy, geotechnical conditions and stability, expected production capacity and type of equipment to be used. There have been multiple types of layout used in block caving mines in the past including incline layouts (Laubscher, 2000); however the most popular designs with the introduction of mechanized equipment (LHDs) are the El Teniente, herringbone and offset herringbone layouts (Figure 2-2).

The herringbone layout was one of the earliest most popular designs to accommodate mechanized equipment in caving operations, as it is simple and can be mirrored for larger footprints. The main disadvantages are the large spans being created between the breakaways for each drawpoint, and that the drawpoint crosscut intersects the drawbell at an angle such that the material is not drawn uniformly (Laubscher et al., 2017). The offset herringbone addresses the stabilities issues with the herringbone layout by placing the breakaways for the drawbell crosscuts at an offset. The El Teniente layout was developed at the El Teniente mine in Chile. In this layout, the drawpoint crosscuts are developed in straight 60° lines relative to the production drifts.

Esterhuizen & Laubscher (1992) showed through numerical modeling that the El Teniente layout offers the best geotechnical stability due to the uniform draw zone spacing. Moreover, the drawbell and drawpoint brow positions provide a more uniform draw. From a speed of development and preparation point of view, Ahmed, et al. (2014) performed a discrete event simulation study based on a representative layout in which they reported that El Teniente

layout offers a 9% decrease in time required for development in comparison with the offset herringbone layout.

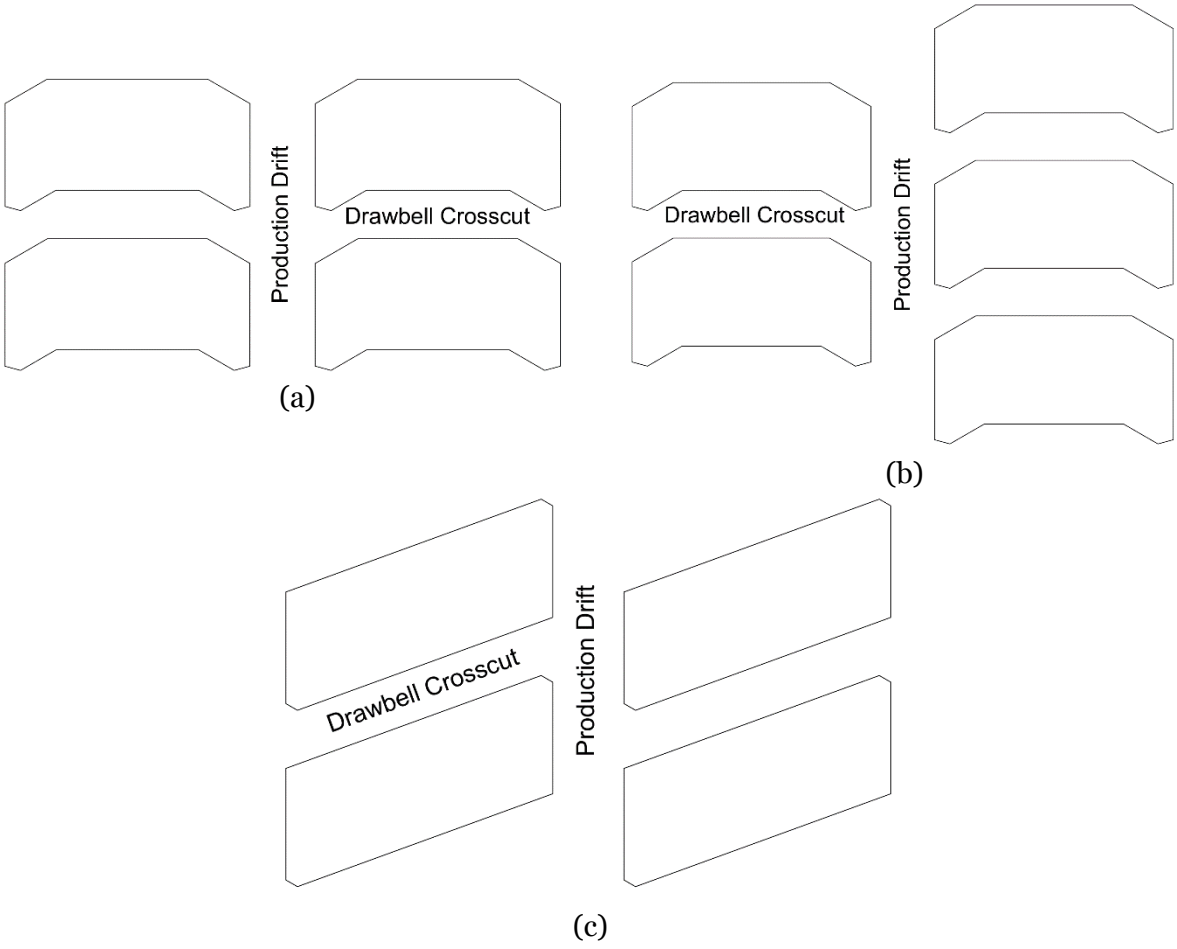


Figure 2-2. Typical (a) Herringbone, (b) Offset herringbone and (c) El Teniente layouts.

One of the main operational parameters in block caving mines is the rate of draw from each drawpoint. It is imperative to reach a balance in which the caved material is drawn fast enough such that it will not consolidate in the column above interrupting cave propagation, without leaving an excessive air gap in the cave back that could lead to geotechnical or other problems. Draw rates vary throughout the operating life of the drawpoints, with vertical rates at around 0.1 m to 0.6 m per day (DLaubscher et al., 2017).

2.1.2 Gravity Flow in Caving

One of the first theories presented to explain and model the gravity flow of granular material with application to caving methods was proposed by Kvapil (1965), building on previous empirical knowledge of granular flow in hoppers and silos (Kvapil, 1960, 1964, 1965).

This initial theory was based on the observation of simple physical models consisting of filled vertical glass sections, where the fill material usually comprised colored sand to keep track of the movement pattern. An opening at the bottom of the model is created to simulate the extraction opening in the draw columns of caving operations. From these early tests, the movement of particles due to the extraction void was identified to follow a defined shape, similar to an ellipsoid, denominated the active zone, with certain boundaries from which the remaining intact material was usually referred to as the passive zone.

Further observations by Janelid (1966), along with the result of some in-situ mine tests (Janelid, 1975) provided evidence that the ellipsoidal active zone in the caved rock was actually made up of two ellipsoidal bodies: the ellipsoid of extraction (EE) and the ellipsoid of loosening (EL). After a certain volume of material is extracted, comprising the EE, the remaining material has to cover the void created by loosening, and caving, towards the EL.

The basic geometrical relationships between the EE, the EE, and the extracted material are established in (Janelid, 1966) , with empirical evidence suggesting that the volume of the EL is about 15 times greater than the volume of the EE and in turn, assuming the same eccentricity for both ellipsoids, the height of the EL is about 2.5 times the height of the EE (Figure 2-3).

Kvapil (1982) mentions the influence of particle size distribution and shape on the form of the ellipsoids. Finer materials, such as the sand used in the early physical models, lead to slender EE and EL while coarser material creates a very broad active caving zone. Kvapil (1982) also refers to the many characteristics of the broken rock particles, such as roughness, shape and density, that influence the behavior or 'mobility' of the material and proposes a conceptual classification of the mobility of the materials related to the shape and eccentricity of the EE.

The opening or extraction width also holds a significant influence on the behavior of the caved material, as noted in (Kvapil, 1982). A wide enough extraction opening induces a phenomenon denominated as mass flow, where the gravity flow zone consists of a central part, directly above the opening, that undergoes a downwards movement as a whole column, with side sections under normal gravity flow. The mass flow phenomenon provides a more efficient way of ore extraction with larger draw rates and less dilution.

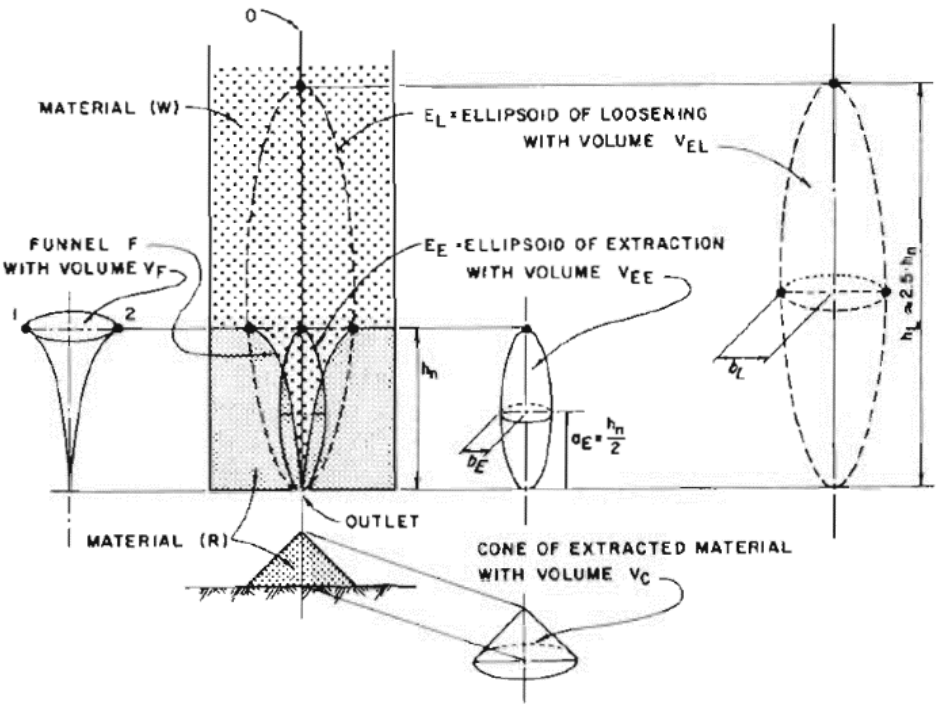


Figure 2-3. Geometrical relationships in the gravity flow of caved rock (Kvapil, 1982)

Establishing the diameters of these ellipsoids, which have a direct impact in fragmentation and productivity and serve as the basis for key design parameters such as drawpoint spacing, have been the focus of multiple experimental research efforts. Most notably, Castro (2006) performed large scale experimental tests with coarse gravel, simulating coarse fragmented rock, and showed that the extraction zone diameter could reach 28.5 m at around 100 m of draw.

There is not much information on full-scale mine tests of gravity flow at block caving operations. Alvial (1992) presented one of the first documented mine test in block caving, which was carried out at El Teniente mine with the use of markers, and concluded that the horizontal displacement of broken rock in the columns can range between 2 and 42 m, and the vertical slip angle, which describes the vertical component of the flow, ranges between 60 to 80 degrees. Brunton, et al. (2012) performed another full scale by using smart markers from 2008 to 2010 and reported that the behavior at the near field of the draw zone is chaotic and highly irregular. Garces, et al. (2016) presented interesting results on a full-scale test in a caving block at El Teniente mine. Some of the conclusions reached were that the extraction pattern holds a significant influence on the flow behavior of caving mines. If the extraction is regular from both drawpoints of a drawbell, the extraction zone was found to grow symmetrically, and a low percentage of coarse material was reported, while an irregular extraction plan would lead to the

development of a preferential flow behavior and a larger fraction of coarse material entering the drawpoint.

2.1.3 Drawpoint spacing in block caving

Laubscher (1994) discusses the practical design approaches and state of knowledge in cave mining at the time, and proposes some of the most well-known guidelines for the selection of design variables, most notably drawpoint spacing. Figure 2-4 shows the selection guide for the minimum and maximum suggested drawpoint spacing based on drawpoint width and the fragmentation size within the draw column. It was suggested that the drawpoint spacing should be at least 1.5 times the diameter of the isolated draw zone (IDZ), which is the diameter of the draw zone generated by a single drawpoint and is greatly dependent on the broken rock sizes. Still, the need for 3D physical models to further continue with the development of what he refers as poorly defined principles is strongly remarked by Laubscher.

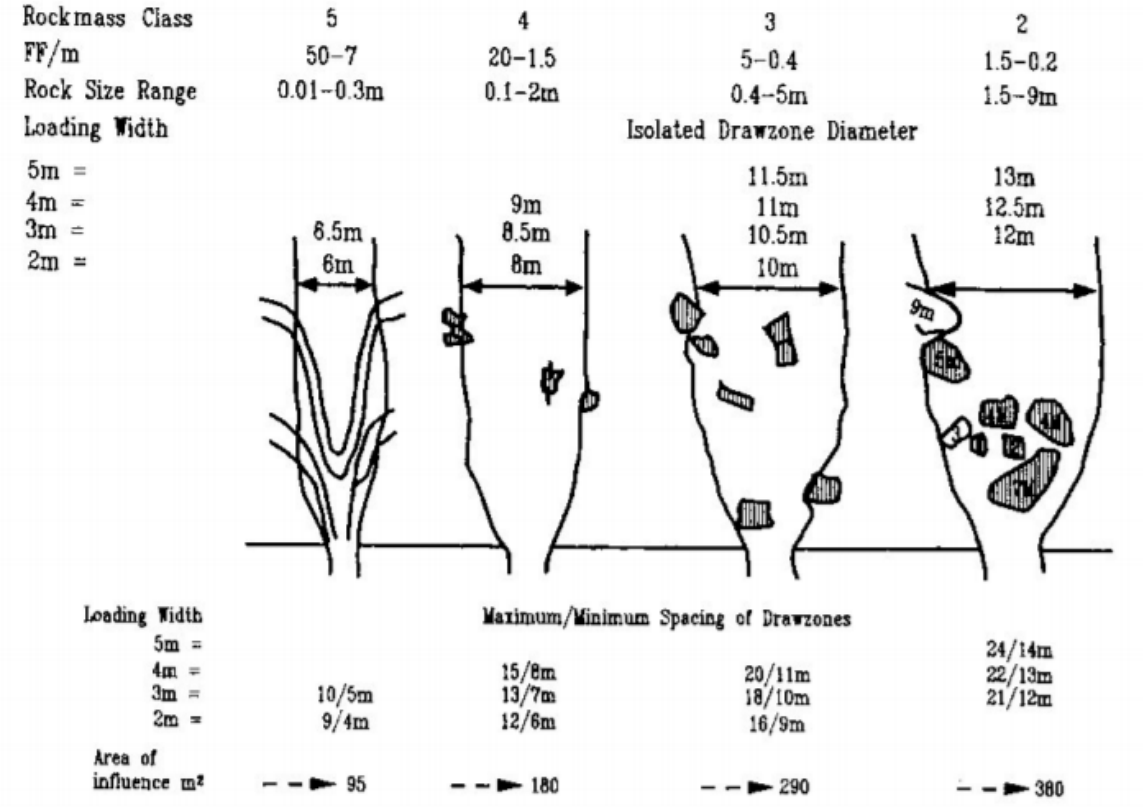


Figure 2-4. Laubscher drawpoint spacing selection guideline (Laubscher, 1994).

Halim (2006) states that Laubscher theory and guidelines for drawpoint spacing are not applicable to real mining operations, as the physical models used to develop them, based on sand, differ too much form the in-situ conditions found in caving mines. By carrying extensive

experimental tests on a large 3D physical model, some of the interesting conclusions reached were that the Isolated Movement Zone (IMZ), or the ellipsoid of loosening, was wider than the IEZ when considering the volume of material drawn. However, the IMZ was narrower that the IEZ at the same height of draw. Halim suggested that the drawpoint spacing should be so that the overlapping occurs before the draw zones reach the surface, and considering that the IMZ was found to be narrower than the IEZ for a particular height of draw, the drawpoint spacing must always be less than the width of the IEZ.

Halim found that with a drawpoint spacing less than the width of the IEZ, the draw zones do not overlap but just touch and no overdraw is produced contradicting Laubscher and previous hypothesis. Halim also found that the influence of the particle size was negligible at small heights, showing some influence after a height of around 50 m, however comments that further experimental repetitions are needed.

More effort was put on the development of 3D physical tests with gravel rather than sand to provide more realistic conditions of that on caving mines. Castro et al. (2007) tested crushed gravel with a wide (18 mm mean size) and narrow (8 mm mean size) on a large 3D physical model to evaluate the effect of the draw height and the drawpoint dimensions on the shape of the IEZ and IMZ of an isolated drawpoint. They concluded that the fragmentation size distribution and the drawpoint width have little influence on the geometry of the draw zones when considering cohesionless material, that is material with a small fraction of fines and no presence of water. The parameters that have the most control over the draw zone geometry were found to be the mass drawn and the height of draw, and that the IMZ height is controlled by the development of a stress arch zone that continuously collapses as the material is drawn. The authors, however, reported the need of more experiments with smaller fractions sizes, and at constant particle shape and friction angle, would be needed to effectively quantify the influence of the particle size fragmentation on the flow mechanism.

Trueman, et al. (2008) used the same large 3D physical model to study the influence of multiple draw zones to evaluate the interactive draw theory proposed by Laubscher. Using crushed gravel with a narrow size distribution, 86% within 9.5 – 6.7 mm passing and a mean of 8 mm, nine drawpoints were constructed in the model and drawn concurrently spaced at 740 mm corresponding to 1.2 times the estimated mean width of the IMZ at full height of draw. The authors concluded that the condition of uniform draw down and mass flow in the gravel model is achieved at a drawpoint spacing of less or equal to the IMZ, and that the previous understanding of the expansion of the draw zones would not be achieved at larger distances as suggested by Laubscher. The authors provide an explanation citing the development of

relatively large vertical stress state in the sand model experiments in conjunction with the fact that the shear strength of the sand is less than that of the gravel, as no collapse of unmoved zones between the draw zones was observed in the model.

2.2 Mine Production Scheduling in Block Caving

2.2.1 Mathematical Programming in Caving Production Scheduling

Mathematical programming is a branch of operations research that comprises a collection of methods and techniques to optimize a system that is modeled through a set of mathematical expressions (Sinha, 2006; Williams, 2013). Mathematical programming methods concern the optimization (maximization or minimization) of a single or multiple objective functions, defined in terms of decision variables, subjected to a set of constraints expressed as mathematical inequalities or equalities in terms of the decision variables as well.

The general form of a mathematical programming model can be expressed as:

$$\begin{aligned} &\textit{Maximise} && f(X) \\ &\textit{Subject to} && g_i(X) \le 0 \qquad i = 1, 2, ..., m \\ & && X \ge 0 \end{aligned}$$

Where the vector $X = (x_1, x_2, ..., x_n)^T \in R^n$ contains the decision variables and the set of constraints $g_i(X)$ are all real-valued functions of X. If the objective function and set of constraints are all expressed as linear equations, it is denominated a linear programming model, with variations such as integer programming (integer decision variables), mixed integer programming (integer and continuous decision variables) and nonlinear programming (non-linear objective functions and/or constraints) to provide better representations of certain types of systems.

The applications of operations research techniques in mining date back to the 1960s, with linear optimization methods in particular being widely adapted to solve ultimate mining limits and production scheduling problems The tractability of mathematical models is one of the main concerns as practical applications to scheduling problems or mining systems can easily become very complex. With the advances and developments in computing power, the application of mathematical programming to solve real mining problems has become more widespread (Askari-Nasab, et al. 2011; Newman, et al. 2010).

One of the first documented applications of mathematical programming is presented in Song (1989), in which a MILP model was developed along with a caving simulation process in

order to optimize the mining sequence and production schedule with the objective of minimizing cost at the Tong Kuang Yu mine in China. Chanda (1990) developed a simulation and MILP model to optimize the production scheduling of drawpoints for production at the Chingola mine in Zambia. The model was developed at a shift time scale with the objective of minimizing the deviation of average grades over the shifts. Guest, et al., (2000) provided a scheduling MILP model with insights into its application in an industrial setting. In this model, the objective function is to minimize the mining of waste as it is mentioned that this will lead to the maximization of the NPV of the mine. The constraints considered in this model are draw rates and geotechnical constraints between adjacent columns, ore flow capacity constraints and metallurgical constraints. The model was developed for a diamond mine in South Africa.

Rubio (2002) developed a MILP model that aims for the maximization of the NPV and the mine life for block caving mines. The model is applied at a column and slice resolution for long-term planning. The main constraints considered are the undercut rate, the undercut sequence precedence, the opened area at the undercut level, and the draw rate. The concept of opportunity cost is also incorporated to account for the concept of definition of reserves, in which each column pays an extra cost for the delaying of the opening of adjacent drawpoints. Draw control is also introduced by controlling the angle of draw between adjacent columns. The algorithms were applied to two different mines obtaining better results relative to a base case production schedule.

Other similar applications of mathematical programming methods in block caving mines appear in Alonso-Ayuso et al. (2014); Diering (2004); Khodayari & Pourrahimian (2014); Malaki, et al. (2017); Nezhadshahmohammad, et al. (2017); Parkinson (2012); Pourrahimian, et al. (2013); Rahal, et al. (2003); Smoljanovic (2012); Weintraub, et al. (2008).

Different approaches have been proposed to improve the computing times of solving a complex mathematical model resulting from the modeling of a block caving mining system. Weintraub et al. (2008) developed a priori and a posteriori clustering procedures to reduce the problem size while keeping it feasible when disaggregated. The general model structure divides the mine into columns, which are then divided in blocks that define the basic unit of extraction. A priori aggregation consists of the clustering of blocks according to similarities in tonnage, grade and extraction rate, using a K-means clustering algorithm. For a particular mine in Chile, they showed that this aggregation procedure reduced the problem size by 90% and the execution time by 73,68%. The a posteriori approach consists of the clustering of columns based on a fixed weight combination. A posterior approach reduced the problem size by 15% and improved the solution time by 88%.

Pourrahimian et al. (2013) presented a multi-step clustering procedure to reduce the computing times in the long-term planning of block caving mines. Three consecutive "levels" or resolution scales were considered, with the solution of each one used to guide the following more detailed problem. The mining system considers a production layout at the drawpoint and slice model, where the main decision is the definition of the extraction period and tonnage drawn from each slice. The initial level clusters the columns based on a similarity index defined by tonnage, grade and physical location. A mathematical model is then formulated that considers the extraction from each cluster, rather than individual columns. The period at which each cluster is opened and the cluster life, are used to define the earliest period and maximum life of the drawpoints that are contained within, with some flexibility added. The second level considers the drawpoint scale, with no slice division, and uses the solution obtained in the previous one to reduce its size. In a similar way, the results obtained at this scale are used to guide the third and more detailed level at the drawpoint and slice resolution. This procedure significantly reduces the computing time required to solve the model, while preserving the feasibility and optimality of the solution.

Nezhadshahmohammad, et al. (2018) presented a multi-index clustering algorithm to reduce the size of MILP models for the long-term planning of block caving mines. Draw columns are aggregated into clusters based on the center-by-center distance, grade distribution, maximum draw rate and advancement direction. Based on a search radius, each draw column is initially considered as a cluster, and similarity values are calculated between each cluster based on the mentioned parameters. The most similar pair of clusters is merged and the procedure is repeated until a desired maximum number of clusters are reached. The procedure reduced the number of variables in the case study presented by 90.4% and the solution time (at a 9% optimality gap) and the solution time from 2 hours to 10 seconds, with an NPV within less than 1% difference. At tighter optimality gaps the original model was not able to reach a solution while the clustering algorithm provided a schedule within 5 minutes.

2.2.2 Incorporation of Uncertainties in Block Caving Planning

Incorporation of geological uncertainty in mine planning workflows has been showed to offer good results in terms of managing the risk associated with the estimation of geological attributes in mineral deposits for life-of-mine and short-term planning (Dimitrakopoulos, 2011).

The stochastic optimization workflow starts with the generation of multiple numerical deposit models through the use of, most commonly, SGS techniques (Rossi & Deutsch, 2014).

Each realization represents an equi-probable scenario for the variable being modeled, from metal grades to density or other characteristics. The multiple scenarios are then used as input to produce a single mine schedule that produces the maximum expected economic benefit (NPV) subjected to operational constraints. The risk is represented as the potential deviations from production and/or material quality targets that can arise with a given extraction plan, due to the multiple deposit scenarios. Stochastic mathematical programming techniques are used, minimizing these deviations to obtain a robust production schedule (R. Dimitrakopoulos & Ramazan, 2008; Ramazan & Dimitrakopoulos, 2013).

The incorporation of geological uncertainty in block caving planning has been very limited. Dirkx et al.(2018) presented a comparative study for a caving footprint in which a production schedule was generated for a deposit model estimated with kriging techniques, a production schedule for a set of ten grade realizations for the same deposit simulated with SGS, and another schedule scenario for the metal grade simulations adding hangup uncertainty. The production schedules were obtained through a stochastic mathematical programming model. The results provided suggest that the "conventional" schedule, obtained with the kriging estimated deposit, is rendered unfeasible when subjected to the grade and hangup scenarios. The incorporation of grade and hangup uncertainty provided a more realistic and feasible estimate of the project NPV.

Sepúlveda, et al. (2018) proposed a methodology to incorporate grade and geometallurgical uncertainty in block caving mine planning. Geometallurgical uncertainty is referenced as the predicted metallurgical responses (and its uncertainty), such as grindability and recoveries, due to the uncertain geological variables as rock types and grades. The optimization framework is a bi-objective in which the first goal is to maximize the Net Smelter Return (NSR), including penalties for deleterious elements, and different measures of risk were tested as the second optimization goal, including volatility, Value at Risk (VaR), Conditional Value at Risk (CVaR) and deviations from planned production targets. The framework was solved using a genetic algorithm. They concluded that the maximization of the NSR and VaR combination with the minimization of the deviations from planned production targets yielded the best results.

Both studies show the value of incorporating different sources of uncertainties that can arise in block caving production schedule to offer a more reliable NPV estimate and production schedule.

There is little documentation on explicit incorporation of material flow and mixing in block caving mine planning algorithms. Khodayari (2018) developed a mathematical programming

model, analogous to a stochastic optimization, in which multiple scenarios are generated that are representative of the potential dilution due to the flow of particles within the caving environment. The model is developed for a production layout at a drawpoint and slice scale. Material flow simulations are generated based on a cone of movement that is defined from a horizontal displacement and vertical slip angles. These parameters are commonly used to characterize the flow of broken ore on operating caving mines. Each slice, as it is extracted, leaves a void that can be filled by any other slice that is within its cone of movement, therefore generating multiple possibilities. Rather than trying to calculate or estimate probabilities, a sampling approach is performed and multiple scenarios are generated, each one representing the drawpoint and slice model with different grades updated based on the random sample from the cone of movement concept. The key points to note are that the flow is modeled based on slices, which means that either a full slice is assumed to fill a void or not. No portions or percentages of different slices are considered as candidates to update the diluted model. Also, since only slices are considered, no waste material in the boundaries of the layout is included in the mixing simulation.

Once a set of scenarios are generated, a MILP model is defined to obtain a schedule that maximizes the NPV of the mine while minimizing the deviations in average grade on a period basis from a defined target. The model was tested on a case study and benchmarked against a PCBC generated schedule, with the same parameters. The incorporation of material flow based on this concept produced schedules with an NPV 4% to 11% higher depending on the number of constraints and detail considered.

The impact of geological, material flow and other sources of uncertainty in block caving planning is not yet well documented. Moreover the incorporation of these into a single planning and scheduling algorithm would help decision makers make more informed decision at earlier stages of the project. In this research a method is developed to integrate geological and material flow uncertainty in a mathematical optimization model to generate a robust LOM schedule. Moreover, the impact of these sources of uncertainty is quantified to get a better understanding.

2.3 Summary and Remarks

Block caving mining is presenting itself as a very attractive massive mining method that can achieve the economics of large scale open-pit mines, with the possibility of accessing lower grade and deeper orebodies thanks to lower operating costs. However, due to the operating principles behind the gravity flow of broken rock, block caving projects require careful and detailed planning at all stages to provide good estimates and forecasts.

A major source of risk in the development of mining projects, and later in the underperformance during operation, is the geological uncertainty. This uncertainty is unavoidable as the metal content of a mineral deposit is estimated based on very scarce samples from exploration programs. The importance of incorporating geological risk in mine planning has been widely demonstrated. Block caving projects, in particular, are also subjected to uncertainty in the metal content from the ore drawn due to the flow of broken rock through the draw columns.

Mathematical programming has been established as a practical tool to develop optimal production schedules and mine plans that maximize a particular objective while considering operational constraints. Multiple successful applications of mathematical programming models in block caving planning with different levels of detail are documented. Stochastic programming, in particular, offers a tool to integrate different sources of uncertainty and produce risk resilient schedules. However, the incorporation of uncertainty in block caving mine planning workflows is limited. The few documented applications show the value of adopting an uncertainty based approach to generate more reliable forecasts and mine plans.

This research aims to provide a method that incorporates geological and material flow uncertainty in the development of block caving boundaries and production schedule at a block model scale.

CHAPTER 3

THEORETICAL FRAMEWORK

This chapter contains the mathematical programming formulations for the boundary and production scheduling optimization of block caving mines at the block model support considering geological and material flow uncertainty. The rationale behind the structure of the objective function and the incorporation of multiple deposit simulations to arrive at a expected economic value is detailed. The considered operational constraints are explained and the structure of the different coefficient matrices is presented. To reduce the computing time, two heuristics are implemented: an early start algorithm in which the earliest possible extraction period for each MU is defined to eliminate decision variables, and a sliding time window heuristics. The production schedule defines the extraction period for each MU to maximize the expected NPV while minimizing the deviations incurred in production and grade quality targets due to the variability represented in the multiple deposit simulations.

3.1 Introduction

Geological uncertainty has been acknowledged as one the main causes for the underperformance of mining projects, and traditional approaches to mine planning fail to fully capture it in the development of mine sequences using a single estimated model. Traditional mine planning workflow uses the ordinary kriging (OK) estimation to develop a strategic schedule and plan. However, it fails to incorporate the inherent uncertainty in this estimation process as only a single estimated input is considered. The development of geostatistical simulation techniques has proven that mine plans generated considering only the OK deposit model have a low probability of actually reaching the forecasted outputs or fail to capitalize value (Deutsch, et al. 2015; Dimitrakopoulos et al., 2002) .

Due to the operating principles of block caving mines, the risk of dilution caused by the gravity flow movement of the broken rock mass constitutes a major source of uncertainty as well. Mine plans are built based on estimated grades and tonnages, and the profit generated by the extraction of ore. However, in block caving mines the extraction of material depends on the flow characteristics of the broken rock mass, and the initial deposit model is not representative of the actual grades as the "blocks" flow through the draw zone (Castro & Paredes, 2014; Esterhuizen & Laubscher, 1992; Pierce, 2010).

In the following sections, the details of the developed optimization framework for block caving mines under geological and material flow uncertainty is presented.

3.2 Geostatistical Modeling

The initial step considered for geostatistical modeling is the definition of the stationary domains, which is usually done based on rock types or other geology control such as structures or mineralized zones. Estimation is then performed separately for each domain. The general steps are:

- Declustering to obtain a representative distribution of the parameters in each stationary domain defined. This is due to the preferential sample usually followed in exploration programs to target higher-grade areas.
- Exploratory and multivariate data analysis. Summarize the main characteristics of each variable individually to get more insight into its behavior, as well as the correlation between multiple variables (when applicable).
- Transform data into Gaussian units. This step differs from traditional geostatistical modeling and is required for the assumptions that sequential

gaussian simulation techniques are based on them. A normal score transformation is the usual approach, where the cumulative distribution function (CDF) of the variables is transformed into a standard normal distribution CDF.

- Experimental variograms are calculated for the variables to be simulated. These quantify the spatial correlation of the variables in different directions. It is necessary to establish the direction of maximum continuity. The direction of maximum continuity is usually determined based on geological knowledge of the deposit, as well as calculating and evaluating variogram in multiple directions. For rock type variable (categorical variables), indicator variograms are calculated.
- A grid is defined for estimation, and the rock type variable is estimated at every node by sequential indicator simulation. This algorithm generates a set of rock type realizations that serve as the domain for the grade simulation.
- For each rock type or domain, the grades and other properties such asdensity are simulated through the sequential Gaussian simulation at every node of the grid. This is performed for each rock type realization, generating a set of conditionally simulated deposit models.

The set of geology realizations have to be checked for histogram and variogram reproduction. This ensures that while the set of simulated models represent different equiprobable numerical models, they follow the same statistical and spatial behavior.

In this research, this workflow is applied using the GSLIB software (Deutsch & Journel, 1997).

3.3 Production Units Aggregation

In this stage, the blocks are aggregated into mining units (MU), with the objective of maximizing the expected metal content contained within, for scheduling purposes. Due to the operating conditions of block caving mines, the MU aggregation is on a column-wise scheme based on a reference undercut elevation from which the overlying material comprise the mineral reserves.

To simplify this procedure, the columns are represented as production unit (PU) in a plan view at the undercut section, based on drawpoint spacings. These PUs span the columns up to a certain height, which at this stage is determined either by reaching a predefined maximum height, or once a dilution percentage limit is surpassed to account for the deposit ore-waste boundaries. Dilution is evaluated based on the potential MU from each PU, as the percentage fraction of waste tonnage in relation to the total tonnage of the unit.

As mentioned previously, metal content is selected as the optimization objective rather than economic value in order to give the later scheduling model the decision of boundaries such as BHOD while considering the whole mining system.

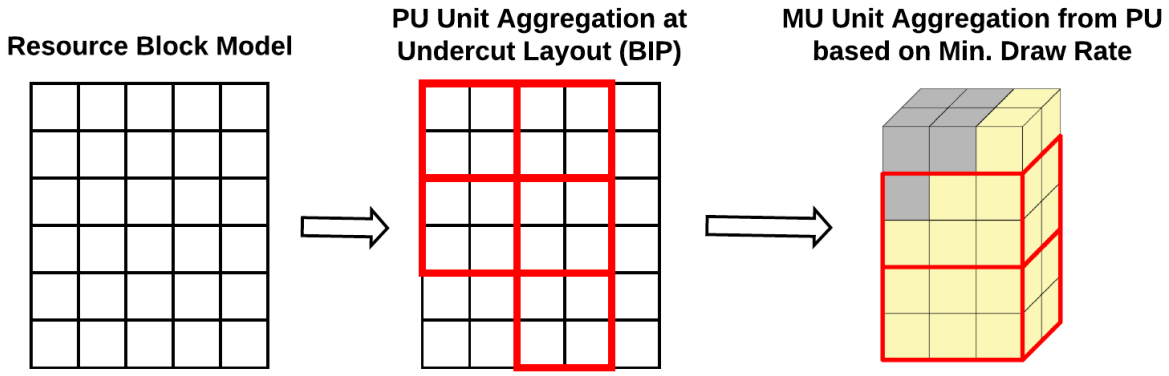


Figure 3-1. Schematic representation of the aggregation procedure.

3.3.1 Model Assumptions and Notation

The following assumptions are used in the formulation of the BIP model for the layout aggregation:

1. The aggregation is done to the highest integer of the desired drawpoint spacing in relation to the block size. Reblocking of the resource model can be performed to achieve the desired level of detail.
2. Each PU contains multiple MU that span vertically up to a certain maximum height or maximum dilution %. The metal content and dilution % evaluation is carried out on MU aggregates at a time for each column.
3. No mixing or dilution is considered at this stage.
4. The expected metal content is calculated as the average metal content amongst all realizations, this is used for the coefficient calculation of the stochastic model. For kriging mine plans, the kriging metal content is used.

A general overview of the notation, including indices, sets and decision variables, used in the aggregation step is presented in Table 3-1.

Table 3-1. Notation used in the aggregation optimization step

Indices	
$c \in \{1,...,C\}$	Index for all possible production units
$b \in \{1,...,B\}$	Index for all possible blocks in a given undercut level that are part of a PU with non zero metal content
Sets	
O^b	Set containing all PUs that overlap with block *b*, with number of elements $N(O^b)$
Parameters	
M_c	Metal content of production unit *c*
Decision Variables	
$x_c \in \{0,1\}$	Binary variable. Takes the value of 1 if unit *c* is included in the layout and 0 if not.

3.3.2 Objective Function

In initial aggregation step, the objective is to maximize the expected metal content contained within the selected mining units, in order for the later scheduling model to decide the best extraction sequence and boundaries. As mentioned previously, economic value is not considered at this stage to avoid making decisions based on individual columns (BHOD selection) that does not consider the whole system.

The decision variables are considered binary to represent the selection or not of a given PU to be included within the potential footprint of the mine.

$$x_c \in \begin{Bmatrix} 1, & if\ PU\ is\ selected \\ 0, & otherwise \end{Bmatrix}$$

Each PU is the aggregation of blocks over the drawpoint spacing between drifts and across the minor pillar on a plan view. Then for the scheduling purpose, the MUs are created with aggregating blocks inside the PUs based on the desired minimum draw rate. For a particular layout, each PU will contain a certain number of MU that represent the column above it up to either a specified maximum height of draw, or the defined maximum acceptable dilution %. The expected metal content associated with each MU is then calculated as the sum of the expected metal content of the blocks that form the MU (average across all realizations). The expected metal content of each PU is calculated as the sum of the expected metal content of all the MUs within the PU. The PU metal content serves as the objective function coefficients. Equation (3.1) shows the objective function.

$$Max \sum_{c=1}^{C} E[M_c \cdot x_c] \tag{3.1}$$

3.3.3 Constraints

The main constraints to consider for this model are the maximum and minimum column heights, and to generate a layout with no gaps between PUs while selecting only non-overlapping PUs.

The maximum and minimum column height considerations are implicitly included during the calculation of the metal content of each PU, by setting the total metal content of PUs that do not reach the minimum height (due to dilution or geometry of the mineral deposit) to 0.

To guarantee a continuous layout with no gaps between PUs at the undercut section, all individual blocks in the section that are within a PU and has a non zero metal content value have to be included. This will guarantee that the whole deposit is considered at this stage, while assuring a continuous layout. Equation. 3.2 shows the constraint to avoid overlapping PUs while generating a continuous PU layout.

$$\sum_{c=1}^{N(O^b)} x_c = 1 \qquad \forall b \in \{1,\ldots,B\} \tag{3.2}$$

3.4 Material Flow Simulation

Material flow simulation is based on a cone of movement defined by a horizontal displacement (HD) and vertical slip angle (VSA) to model the potential movement of broken rock mass. This concept is adapted from Khodayari (2018) and Khodayari and Pourrahimian (2019).

For each PU, the cone of movement is initially placed at the bottom of the lowest MU after an established point of entry dilution, the point of entry dilution defines the height at which draw interaction between adjacent drawpoints occurs. Figure 3-2 shows the cone of movement with the balls representing the center points of the blocks from the resource model. For each MU, a set of candidate blocks is defined as those blocks whose center points lie within the cone.

From this candidate blocks, random blocks are drawn forming a sample set. This sample set keeps drawing random blocks from the candidate set until the total tonnage contained matches the tonnage of that particular MU (considering a small tolerance to account for density differences between ore and waste). This means that the MU has been “filled” with blocks from

the resource model that can potentially show up when it is extracted. The procedure is repeated for every MU, flagging "used" blocks to avoid having the same individual block fall in MU that have overlapping cones (adjacent or overlying ones).

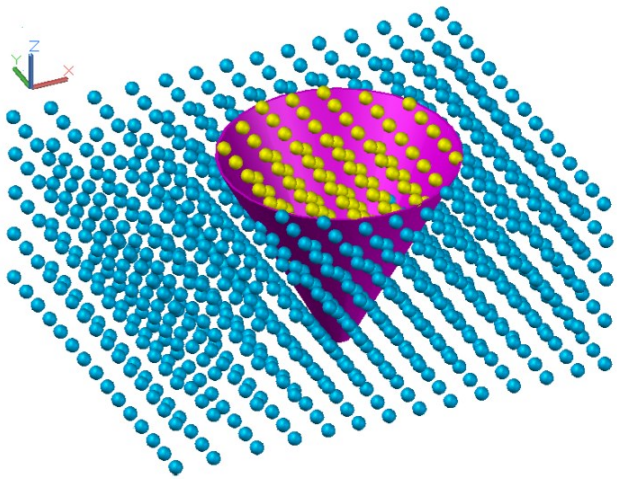


Figure 3-2. Material flow simulation (balls represent center of blocks from the resource model). The cone is placed at the bottom of each MU and draw a random set of blocks from those within it, until the tonnage of the MU is reached.

This model also accounts for dilution at the boundaries of the footprint as waste blocks outside the potential layout are also considered based on the cone concept. The grade and tonnage of each MU are then calculated as the sum of the tonnages, and the tonnage-weighted average grade, from the sample set drawn for it.

In contrast with the formulation presented by Khodayari (2018) and Khodayari and Pourrahimian (2019), in this research the material flow simulation is performed at a block model scale rather than slice scale to update the aggregated units, which gives more detail to the potential mixing within the cave especially at the boundaries.

Since there is a number of deposit realizations, flow simulations are carried out for each of them. Therefore, a set of grade and material flow scenarios at the MU aggregation levelare generated for scheduling purposes.

3.5 Life-of-Mine (LOM) Production Scheduling Incorporating Geological and Material Flow Uncertainty

Each MU has a set of possible tonnages and grades for each geology-material flow simulation performed, that represent the uncertainty based on the estimation procedure and behavior of block caving mines. The following section details the formulation of an optimization model that takes as an input the whole set of grade-tonnage scenarios, rather than a single estimate, and

generates a mine plan that maximizes the NPV of the project while minimizing the potential deviations due to the mentioned uncertainties.

3.5.1 Model Assumptions and Notation

The following assumptions are made in the production scheduling optimization model.

- The decision to extract a MU is binary, which means that for each period it will decide whether to extract the whole unit or not.
- No hard-fixed cut-off grade is considered. The optimization model decides whether to initiate extraction within a PU and up to which height based on its overall discounted expected profit contribution to the mine plan NPV.
- To estimate expected values in the objective function coefficients, arithmetic average is used due to the assumption that the geological simulations are equi-probable realizations.
- Uncertainty is assumed to impact production targets and grade quality per period, and the model is developed to minimize deviations in these outputs only.

A general overview of the notation including indices, sets, and decision variables is presented in Table 3-2.

Table 3-2. Notation for indices, sets and decision variables.

Indices	
$u \in \{1,\ldots,U\}$	Index for all mining units
$t \in \{1,\ldots,T\}$	Index for all periods
$c \in \{1,\ldots,C\}$	Index for all production units
$s \in \{1,\ldots,S\}$	Index for all geological and material flow realizations
u_c^1	Index for the first or lowest mining unit within the production unit c
Sets	
C^c	Set containing the mining units that are within the production unit c. Each set has a total number of elements of $N(C^c)$
$V^{u,c}$	Single element set containing the mining unit directly below unit u of the production unit c. Used for vertical precedence constraints
$H^{u,c}$	Set containing the production units that have to be opened before the extraction of the mining unit u of the production unit c, based on the mining direction. Each set has a total number of elements of $N(H^{u,c})$

A^c	Set containing all adjacent production units to the unit c for cave back slope constraint. Each set has a total number of elements of $N(A^c)$
Decision Variables	
$z_u^t \in \{0,1\}$	Binary variable controlling the decision to extract the mining unit u of production unit c in period t
$tonDev_{t,s}^{+} \in [0,\infty]$	Continuous variable representing the ore tonnage "positive deviation" (overproduction) incurred on period t by deposit simulation s
$tonDev_{t,s}^{-} \in [0,\infty]$	Continuous variable representing the ore tonnage "negative deviation" (shortage) incurred on period t by deposit simulation s
$gradeDev_{t,s}^{+} \in [0,\infty]$	Continuous variable representing the grade quality "positive deviation" (over upper bound) incurred on period t by deposit simulation s
$gradeDev_{t,s}^{-} \in [0,\infty]$	Continuous variable representing the grade quality "negative deviation" (under lower bound) incurred on period t by deposit simulation s
Parameters	
MetalPrice	Metal price per ton of metal ($/t)
p^+	Cost for positive deviation (overproduction) of ore tonnage ($/t)
p^-	Cost for negative deviation (shortage) of ore tonnage ($/t)
q^+	Cost for positive deviation (over upper bound) of grade quality range ($/t)
q^-	Cost for negative deviation (under lower bound) of grade quality range ($/t)
$\mathrm{Re}c$	Metallurgical recovery of the processing operations (%)
g_u	Grade of the mining unit u of production unit c (g/t)
Ton_u	Tonnage of mining unit u of production unit c (t)
$S_{u_c^1}$	Planar surface area of production unit c (m^2), based on the first (lowest) MU of the PU
$Height_u$	Height of mining unit u (m)
MC	Mining cost per ton of ore in period t ($/t)
PC	Processing cost per ton of ore in period t ($/t)
$OpCost_c^t$	Development cost of production unit c, accounted for when extracting the first (lowest) MU, u_c^1, of production unit c ($)
i	Discount rate (%)
$\overline{M}^t$	Ore production target in period t (t)
$\overline{G}^t$	Average grade upper bound on period t
$\underline{G}^t$	Average grade lower bound on period t
$\underline{DR}$, $\overline{DR}$	Minimum and maximum Draw Rate of production units per period (t/period)
$MaxOpRate^t$	Maximum undercutting rate (m^2/period)

$MaxAbsDiff_c^{c'}$	Maximum allowable absolute difference in height between production unit c and each member c' of its adjacent set A^c to maintain cave back slope constraints
$EPGAP$	Relative gap for the branch and bound MILP solution method

3.5.2 Objective Function

Different strategic goals can be considered when building a mine plan. For long-term mine plans however, the usual target is to maximize the total NPV of the project. The NPV of the mine plan comprises the summation of the discounted profit from the extraction of ore throughout the different periods of the mine life.

Geological and material flow uncertainties are incorporated in the model by the use of deviation variables. These deviations have associated costs, set by the user to control the flexibility of the outcome mine plan, which is discounted by a rate frequently denominated geological discount rate g. The geological discount rate serves a similar role to the economic discount rate in that it reduces the cost of production deviations in the later periods of the mine life, therefore prioritizing the achievement of targets in the initial stages of the project.

The stochastic formulation presented in this research takes as an input the generated s geological and material flow MU scenarios and generates a single mine plan that maximizes the expected NPV of the project while minimizing the incurred deviations in ore production and grade quality from their target and bounds respectively.

The NPV of the extraction of a mining unit u in period t is estimated as the revenues, $\mathrm{Re}v_u$, obtained from selling the metal contained within minus the mining cost, MC_u, and processing costs, PC_u, discounted to that particular period, based on discount rate i (Equation 3.3).

$$NPV_u^t = \begin{cases} \dfrac{\mathrm{Re}v_u - MC_u - PC_u}{(1+i)^t}, & if\ \mathrm{Re}v_u > PC_u \\ \dfrac{-MC_u}{(1+i)^t}, & \text{otherwise} \end{cases} \tag{3.3}$$

e Equation 3.4 shows the revenue calculation.

$$\mathrm{Re}v_u = (\mathrm{MetalPrice} - \mathrm{SellingCost}) \cdot \mathrm{Rec} \cdot g_u \cdot Ton_u \tag{3.4}$$

Each geological and flow simulation generates a grade and tonnage realization for each MU, building a set representing the associated uncertainty. An NPV is calculated for each grade

realization, and the average between this set is taken as the expected NPV. Figure 3-3 shows an example of the expected NPV calculation for a MU.

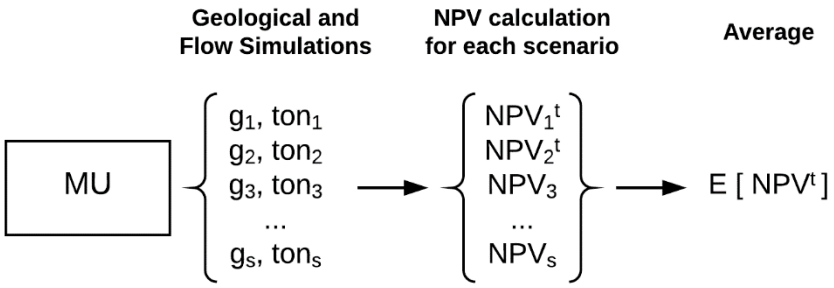


Figure 3-3. Expected NPV calculation from the geologicaly and flow simulations of a MU.

An additional cost is considered to account for the preparation and developing of each PU. The "opening" of drawpoints require the excavation of infrastructure such as tunnels for haul equipment access as well as the drawbell. In this formulation, the development costhe t of production unit c, $OpCost_c^t$, is "charged" to the extraction of its first or lowest MU. This means that for cost calculations the model implicitly assumes that once a draw column is developed it enters continuous production, and there is no delay between the activities. This development cost is an input value and is discounted based on the economic rate.

Deviation variables are used to account for the impact of the uncertainty scenarios in ore production and grade quality. A cost is associated with the deviations incurred from a desired ore production target and grade quality bounds, which is input by the user in order to control the flexibility and risk profile of the mine. As mentioned previously, these costs are discounted by the concept of a "geological discount rate", g, which serves to prioritize the minimization of deviations during the early periods of the mine life. The expected deviation cost is considered as the average total cost amongst all the S scenarios. Deviation variables are considered for both over and under production cases in ore tonnage and grade quality, with costs assigned individually giving the user the option to prioritize or ignore any of them. There is a deviation variable per each period and scenario as well, to account for the uncertainty over the whole mine life.

The objective function is presented in equation 3.5. The first term is the described expected NPV from the extraction of the MU in a particular sequence, the second term comprises the opening or development cost for each PU, and the third term accounts for the expected cost from the deviations in ore production targets and grade quality bounds amongst the realizations.

$$Max \quad \sum_{t=1}^{T}\sum_{u=1}^{U} E[NPV_u^t]\cdot z_u^t \; - \; \sum_{t=1}^{T}\sum_{c=1}^{C}\frac{OpCost_c^t \cdot z_{1,c}^t}{(1+i)^t} \; - \; \frac{1}{S}\sum_{s=1}^{S}\sum_{t=1}^{T}\frac{(p^+ \cdot tonDev_{t,s}^+ + p^- \cdot tonDev_{t,s}^- + q^+ \cdot gradeDev_{t,s}^+ + q^- \cdot gradeDev_{t,s}^-)}{(1+g)^t} \tag{3.5}$$

3.5.3 Constraints

The following set of constraints are included in the formulation:

- Mining capacity target

This constraint controls the total mining capacity on a period basis. It ensures that the total tonnage of material extracted in each period is as close as possible to the desired target amongst all the realizations.

- Grade blending range

This constraint ensures that the average grade of the material extracted in each period is within the desired range, by the definition of lower and upper bounds, amongst all the realizations.

- Maximum draw rate

This constraint guarantees that the vertical extraction rate from each PU is less or equal than a maximum draw rate established.

- Undercut development rate

This constraint controls the number of PU units opened on each period, reflecting the maximum undercut area that can be developed based on an established capacity.

- Maximum adjacent height of draw

This constraint ensures that the relative height of draw between adjacent columns is within a certain specified range, which is established as number of MU in order to be coherent with the aggregation level, to control the cave back slope during the mine life.

- Mining precedence

This constraint forces the sequence to be feasible based on the starting position and mining advancement direction. Two types of precedences are defined: horizontal precedences which control the opening of PU on the undercut layout, and vertical precedences which control the extraction of MU within each PU.

- Reserves

This constraint ensures that each MU is either extracted once or not extracted at all.

All the constraints except the production target are built as inequalities defined by a lower or upper bound, while the production target constraint is modeled as equality. The general structure of the inequality and equality constraint matrices are shown in Figure 3-4 and Figure 3-5 respectively.

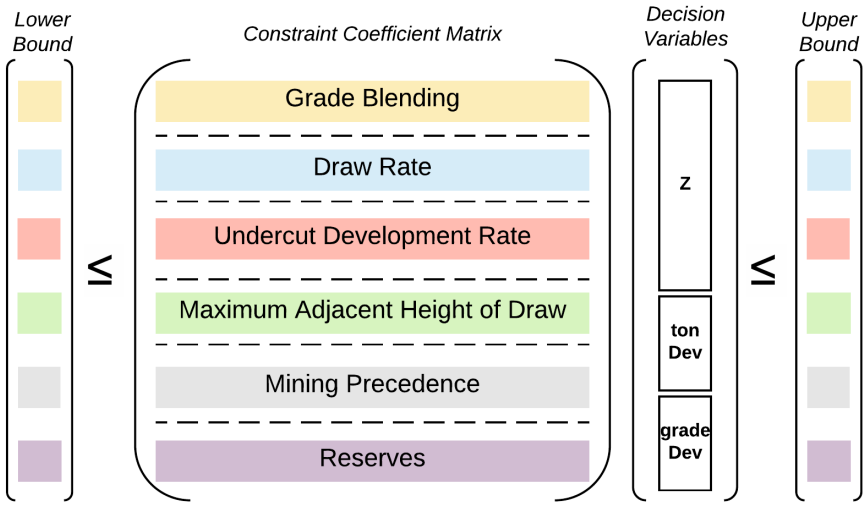


Figure 3-4. Coefficient matrix general structure for the inequality constraints.

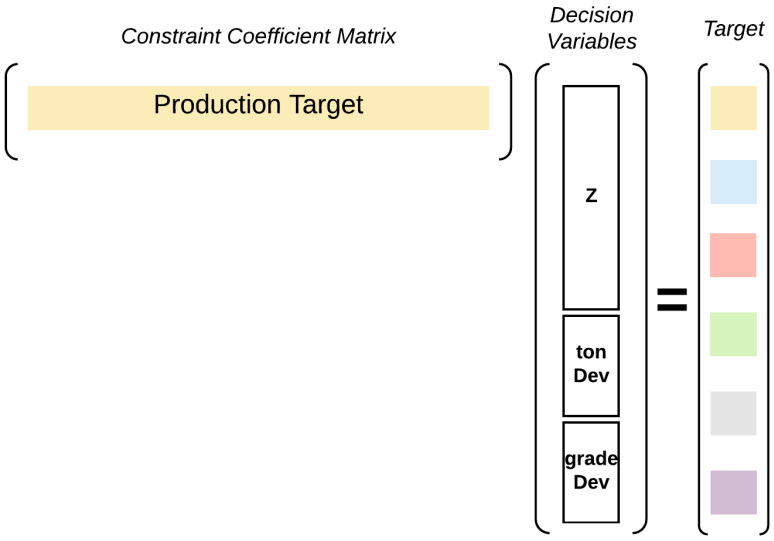


Figure 3-5. Coefficient matrix general structure for the equality constraints.

Each constraint represents a number of rows of the coefficient matrix depending on what it is representing. More detail on the coefficients and variables of each individual constraint is given in subsequent sections.

The constraints coefficient matrices are further divided into different areas based on the defined decision variables, and ordered to match the decision variable vector, as shown in Figure 3-6. The coefficients for the variables regarding the decision of extraction of a MU, z, are placed in the first columns of the constraint coefficient matrix and are structured initially by the index for MU, $u = 1,2,...,\mathrm{U}$, then by period, $t = 1,2,...,T$. After this, the deviation variables are included starting with ore tonnage deviation and then the grade quality deviation. There is a distinct variable for over and under production in both ore and grade quality deviations. The deviation variables are structured initially by period, $t = 1,2,...,T$, and then by geology and material flow simulation, $s = 1,2,...,S$. The only constraints that have non-zero coefficients in the deviation variables are the production target and grade quality, as it is assumed that uncertainty affects these two KPIs. The other constraints have zero coefficient variables in this section of the matrix.

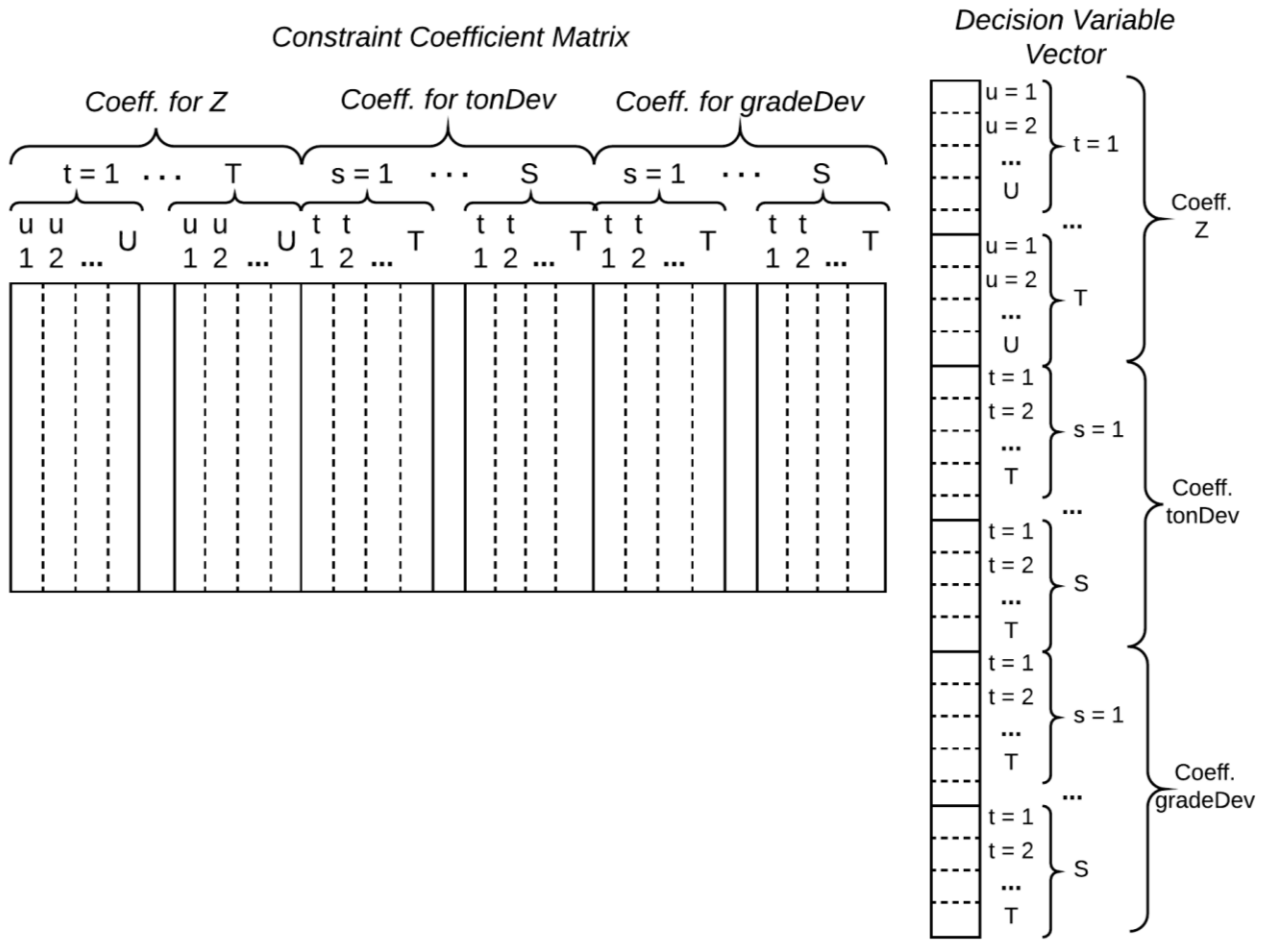


Figure 3-6. Structure of each variable in the constraint coefficient matrix and decision variable vector.

3.5.3.1 Mining Capacity Target

The mining capacity target forces the solution to be as close as possible to a defined production level in a period basis for the mine life. The constraint is implemented for all the geological and material flow realizations to represent the uncertainty in the ore tonnage estimation, and is modeled as equality with the implementation of deviation variables to balance it. These deviation variables reflect the potential shortage or overproduction due to the variability across the simulated deposit models. The constraint is presented in equation (3.6).

$$\sum_{u=1}^{U} T_u \cdot z_u^t - tonDev_{t,s}^{+} + tonDev_{t,s}^{-} = \overline{M}^t \qquad \forall t,s \tag{3.6}$$

The constraint forms $T \times S$ rows of the equality coefficient matrix, where it is the single constraint as all the others are defined as inequalities.

The definition of the right-hand-side (RHS) values reflect the user desired ore production target on a period basis, this gives the flexibility to define ramp-up and ramp-down periods.

3.5.3.2 Grade Blending

This constraint ensures that the average grade from the material extracted in a period is as close as possible to the desired range. This user-defined range reflects processing stream operating requirements as well as metal content targets on a period basis. The constraint is implanted for all the geological and material flow realizations to represent the uncertainty and variability in the metal grade estimations. A positive deviation variable is used to control average production grades over the desired upper bound, and a negative deviation variable is used to control average production grades under the desired lower bound. The constraint upper and lower bound are presented in equations (3.7) and (3.8) respectively.

$$\frac{\sum_{u=1}^{U} g_u \cdot Ton_u \cdot z_u^t}{\sum_{u=1}^{U} Ton_u \cdot z_u^t} - gradeDev_{t,s}^{+} \le \overline{G}^t \qquad \forall t,s \tag{3.7}$$

$$\frac{\sum_{u=1}^{U} g_u \cdot Ton_u \cdot z_u^t}{\sum_{u=1}^{U} Ton_u \cdot z_u^t} + gradeDev_{t,s}^{-} \ge \underline{G}^t \qquad \forall t,s \tag{3.8}$$

The constraint forms $T \times S$ rows of the inequality coefficient matrix. The deviation variables reflect the shortage or overproduction of tonnage of metal, $Ton_u \cdot g_u$, in reference to the established upper and lower bounds. This assumption eases the computational implementation of the constraints.

3.5.3.3 Maximum Draw Rate

This constraint ensures that the extraction rate from each PU does not exceed an established limit. Drawing at larger rates can lead to operating and safety problems as the air gap between the muckpile and the cave back increases rapidly, while drawing at slower rates can lead to the consolidation of the material within the column and the requirement of secondary breakage methods. The constraint is implemented for the upper bound, as the minimum draw rate requirement is already met in the aggregation step of the MU. The constraint is presented in equation (3.9).

$$\sum_{c=1}^{N(C^c)} T_u \cdot z_u^t \leq \overline{DR} \qquad \forall c,t \tag{3.9}$$

The constraint is applied for each PU at every period, forming $C \times T$ rows in the coefficient matrix. The maximum draw rate limit is applied to the PU which is the set of two drawpoints that draw material from the same drawbell, based on the presented aggregation scheme.

3.5.3.4 Undercut Development Rate

This constraint controls the maximum number of PUs that can be opened per period. The undercut development rate is a common parameter used in the planning of block caving mines. The rate at which the number of PU that can be advanced and opened for production is restricted by the development equipment capacity, with geotechnical considerations as well to maintain a safe caving environment. The constraint is generated for every period, and works by considering the sum of the areas at the undercut layout of the first MU from each PU on the period it is extracted, constraining it to a maximum development rate expressed in m^2 of excavation per period. The constraint is presented in equation (3.10).

$$\sum_{c=1}^{C} S_{u_c^1} \cdot z_u^t \leq MaxOpRate^t \qquad \forall t \tag{3.10}$$

The constraint is applied for each period, forming a total of T rows in the coefficient matrix.

3.5.3.5 Maximum Adjacent Height of Draw

This constraint limits the maximum relative height of draw between adjacent PU at any period in order to control the cave back slope, to ensure a smooth caving profile, and minimize dilution and geotechnical risks. The MU heights are used to control the slope profile, rather than tonnages drawn, to make it more intuitive and coherent with the model formulation. The adjacent units for each PU are defined based on a circular search neighborhood as proposed by Nezhadshahmohammad, et al. (2017) to guarantee a tighter control on the adjacent draw profile between PU over the production schedule. The radius of this search neighborhood can be increased to enforce the draw profile constraint over larger areas around each PU, while the minimum suggested value would be equal to the largest dimension of the PU in order to include the directly adjacent units. The constraint is presented in equation (3.11).

$$\sum_{u=1}^{U}[Height_u \cdot z_u^t] - \sum_{u=1}^{U}[Height_u \cdot z_{u,c'}^t] \leq MaxAbsDiff_c^{c'} \qquad \forall c,t \;\; \forall c' \in A^c \tag{3.11}$$

The constraint is applied for each PU, in reference to the set of adjacent PU defined by the search radius, forming a total of $C \times T$ rows in the coefficient matrix.

3.5.3.6 Mining Precedence

This constraint controls the vertical and horizontal precedence of the different MUs and PUs respectively. The vertical precedence is controlled by limiting the extraction of a specific MU only when the MU directly below has been extracted. The lowest MU from each PU does not have vertical precedence. The constraint additionally enforces continuous mining in the PU, which is required in caving operations in order to avoid compaction of the columns, with each period extracting at least one MU equivalent to the minimum draw rate. The horizontal precedence is defined over a convex V-shape mining advancement front, which is common industry practice as it has been found that it helps to control the stresses induced as the cave front progresses. This horizontal precedence is determined based on a selected starting point and the mining front angle, by finding the first V-shape that contains the center of the PU and selecting the directly adjacent PU as precedent (Figure 3-7).

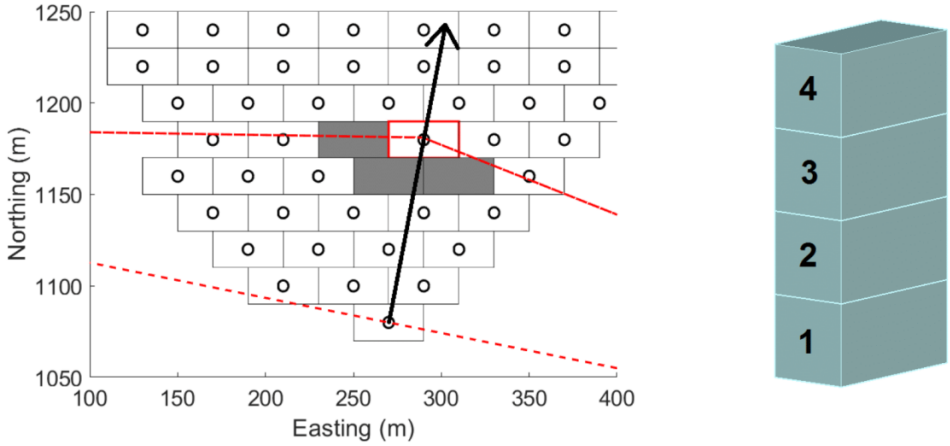


Figure 3-7. Determination of mining precedences. On the left the horizontal precedence between PUs based on the advancement direction and starting point. On the right the vertical precedence of MUs within each PU.

Equations (3.7) and (3.8) shows the vertical precedence and horizontal precedence constraints repectively. The mining precedence constraints form $U \times T$ and $C \times T$ rows in the coefficient matrix.

$$z_u^t \leq z_{V^{u,c}}^t + z_{V^{u,c}}^{t-1} \qquad \forall u,t \tag{3.12}$$

$$N(H^{u,c}) \cdot z_{u_c^t,c}^t \leq \sum_{t=1}^{T} \sum_{c=1}^{C} z_{u_c^t,c}^t \qquad \forall c,\mathrm{t} \tag{3.13}$$

3.5.3.7 Reserves

This constraint ensures that each MU is only extracted once. It also gives freedom to the model to decide whether or not a MU is extracted, providing the BHOD for each PU and the undercut footprint limits as part of the solution. The constraint is presented in equation (3.12).

$$\sum_{t=1}^{T} z_u^t \leq 1 \qquad \forall u \tag{3.14}$$

The constraint is applied for each MU, and forms a total of U rows in the coefficient matrix.

3.6 Solution Approach

The model is implemented in a MATLAB environment, using IBM CPLEX optimization engine which uses a branch-and-cut method to obtain a solution under a given MILP gap. To

further improve, the earliest start algorithm presented by Topal (2008) is implemented. The precedence constraints of the mining units, both vertical and horizontal, together with the mining capacities, maximum draw rate and relative adjacent height of draw constraints, can be used in order to establish the earliest possible period that a mining unit can be extracted. Therefore, all variables related to the extraction of the MU at an earlier period can be eliminated from the problem.

A sliding time window heuristic (STWH) is also implemented in order to further reduce the computing times and make the optimization framework more useful to mine planners. The STWH was first successfully introduced by Cullenbine, et al. (2011) to produce quick solutions for the block scheduling problem in open-pit mines, and has been implemented in further research efforts on mine sequencing optimization (Dimitrakopoulos and Ramazan, 2008; Lamghari and Dimitrakopoulos, 2016; Rimele et al. 2018) including a block caving long-term planning application by Dirkx et al. (2018). The STWH works by repeatedly solving a relaxed version of the problem, outside of a defined time window for each period, while keeping the solutions obtained fixed into the next iteration until the last time period T is solved.

A time window of size τ, with $\tau < T$, is selected and initially placed on period $t = 1$. Over this time window, the model formulation adds the constraints and variables in the complete form described. All the variables relating to time periods outside of the time window are relaxed to be continuous with the constraints added in their complete formulation. The model is solved for time period t, and the solution obtained for the variables relating to this single period are kept and fixed into the next iteration. The time window is now moved to $t+1$ with the model formulation modified accordingly. All variables within the time window τ, are again added in their regular form, with the variables outside the time window relaxed to be continuous. The process is repeated, moving the time window one period at a time until a solution is obtained for all periods (Figure 3-8).

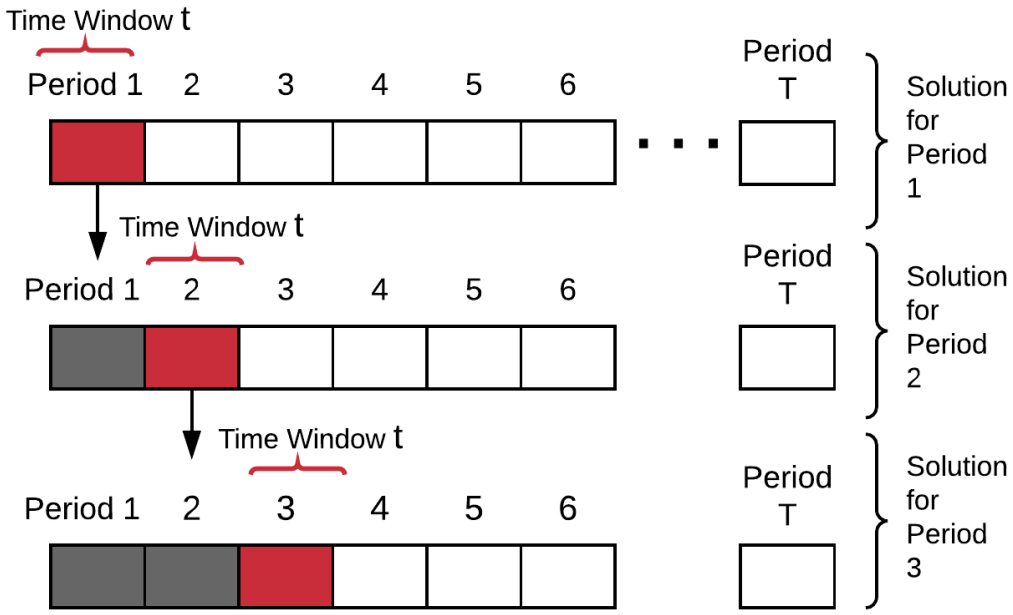


Figure 3-8. The sliding time window heuristic (STWH) as applied to solve the scheduling optimization model.

3.7 Summary and Conclusion

In summary, sequential simulation techniques are implemented to model rock type and grade uncertainty from drillhole data and generate a set of simulated deposit models. These models are initially aggregated into MUs, representing the extraction scheme of block caving mines, and material flow simulation is performed by the definition of a cone of movement based on horizontal displacements and a vertical slip angle. For each geological realization a material flow simulation is done to obtain a set of possible grades for each MU that reflect this behavior.

A stochastic programming model is formulated to devise a mine plan using the whole set of simulations as an input. In order to achieve this, deviation variables are defined to account for the whole set of scenarios. The deviations are considered in ore production tonnage targets and average grade per period and different costs can be assigned to prioritize or ignore any of them. The formulation accounts for operational constraints including mining targets, grade blending, maximum draw rate, undercut development rate, mining precedence and maximum adjacent height of draw.

The optimization procedure is implemented in a MATLAB environment using IBM CPLEX as the optimization engine. In order to improve computing times, two heuristics are applied. Initially, an early start algorithm is used to define, for each MU, its earliest possible extraction period. This is done based on the precedences built as well as production targets, maximum

draw rate and maximum adjacent height of draw parameters. All the variables related to the time period before the earliest start are eliminated. Furthermore, a STWH is applied by solving the full optimization model one period at a time, with the rest of the variables being relaxed to continuous, while keeping the obtained solutions as the window moves along the mine life.

CHAPTER 4

VERIFICATION, EXPERIMENTS AND DISCUSSION OF RESULTS

This chapter presents the application of the developed model in a case study. The case study data available comprises a set of drill hole data. The outline for the geostatistical simulation workflow is presented, as well as a detailed explanation of the technical and economical parameters selected for the implementation. A deterministic LOM schedule, a stochastic model with only geological uncertainty and a stochastic model with both geological and material flow uncertainty are generated for comparison and discussion.

4.1 Introduction

The case study comprises the application of the presented workflow in a mineral deposit. The mineral deposit presented for testing purposes is a copper deposit in which two rock types, ore and waste, are considered for modeling purposes. The methods here presented for geostatistical modeling can be extended to more complex deposits.

A set of geology simulations were generated, as well as an ordinary kriging estimate in order to run the optimization model and evaluate the results. Mining operational parameters were selected to be representative of an average block caving mine, while economic parameters were chosen to be representative of current conditions as well as data released for block caving projects.

The optimization model is evaluated in both a deterministic model, in which only the kriging deposit model is used as an input, in geology uncertainty scenarios, in which the rock type and grade realizations were used as an input, and finally on the geology and material flow uncertainty scenarios. By having these results, the value of incorporating the different sources of uncertainty discussed in this research over a single estimated model can be documented and analyzed.

4.2 Geostatistical Modeling

Two domains are defined as ore and waste rock for geostatistical modeling purposes. Copper is the metal of interest and the grade is modeled as %. The dataset contains a total of 1587 samples from which 837 samples are within the ore rock type. The samples are spaced at around 50 m in easting and northing directions, and at around 10 m vertically. It is assumed that waste rock is sterile and therefore no grade modeling is performed in this domain, assigning a zero Cu grade, however the data is used for rock type modeling and simulation.

Rock type simulation is carried out initially by the use of sequential indicator simulation, assigning a value of 0 for waste rock and 1 for ore. Cu grade modeling and simulation is then performed, and the set of rock type and Cu grade simulations are merged to obtain the set of geology realizations of the deposit.

4.2.1 Rock Type Modeling

The implementation of sequential indicator simulation requires the calculation of experimental indicator variograms to quantify the spatial variability of the rock type data. The mineralized body follows a N-S trend, extending at about 500 m in this direction and at about 150 to 200 m in the E-W direction. Based on this, for indicator variograms the major direction

of continuity is defined at an azimuth of 0°, therefore the minor direction of continuity is set at an azimuth of 90°.

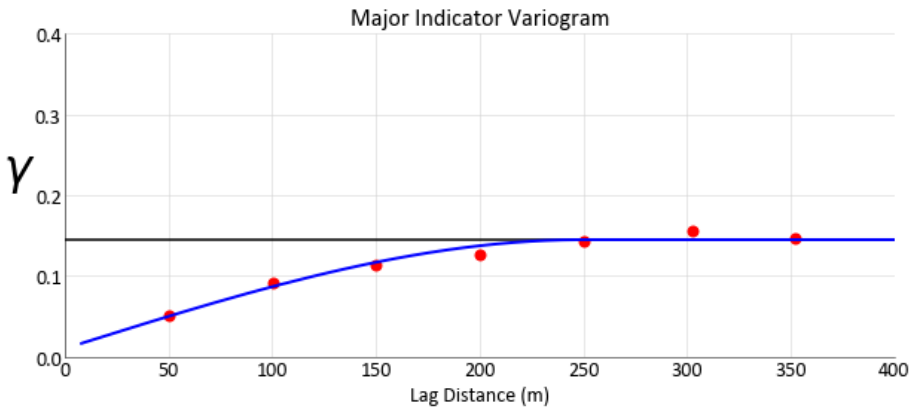
Experimental indicator variograms were calculated at these two directions, with an additional vertical variogram. The parameters for the calculation of the variograms were initially set based on the separation and extent of the samples dataset, then modified through iteration in order to provide a stable variogram for modeling purposes while avoiding oversmoothing the results. The variance of the categorical rock type dataset, assuming 0 for waste and 1 for ore, is 0.145 which is used as the sill for variogram modeling. Figure 4-1 shows the modeled experimental indicator variograms.

A single spherical structure is used for the variogram model, a nugget effect of 0.01 was used based on the vertical indicator variogram. Equation (4.1) shows the variogram model.

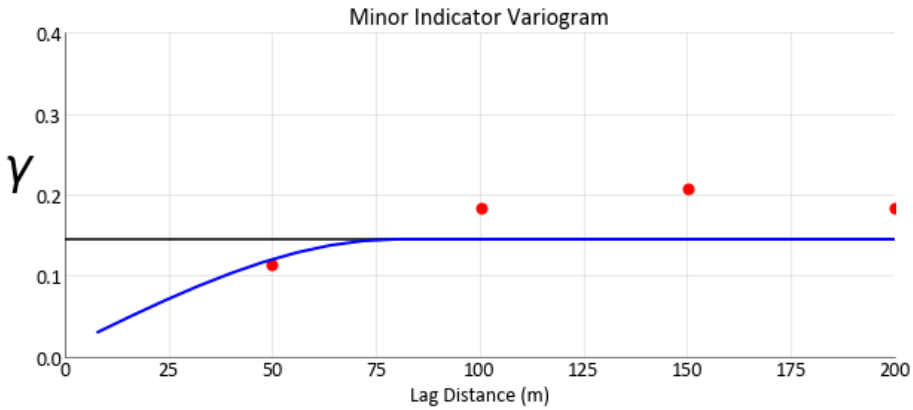
$$\gamma(h) = 0.01 + 0.135 \times Sph_{\substack{\text{major}=250m \\ \text{minor}=80m \\ \text{vertical}=300m}} \tag{3.15}$$

Indicator kriging was used to provide a kriging estimate of the rock type for a kriging deposit model. Sequential indicator simulation, through the GSLIB program *Blocksis*, was used to generate a set of 20 rock type simulations. A block size of $10m \times 10m \times 10m$ was chosen. Figure 4-2 shows a horizontal slice from the estimated rock type model and a particular realization.

(a)



(b)



(c)

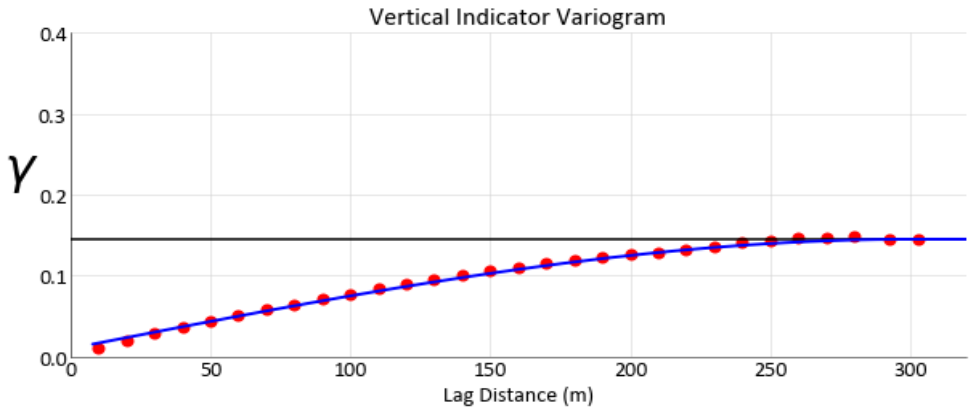


Figure 4-1. Indicator variogram modeling at major (a), minor (b) and vertical (c) directions.

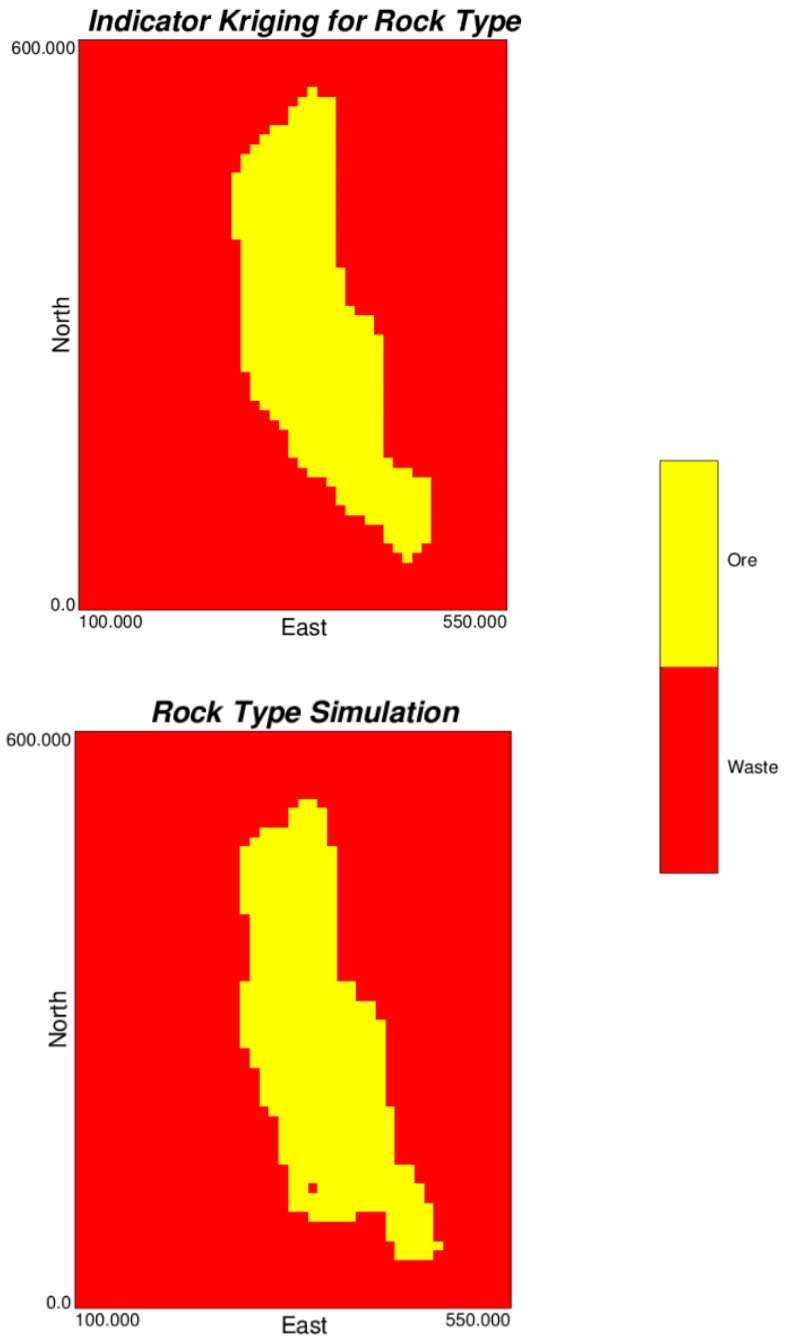


Figure 4-2. Horizontal slice of rock type estimation and a particular realization at elevation 635 m.

4.2.2 Grade Modeling

A total of 837 samples fall within the ore rock type domain, which are used for copper grade modeling. Figure 4-3 shows the histogram and cumulative distribution function for copper grades. The mean of the copper grades is 1.48% and the variance is 0.05, which is used to model the sill of the variograms.

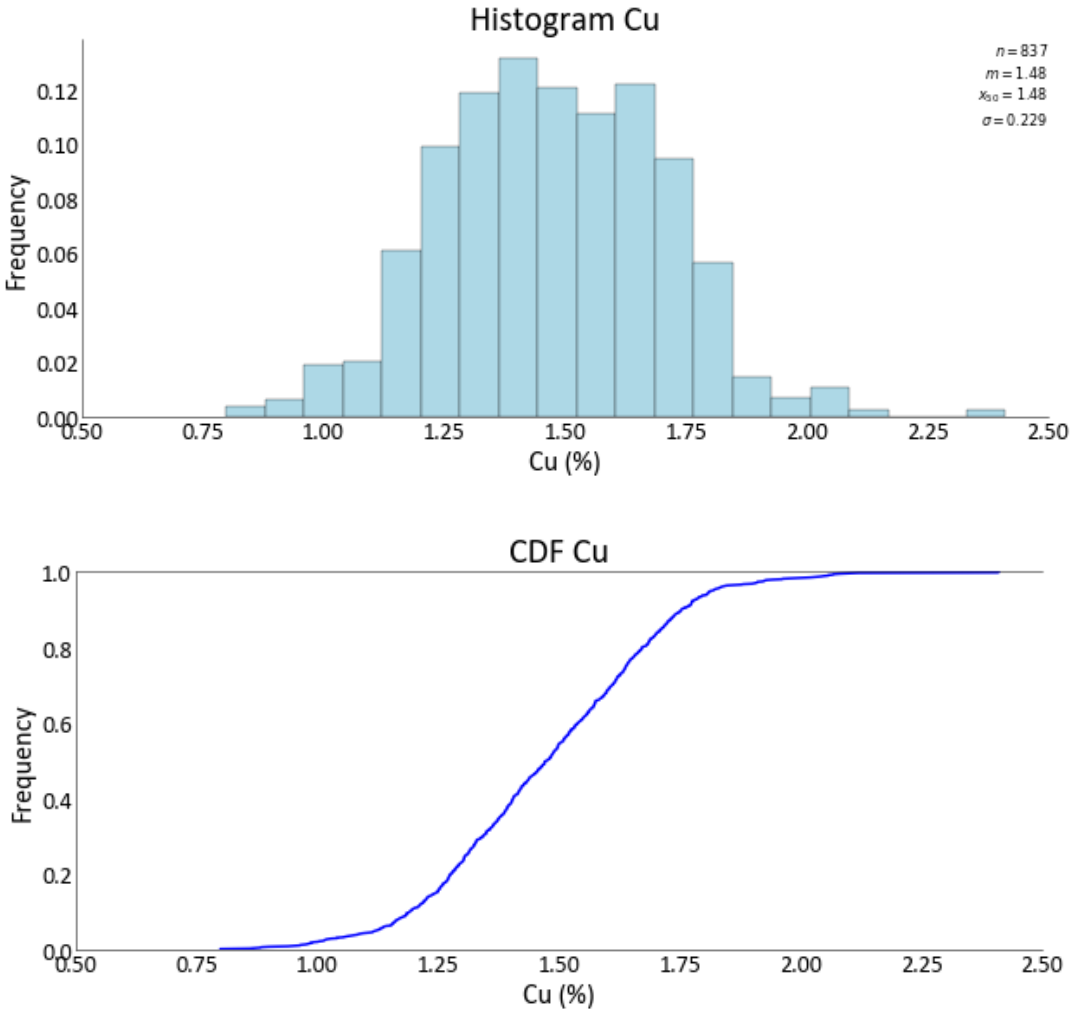
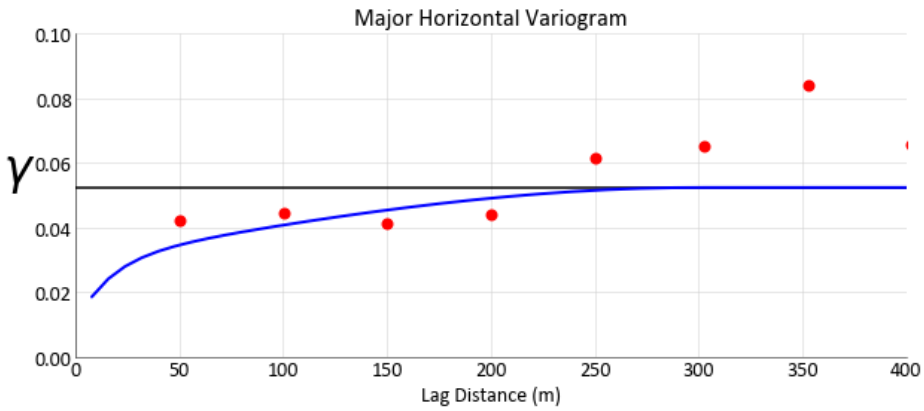


Figure 4-3. Histogram and cumulative distribution function for the copper grades.

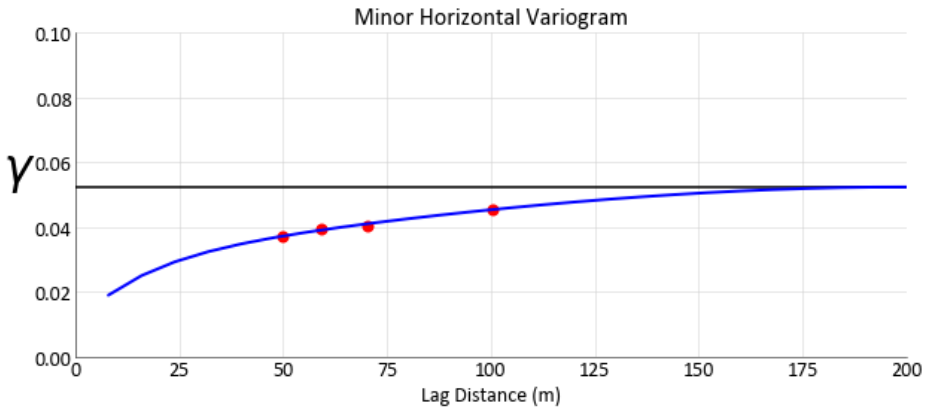
The directions of continuity used for the rock type modeling were tested for the variogram modeling of copper grades, along with different directions, and were found to provide the directions of maximum continuity as well. The major horizontal direction was set at an azimuth of 0° and the minor at an azimuth of 90°. Figure 4-4 shows the variogram models, and equation (4.2) the parametrization of the defined variogram model.

$$\gamma(h) = 0.01 + 0.02 \times Exp_{\substack{\text{major}=50m \\ \text{minor}=50m \\ \text{vertical}=30m}} + 0.0224 \times Sph_{\substack{\text{major}=300m \\ \text{minor}=200m \\ \text{vertical}=300m}} \tag{3.16}$$

(a)



(b)



(c)

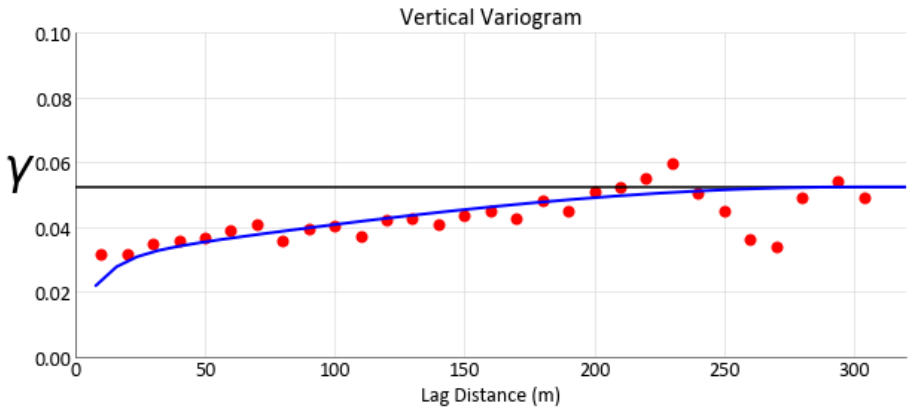


Figure 4-4. Copper variogram models for the major (a), minor (b) and vertical (c) directions.

Afterwards, a set of 20 copper grade simulations were generated using the sequential Gaussian simulation procedure. The dataset was converted into Gaussian units, following a normal score transformation for modeling purposes, and the final model back-transformed. An ordinary kriging model was also generated. The block size was defined at 10m×10m× 10m. Figure 4-5 shows a horizontal slice for a particular copper grade realization.

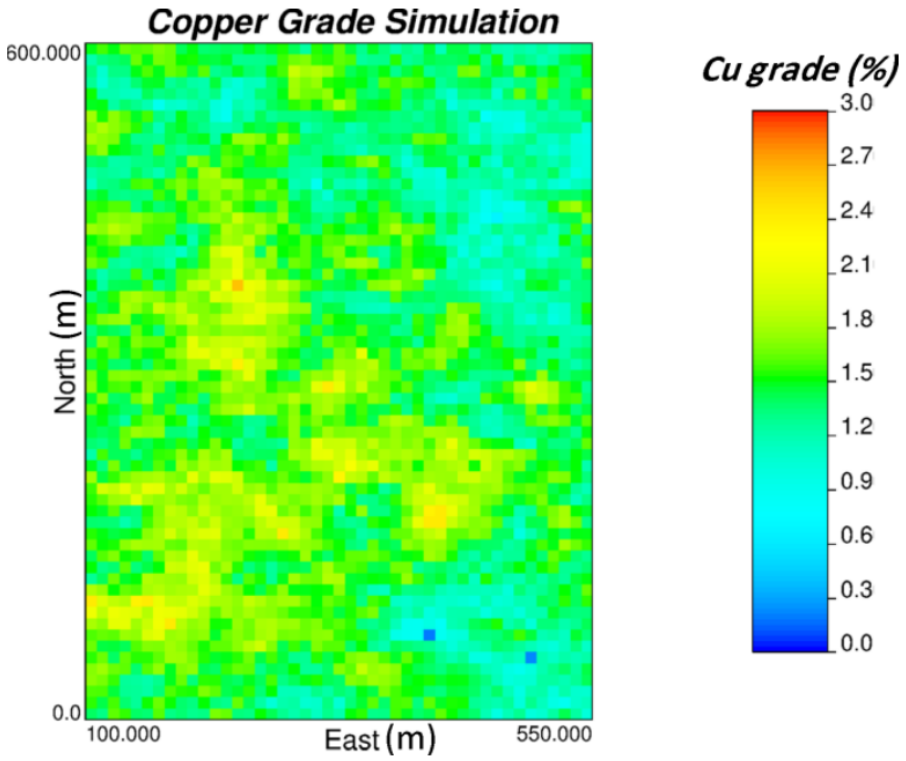


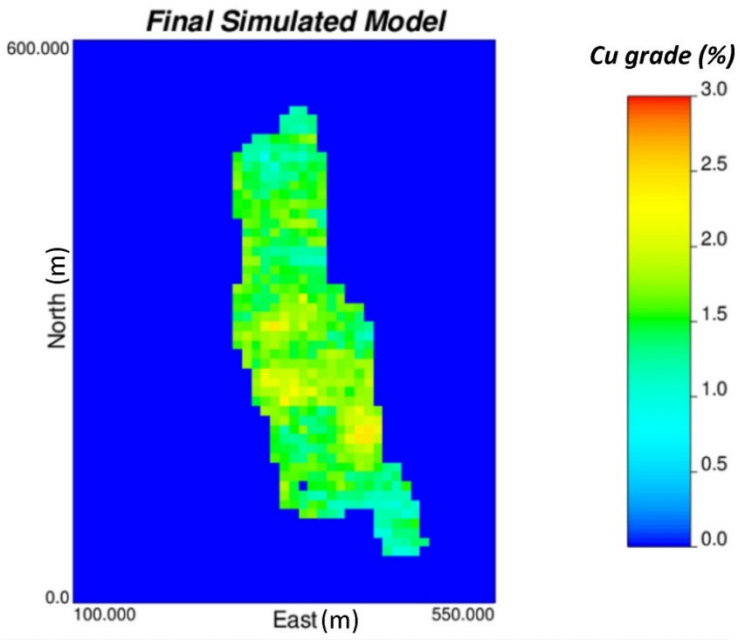
Figure 4-5. Horizontal slice of a copper grade realization at elevation 635m.

4.2.3 Geology Simulations

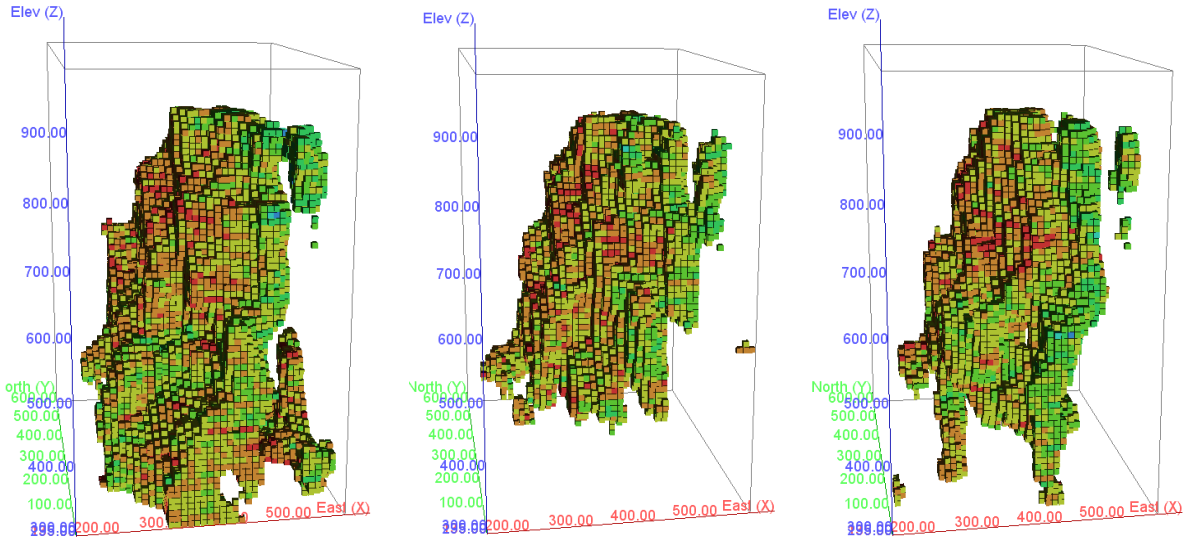
The generated rock type and copper grade simulations were to obtain a set of equi-probable simulated models of the case study deposit. This was performed using the *mergemod* program from the GSLIB library, which merges multiple geostatistical models of different domains together. As mentioned previously, two domains are considered: ore and waste rock.

Figure 4-6 shows (a) a horizontal slice of a particular realization for the merged rock type and copper grade simulation, as well as (b) 3D vies of different realizations of the mineral deposit.

To provide a mean of verification for the deposit realizations, variogram reproduction was checked for both rock type and copper grade at the three directions of continuity. Figure 4-7 shows the variogram reproduction plots. It can be observed that the spatial pattern is reproduced throughout the multiple generated realizations.



(a)



(b)

Figure 4-6. (a) Horizontal slice of a particular deposit realization at elevation 635m. (b) 3D view of different realizations of the deposit.

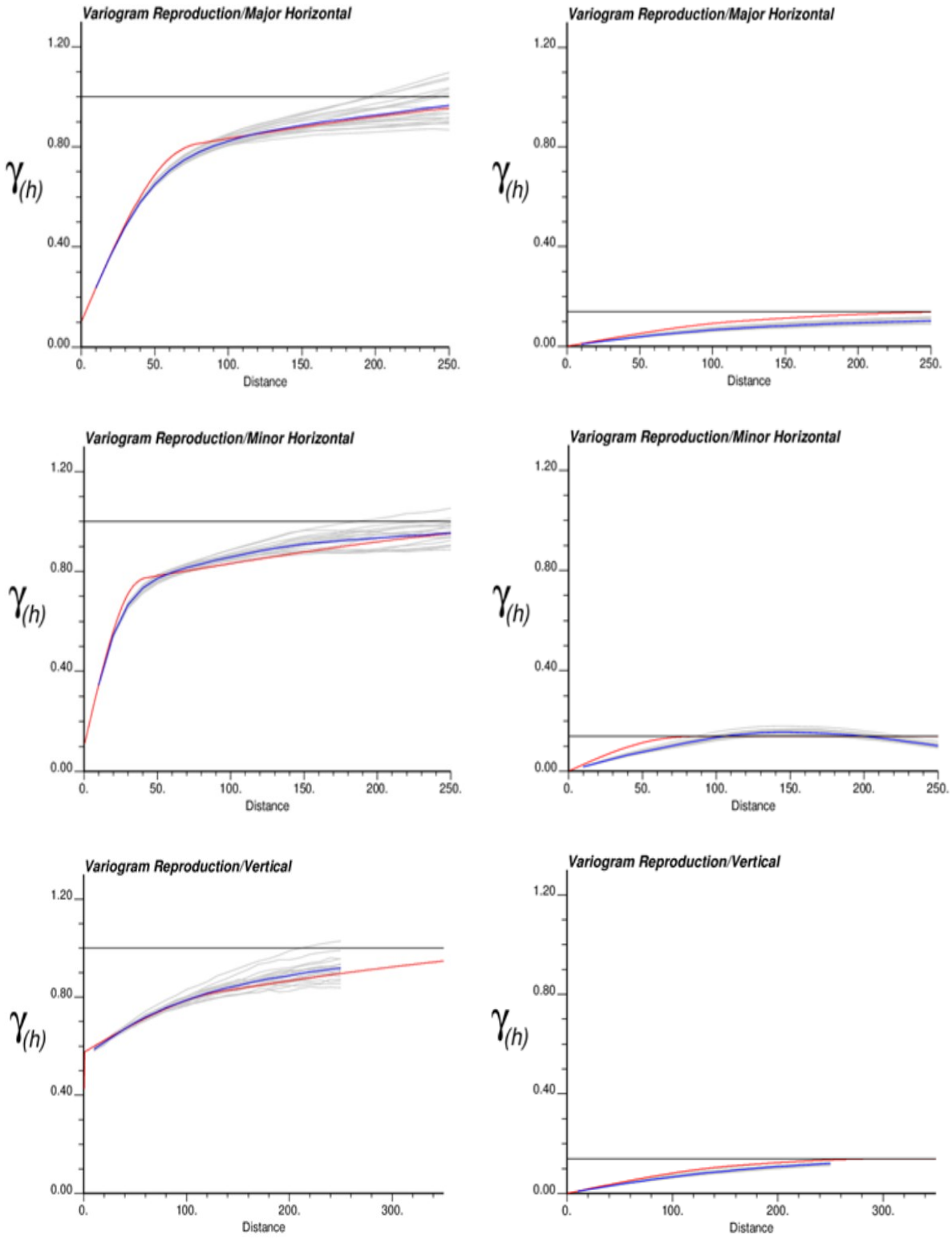


Figure 4-7. Variogram reproduction of copper grades (left) and rock types (right).

In Figure 4-8 the reproduction of the declustered copper grade histogram in the ore domain is shown, in the original units (%). Realizations are represented in gray lines and the kriging model in red. Moreover, the reproduction of the grade-tonnage curves for the deposit is checked as well (Figure 4-9). These standard checks: variogram, histogram and grade-tonnage curve reproduction allow to verify that the realizations as an ensemble are representative of the geological uncertainty throughout the deposit.

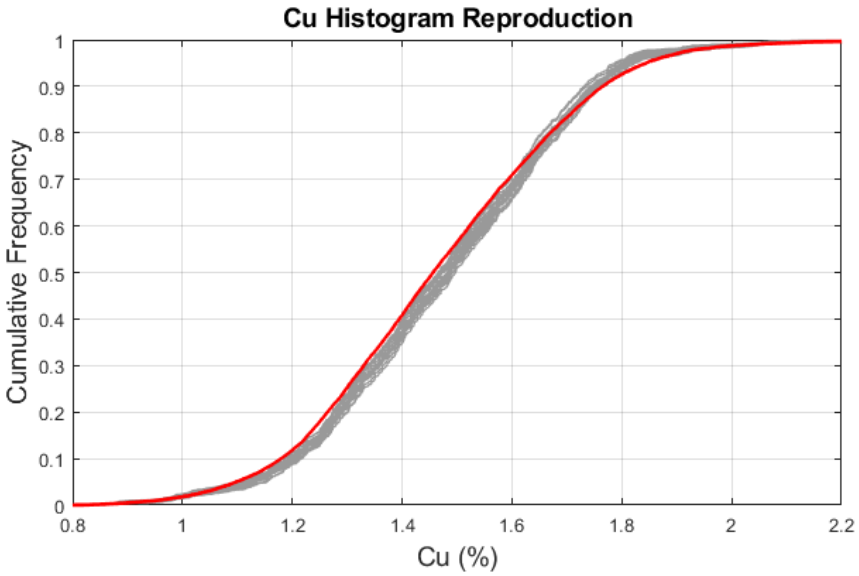


Figure 4-8. Declustered Cu histogram reproduction. Geology realizations in gray and kriging model in red.

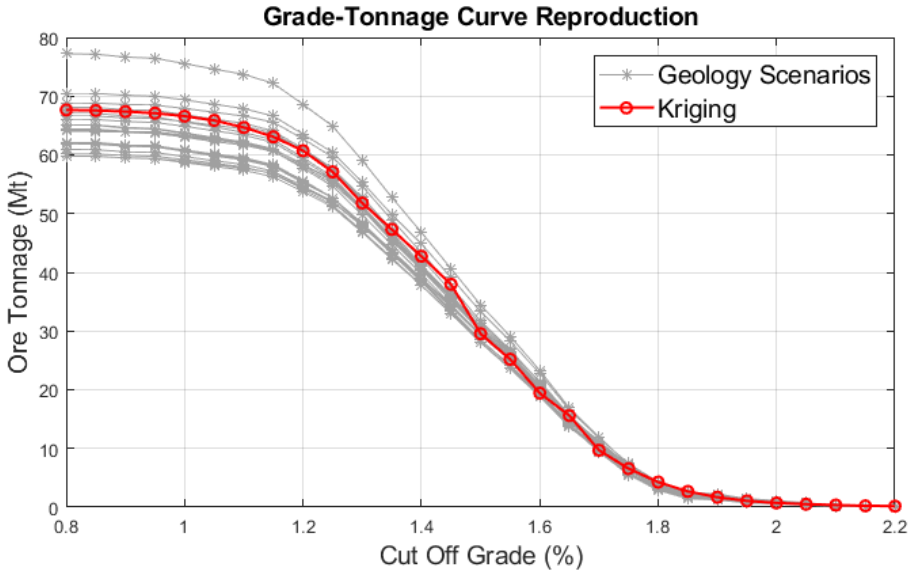


Figure 4-9. Grade-tonnage curve reproduction for the deposit. Geology realizations in gray and kriging model in red.

4.3 Parameters and Implementation Details

The technical and economic parameters used for the case study are defined to be representative of the current market conditions and to simulate a representative block caving operation based on industry reported data (Brown, 2007; D. Laubscher et al., 2017).

The optimization model proposed in this research takes the simulated block model deposits and generates a block caving LOM plan. The initial optimization model automatically aggregates the blocks into MU and PU that resemble the slices and columns commonly used to represent the rock mass in block caving mines. The second step optimization model deals with the production scheduling aspects.

The aggregation maximizes the expected metal content within the defined units amongst all the geological simulations, and varies depending on the undercut elevation that is selected. The definition of the undercut elevation is the initial step in the planning and block caving mines and as discussed previously constraints the reserves and economic potential of the project. In this research, multiple elevations are tested to select the elevation at which the highest NPV is obtained based on the stochastic schedule. To simplify this procedure, an initial run between all levels is done with the deterministic kriging model (faster computational times) to obtain a general idea of a closer range where the optimal undercut elevation lies, and run the stochastic schedules to obtain the best caving boundary and LOM plan. Table 4-1 shows a summary of the general economic parameters used in the application of the model to the case study.

Table 4-1. Economic parameters used in the case study.

Description	Parameter	Value
Selling Price ($/t)	MetalPrice^t	6,000
Mining Cost ($/t)	MineCost^t	9.3
Processing Cost ($/t)	ProcessCost^t	18.4
Recovery (%)	Rec	88.7
Discount Rate (%)	i	12
Development Cost ($/PU)	$OpCost_c^t$	150,000

This parameters are representative of average current market conditions as well as reported caving operating and development costs available. The model can easily accommodate for varying prices or costs over time (defined in periods), however, in this case study they are assumed to be fixed.

Table 4-2 summarizes the technical parameters that were used to simulate the caving operation for the case study. These parameters were defined in order to be representative of an average block caving project.

Table 4-2. Technical parameters for the implementation of the optimization framework for the case study.

Description	Parameter	Value
PU Dimensions		30m × 20m
Maximum Column Height		300 m
Minimum Column Height		60 m
No. of Periods	T	10
Minimum Draw Rate	$\underline{DR}$	35 (kton/period) (~20 m/period)
Maximum Draw Rate	$\overline{DR}$	100 (kton/period) (~60 m/period)
Undercut Development Rate	$MaxOpRate^t$	12,000 (m^2/period)
Maximum Relative Height of Draw	$MaxAbsDiff_c^{c'}$	70 (kton/period) (~ 2 MU)
Mining Starting Point		380 mE, 230 mN
Mining Direction Azimuth		335^o
Convex Front Angle		170^o
EPGAP		5%

The PU dimensions chosen were at 30m by 20m, these represent the desired drawpoint spacing. This decision is also constrained by the block size, which can be reduced using re-blocking techniques. The minimum draw rate was set at 35 kton/period, which along with the drawpoint spacing would be representative of a vertical extraction rate of about 20 m/period. This is also used to aggregate the blocks in the vertical direction into MU, once the optimum PU layout has been obtained. The maximum draw rate was set at 100 kton/period which would be equivalent to about 60 m/period.

The minimum column height refers to the minimum extraction height needed after breaking the rock mass, to sustain continuous caving conditions along the production life of the draw column and its adjacent area. This parameter depends on the geotechnical conditions of the orebody, as well as the desired size of the mine, as larger caves would require large columns for adequate caving. Current industry practice reports minimum column heights of around 50m to

100m (for the largest mines). For the case study, the minimum column height was set at 60m to match industry practice and the block height of 10m of the input resource block model. The maximum column height sets the extraction limit and vertical boundary for each draw column. Current practice determines the best economic height of draw of each column individually before any scheduling efforts. The economic height of draw is also bounded by a maximum column height as larger columns present more geotechnical risks and need to be controlled to sustain caving over their lifespan. Current operating mines report extraction height of 200m to 250m, with some of the largest projects operating or expected to operate columns over 400m height. To establish the maximum column height for the case study, the individual BHOD for each production unit was calculated by evaluating the cumulative economic value based on the technical and economic assumptions listed. Each production unit has a different BHOD; however, the maximum BHOD of all the production units was 300m. Therefore, 300m was used as the maximum column height for the building of the mining units.

The maximum relative height of draw between PUs is used to approximate the cave back slope as the extraction progresses to allow for safe extraction and efficient draw control. The definition should consider the dimensions of the PU and the approximate height of the MU based on the spacing and draw rates evaluated. For the case study, a draw difference of 70 kton/period was considered, which represents two MU and a maximum cave back angle of around 60°.

The mining direction and starting point were defined based on the methodology proposed by Khodayari & Pourrahimian (2015). On a particular undercut elevation, a production block economic value (PBEV) was calculated as the economic value obtained after the extraction of each PU up to its BHOD, defined up to its maximum economic value, along with its neighbor columns based on a search radius. This search radius was set up at 30m based on the PU dimensions. The PU with the highest PBEV was defined as the starting position, and the direction was defined to move from higher to lower value areas. The direction was set at an azimuth of 335° from which a convex front with an angle of 170° is defined.

Table 4-3 shows the ore production targets and average grade bounds per period. The ore production targets account for a ramp-up and ramp-down period, which can be easily modified based on particular projects. The production grades ranges can also be modified on a period basis however in this case study they remain fixed.

The average grade bounds can reflect technological constraints such as processing plant requirements for good operation, as well as strategic goals set up by management. On the

deterministic model, the production schedule would use this targets as a hard constraint. However, in the stochastic model the deviation variables allow for the flexibility required to manage the risk amongst all the simulated scenarios at a certain cost, whereas having fixed constraints would make the unfeasible.

Table 4-3. Ore production targets and average grade bounds per period for the case study.

	Period									
	1	2	3	4	5	6	7	8	9	10
Ore Production Target (Mton)	0.7	2	3.5	3.5	3.5	3.5	3.5	3.5	2	0.7
Minimum Avg. Grade (%)	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
Maximum Avg. Grade (%)	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

The definition of the deviation costs for both ore production targets and average grade bounds is key for the stochastic model as it impacts on the flexibility and cost assigned to the risk-based schedules. Higher costs would lead to a more reliable schedule in which all simulated models are kept within the defined targets, however potentially achieving a lower expected NPV overall. Moreover, costs are defined for positive and negative deviations from targets, as well as ore production and average grade individually. Therefore the user has the flexibility to prioritize certain aspects of the mine plan.

To have a reference to define the magnitude of the deviation costs it is of interest to look at how it affects the objective function (equation 3.5) and the general economics of the mining project. Koushavand (2014) presents a detailed procedure to estimate the optimal deviation costs for uncertainty based mine schedules.

The main principle is to consider that the cost of underproduction of ore tonnage represents the lost revenue from that shortfall. Therefore, the cost of deviation from ore production targets is related to the average revenue obtained from the extraction of a ton of ore. The deposit average grade can serve to provide an initial estimate. Higher values would lead to the optimization model to prioritize avoiding shortfalls in ore production amongst all simulations rather than obtaining the highest overall NPV. The similar principle applies for the overproduction of ore tonnages. Overproduction cost can also be linked to a differential cost that can be potentially incurred in order to manage the excess production.

The estimation of the cost for positive and negative deviations of average grade sent to mill follows a similar principle. However, as presented in equation (3.7) and equation (3.8), the grade deviation variables balance the metal content sent to the mill amongst all realization based on the defined targets. The cost of shortfall or positive deviation of metal is related to the revenue obtained from the selling of one ton of it. The selling cost of the metal of interest serves as a reference for an initial estimation for the deviation cost of average grades sent to the mill.

The geological discount rate is the final parameter related to the uncertainty based production scheduling model. This rate serves as a discount factor for the cost associated with the deviations due to the variation between the simulated models. High values would prioritize minimizing deviations on the initial periods at the cost of potentially lower NPV, while lower values would lead to higher deviations allowed in the initial periods. Table 4-4 shows the deviation costs assigned for the case study.

Table 4-4. Deviation costs and geology discount rate assigned for the case study

Cost for positive deviation (overproduction) of ore tonnage ($/t)	p^+	60
Cost for negative deviation (shortage) of ore tonnage ($/t)	p^-	120
Cost for positive deviation (over upper bound) of grade quality range ($/t)	q^+	6000
Cost for negative deviation (under lower bound) of grade quality range ($/t)	q^-	12000
Geological discount rate (%)	g	15

The cost for positive and negative ore tonnage deviations is based on the average revenue produced by the extraction and processing of a ton of ore, based on the deposit average grade. The cost for overproduction is set at 60 $/t, which matches the mentioned revenue, while the cost for underproduction is set at double to prioritize minimizing potential shortfalls in ore production targets.

The cost for positive and negative deviations in average grades sent to mill are based on the set selling price of copper. The cost of positive deviation is set at 6000 $/t of copper while the cost of negative deviation is set at double to prioritize it.

Finally, the geological discount rate is set at 15%, a slight increase over the economic discount factor. It is important to analyze the sensitivity of these parameters and their impact on the project NPV and ore production and grade risk profiles to get a better understanding.

In relation to the material flow scenarios, the horizontal displacement was set at 35m and the vertical slip angle at 60º. These parameters were defined to be in line with Alvial (1992) measurements and were applied to the whole footprint. The point of entry dilution was set at 40m, which defines the height at which the cone, and therefore mixing scenarios, start being evaluated for each PU.

4.4 Deterministic LOM Schedule with Kriging Estimates

As mentioned previously, the determination of a LOM plan and economic KPIs for a block caving mine depends on the selection of the undercut placement as the extraction is performed on the overlying broken rock mass. Multiple undercuts can be considered for multiple lifts, or production levels, however, the model developed initially considers only one production level.

A deterministic version of the model is used to evaluate the NPV and ore tonnage of the potential block cave mine. The deterministic version follows the same constraints and objective function described above using only the single kriging estimated deposit. This version of the model is run over multiple undercut elevations (placed at the based of each horizontal slice of the block model) to determine the best in terms of NPV for the deterministic case. Laubscher vertical mixing algorithm is used to represent dilution, which has become the standard procedure for the evaluation of mine plans at this level of detail. Based on the results obtained a smaller range of elevations is selected to run the stochastic versions of the model, as the computing times required for the multiple scenarios optimization would be significantly higher. Figure 4-10 shows the results for the deterministic case.

The undercut elevation that yields the highest NPV is 635m, with elevations between 635 to 605 yielding similar values. Each point in this graph corresponds to one LOM and production schedule. For demonstration purposes the highest NPV obtained at 635m was selected, however it would be suggested to inspect in more detail the LOM plans at elevations between 635 to 605 as the values are very close. The range between 645 and 595m was selected to evaluate the geological and flow uncertainty schedules. The NPV and ore tonnages obtained at this range are shown in Table 4-5.

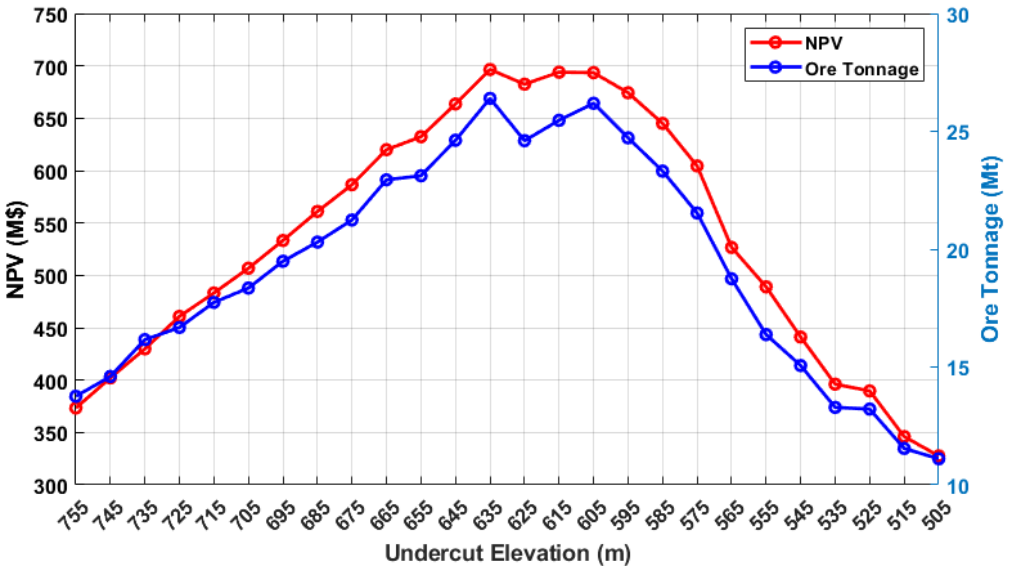


Figure 4-10. NPV and ore tonnage for multiple undercut elevations at the deterministic Kriging deposit model.

Table 4-5. NPV and ore tonnage at the selected undercut elevations.

	645m	635m	625m	615m	605m	595m
NPV (M$)	663.632	696.492	682.484	693.939	693.501	674.318
Tonnage (Mt)	24.621	26.396	24.602	25.466	26.190	24.724

The caving envelope at 635m undercut elevation is shown in Figure 4-11, within the kriging estimated geological model. This caving envelope, footprint limits and height of draw, maximizes the NPV from the production schedule based on the parameters and constraints mentioned. The undercut opening sequence is presented in Figure 4-12.

The undercut opening sequence refers to the periods at which the columns are opened for production at the undercut level, and defines the mining footprint for the caving project. The total footprint area for the deterministic model at 635m undercut is 92,400 m^2.

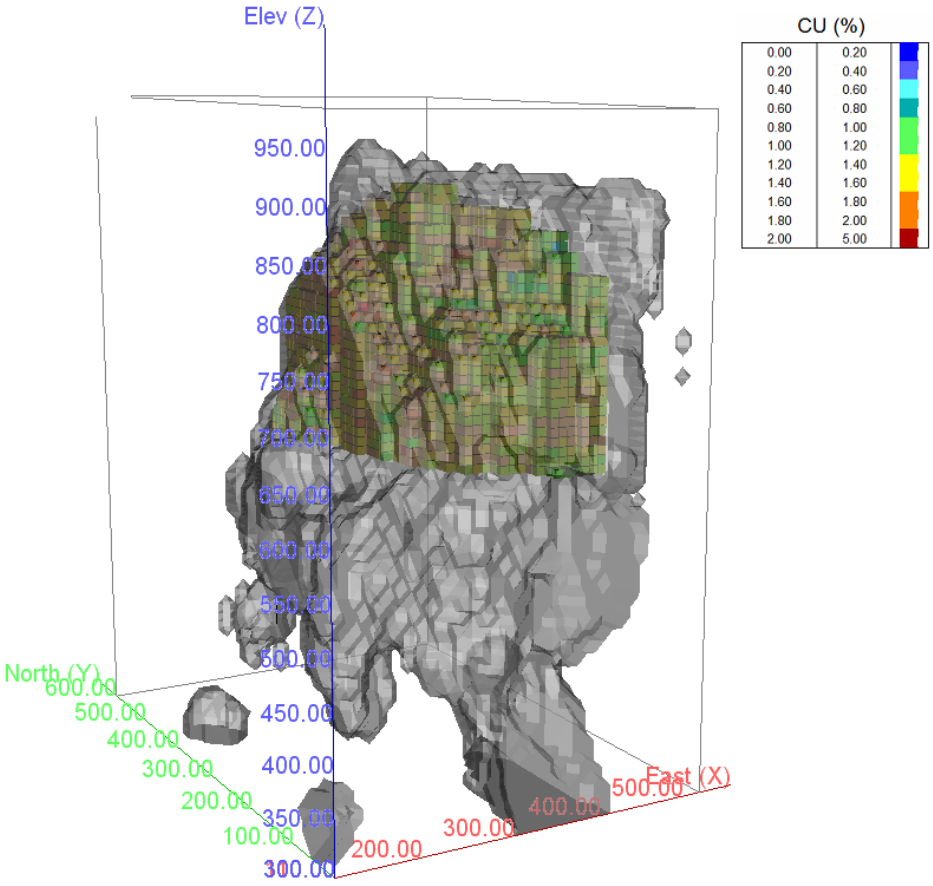


Figure 4-11. Block caving reserves envelope for the deterministic model at undercut elevation 635m.

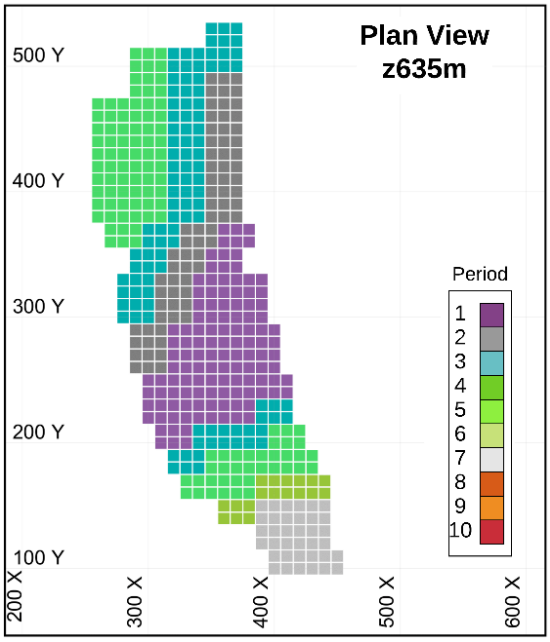


Figure 4-12. Undercut sequence for the deterministic model.

Figure 4-13 shows the production schedule for the deterministic model. The ore tonnage production, at a total 26.4 Mt meets the production targets established including the ramp-up and ramp-down periods. The average grade is shown along with the defined bounds.

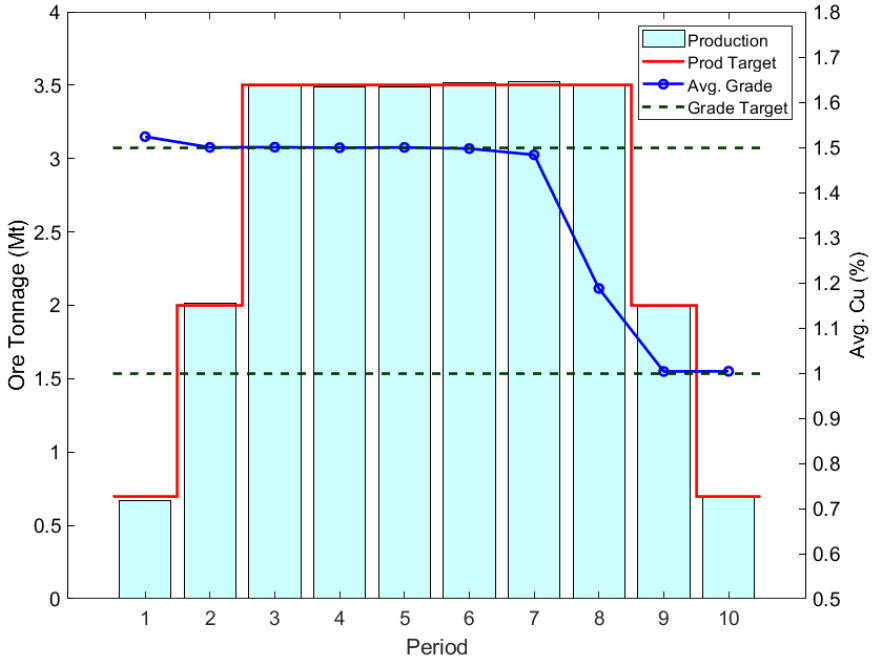


Figure 4-13. Production schedule with targets for the deterministic model.

For the initial seven periods the solution yields an average production grade at the upper bound target, with the first period slightly higher due to the resulting profit paying for the over production cost. This leads to very low grades, at the lower acceptable bound, for the later periods. This type of schedule could lead to operational problems due to the uncertainty in the grade estimates. The extraction heights of the caving envelope on a period basis are shown in Figure 4-14. This represents the caving profile throughout the LOM of the project. This profiles maximizes the NPV, and keeps the cave back slope under the defined parameters. It serves as an initial estimate as the caving profile and propagation requires detailed numerical modeling to evaluate its feasibility.

The cumulative NPV and discounted cash flows for the LOM plan are showed in Figure 4-15 and Figure 4-16 respectively. It can be seen that the highest discounted cash flows are generated at the initial periods, with the largest obtained in period three. This is due to the caving operation being initiated in the higher grade areas and moving towards the lower grade and boundaries in the later periods. This is the ideal case, as practical applications would require geomechanical evaluation to define the mining direction that balances the economic value of the project with the caveability of the rock mass.

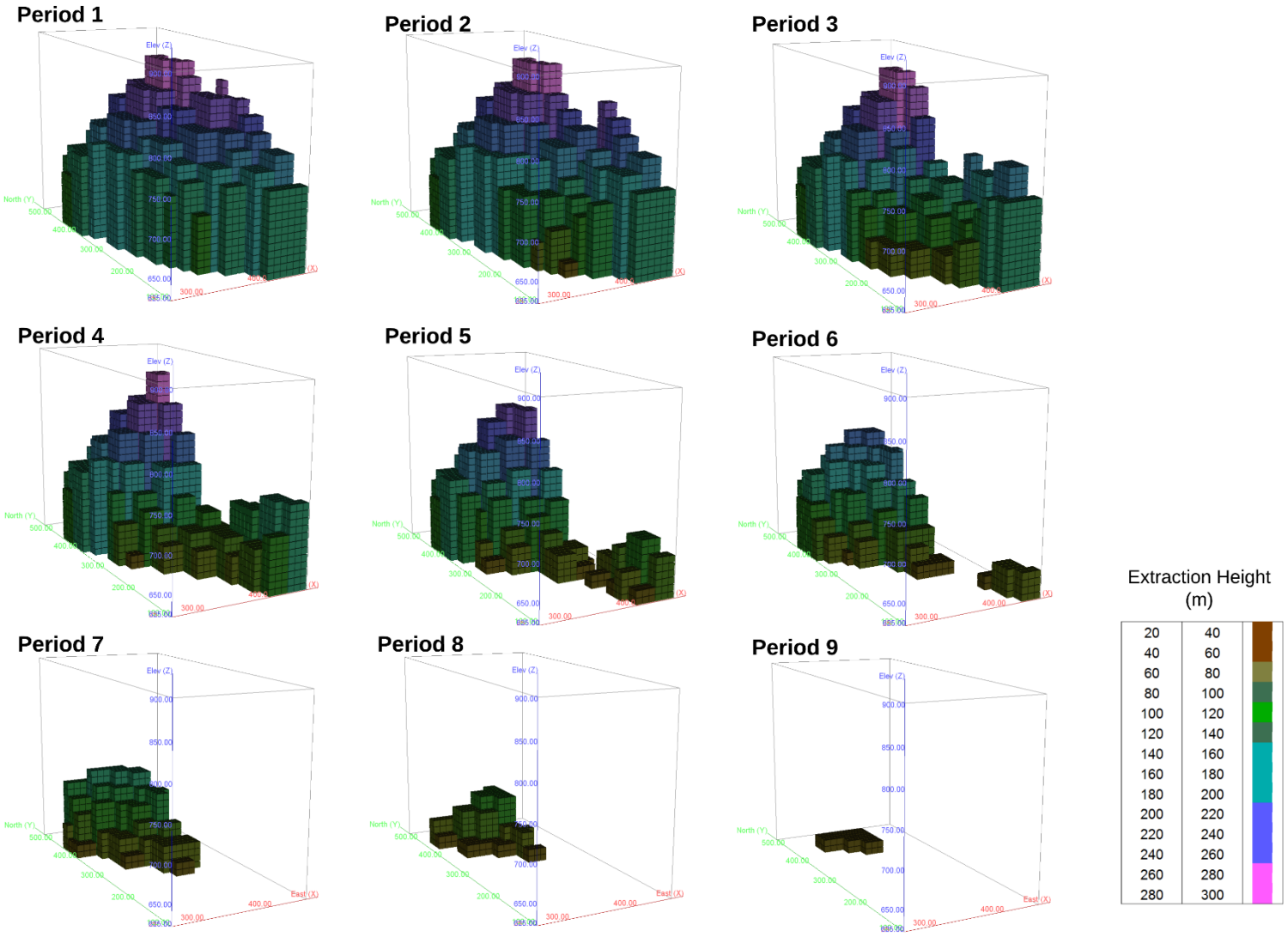


Figure 4-14. Extraction heights for the caving project by period for the deterministic model.

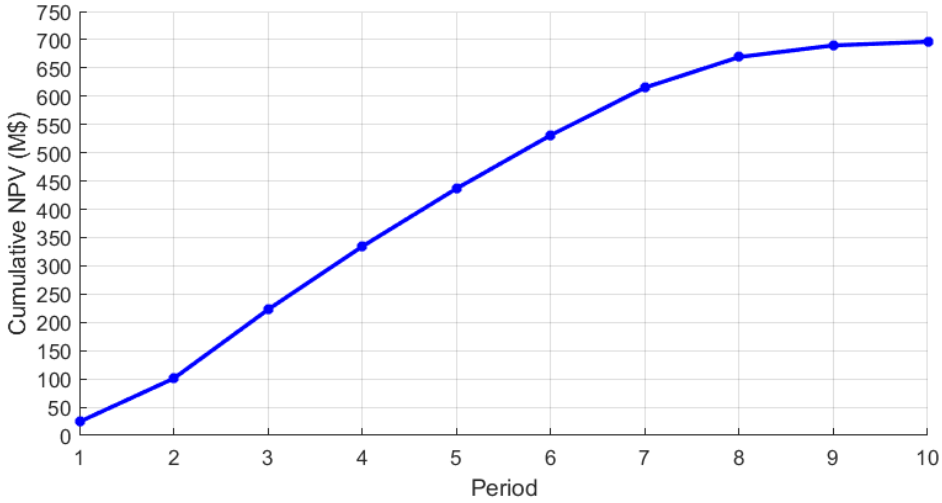


Figure 4-15. Cumulative NPV over the LOM for the deterministic model.

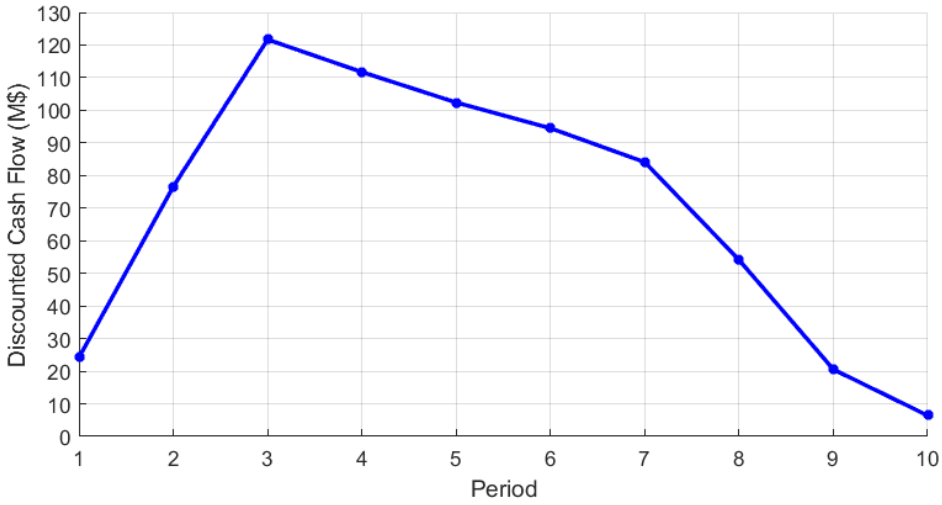


Figure 4-16. Cash Flows over the LOM for the deterministic model.

The indicators presented above consider the LOM plan generated using a single estimated model. On the following sections the impact of the geological uncertainty and the geological and material flow uncertainty in this LOM plan are presented.

4.4.1 Geological Risk Profile

The obtained sequence for the kriging estimated model is evaluated over the 20 generated geological simulations to evaluate the response of the mine plan on the economics of the project and the production schedule.

Figure 4-17 and Figure 4-18 show the uncertainty profile in the cumulative NPV and cash flows over the geological scenarios. The gray lines represent the response of each geological

realization to the deterministic mine sequence, summarized in the expected value, 10th and 90th percentile.

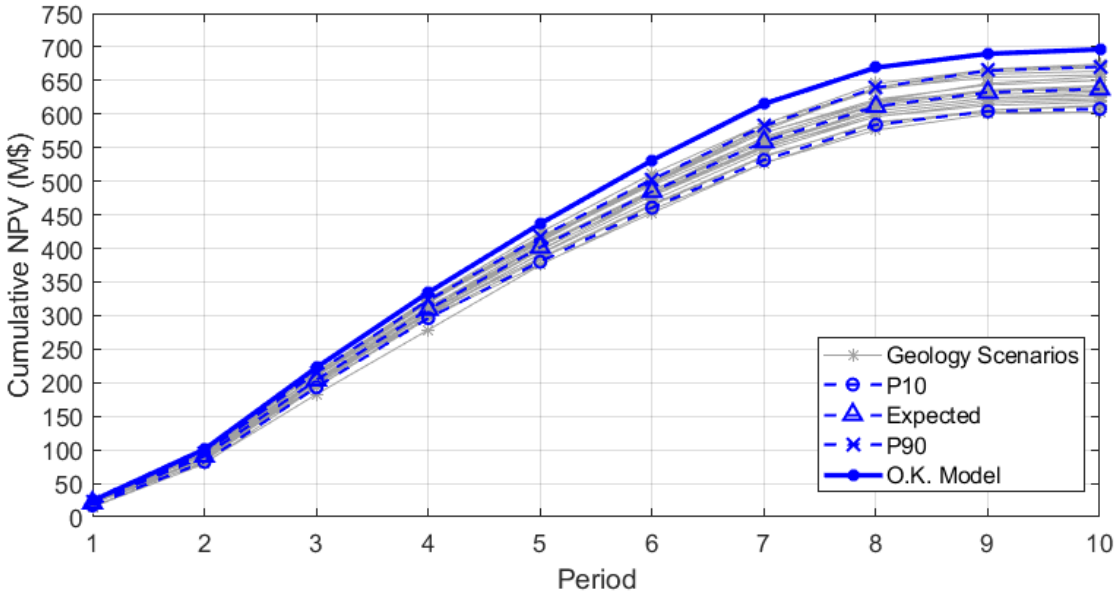


Figure 4-17. NPV uncertainty profile for the deterministic mine plan over the geology scenarios.

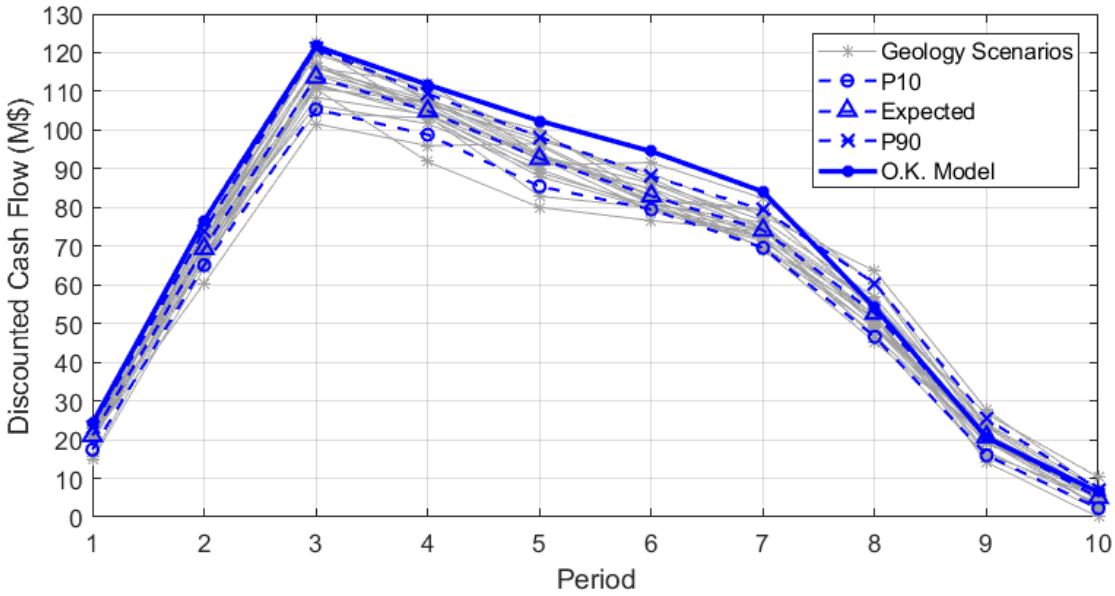


Figure 4-18. Cash flow uncertainty profile for the deterministic mine plan over the geology scenarios.

While the NPV obtained used the deterministic kriging model was 696.492 M$, the expected NPV based on the response of the 20 geology realizations of the mineral deposit was 638.326 M$. The variability and uncertainty due to the grade and rock type estimation process resulted in a 8.35% lower expected NPV, or 58.166 M$. Moreover, the 90th percentile of the final NPV

distribution due to the geological scenarios was 676.623 M$, still lower than the kriging estimated NPV.

The cash flow uncertainty profile showed a similar behavior, especially at the initial periods where the schedule extracts the higher grade areas of the deposit the kriging mine plan estimate is higher than the 90th percentile of the geological scenarios response. At the later periods the kriging cash flow estimates matches the expected response from the geological scenarios. The deterministic model economic forecasts have a very low probability of actually being achievable based on the geology uncertainty.

Geological uncertainty has also impact on the operational aspects of the schedule, especially in the ore production and average production grade. Figure 4-19 shows the average production grade uncertainty profile for the deterministic kriging mine plan over the geology scenarios. No significant differences were found in the ore production tonnage between the geology scenarios and the kriging estimated model.

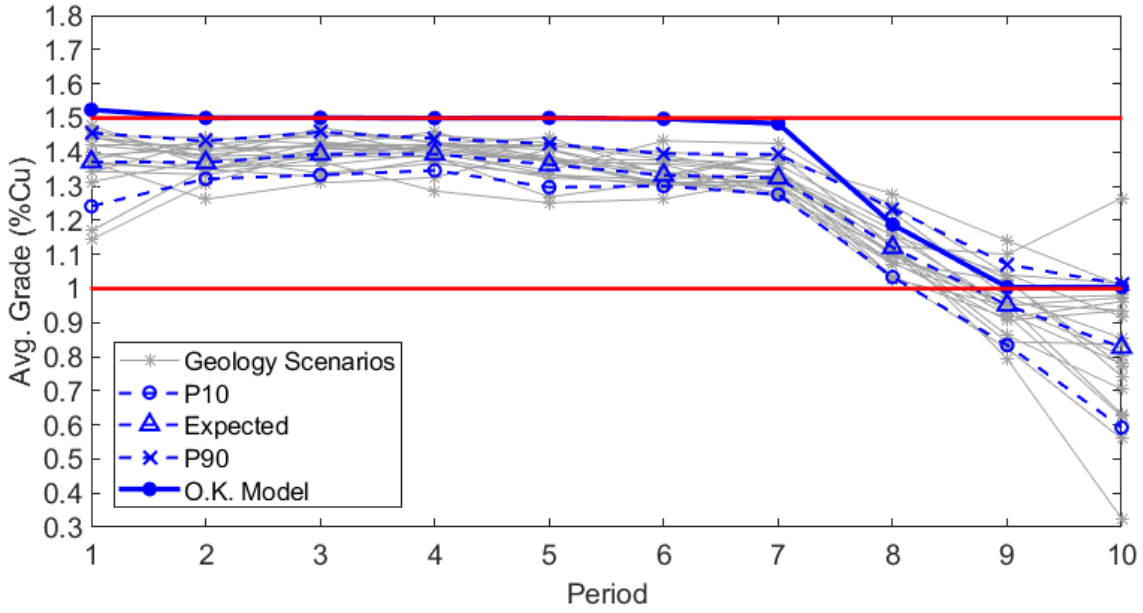


Figure 4-19. Average production grade uncertainty profile for the deterministic plan over the geology scenarios.

The expected average production grade is lower than the forecasted by the deterministic mine plan. However for most of the LOM the response of the geology scenarios to the deterministic mine plan is within the defined operational limits, with problems occurring at period 9 and 10, where the expected average grade is found below the lower operational limit.

4.4.2 Geological and Material Flow Risk Profile

The obtained sequence for the kriging estimated model is evaluated over the 20 genereated geological and flow simulations to evaluate the response of the deterministic mine plan on the economics of the project and the production schedule.

Figure 4-20 and Figure 4-21 show the uncertainty profile in the cumulative NPV and cash flows over the geological and mixing scenarios. The gray lines represent the response of each geological and flow realization to the deterministic mine sequence, summarized in the expected value, 10th and 90th percentile.

While the NPV obtained used the deterministic kriging model was 696.492 M$, the expected NPV based on the response of the 20 geology and flow realizations of the mineral deposit was 557.988 M$. The uncertainty due to the geological estimation and material mixing processes resulted in a 19.88% lower expected NPV, or 138.504 M$. This was mainly a result of the dilution due to horizontal and vertical mixing especially at the boundaries of the footprint, as waste was introduced based on the cone of movement concept presented for the generation of the mixing scenarios. Explicit incorporation of the geological and flow scenarios in the optimization process is highly desired in order to increase the potential value of the project.

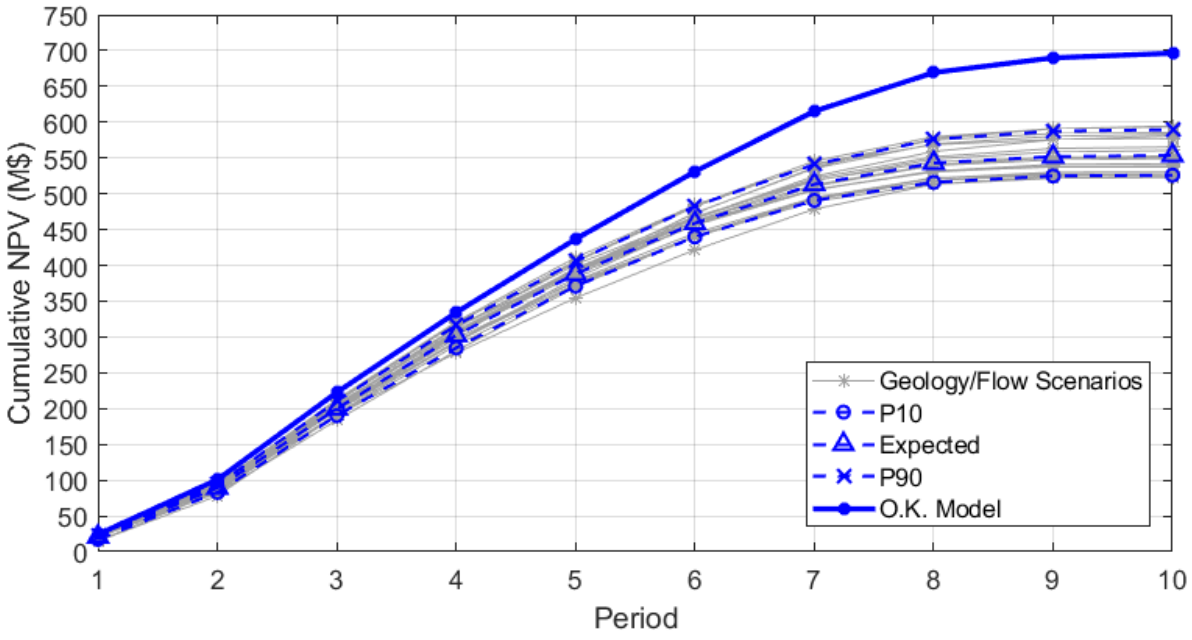


Figure 4-20. Cumulative NPV uncertainty profile for the deterministic mine plan over the geology and material flow scenarios.

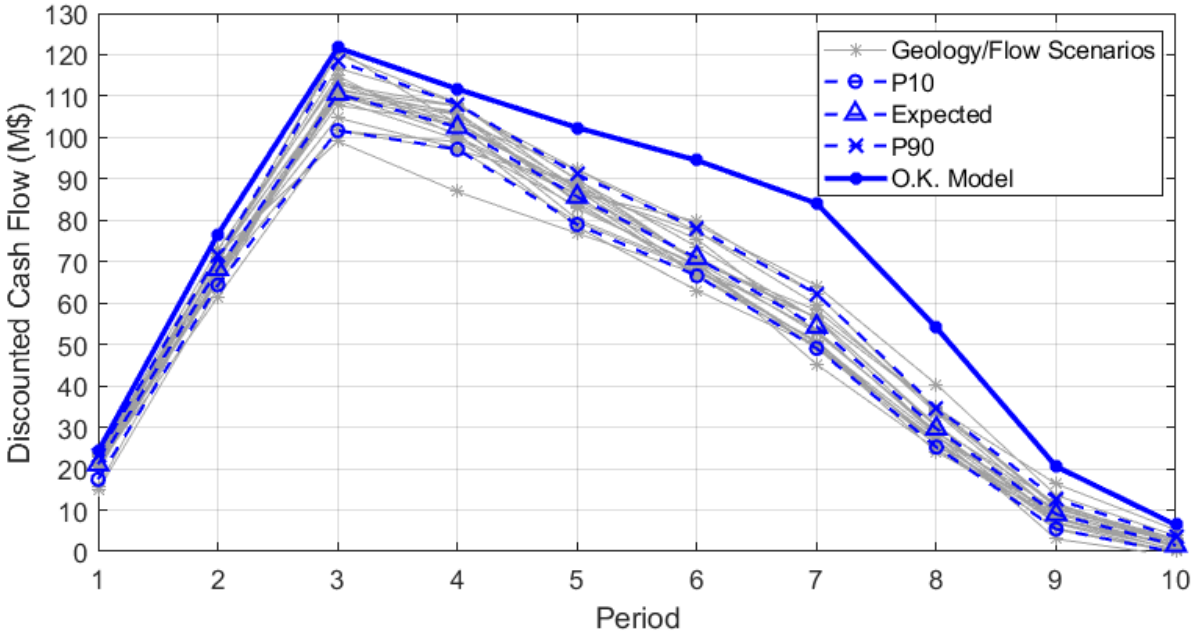


Figure 4-21. Cash flow uncertainty profile for the deterministic mine plan over the geology and material flow scenarios.

The impact of the geological and flow uncertainty scenarios on the average production grade based on the deterministic mine plan is presented in Figure 4-22. The response of the geology/flow scenarios is on overall lower than the forecasted by the deterministic model, with significant problems most probably ocurring from period 7 onwards as the expected average production grade falls below the lower bound required.

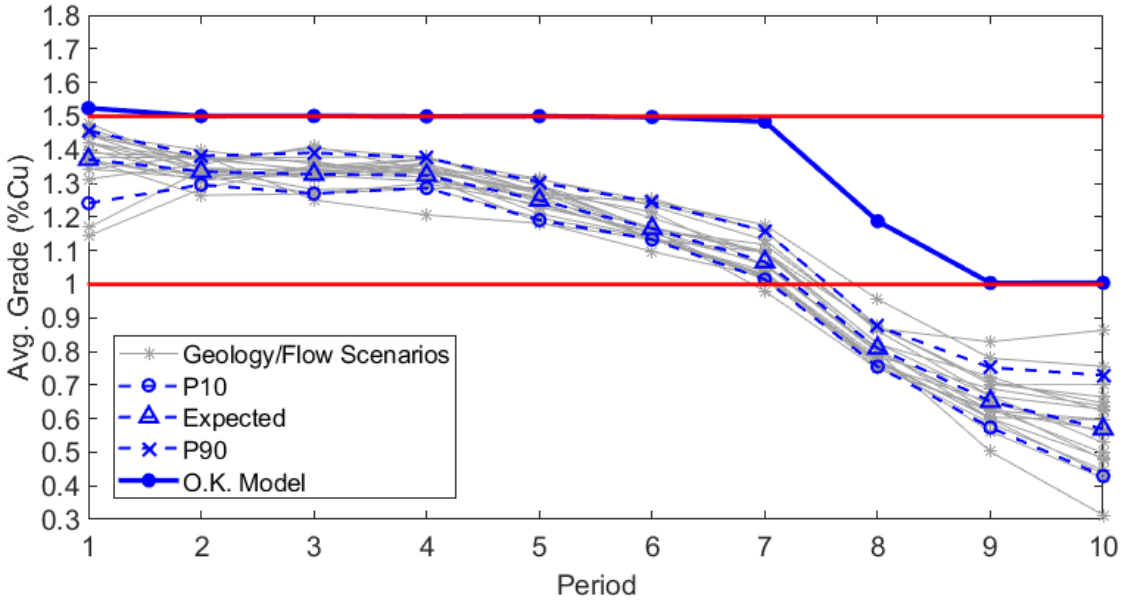


Figure 4-22. Average production grade uncertainty profile for the deterministic mine plan over the geological and flow scenarios.

4.5 Stochastic LOM Schedule with Geological Uncertainty

The stochastic LOM schedule uses the full stochastic formulation described, along with the 20 geological (grade and rock type) simulations. The expected economic value of each MU was used to drive the objective function while minimizing the deviations in ore tonnage and average production grades from the desired bounds.

The model was used in the range of undercut elevations between 595 and 645m that were defined as the highest valued caves in the deterministic case, since the cave with the highest NPV can be expected to be within this range, and running the stochastic formulation on the full resource model would be computationally expensive.

Table 4-6 shows the NPV and ore tonnage for this range of undercut elevations.

Table 4-6. NPV and ore tonnage for the stochastic optimization with geological uncertainty at different undercut elevations.

	645m	635m	625m	615m	605m	595m
NPV (M$)	652.312	667.547	661.894	664.431	672.726	642.876
Tonnage (Mt)	24.761	26.216	26.445	26.447	26.453	24.583

For the stochastic optimization with geological uncertainty, the undercut elevation that yielded the highest NPV was 605m at 672.726 M$. The most profitable cave when introducing geological uncertainty in the optimization process was located 30 m deeper than the one found in the deterministic case. The stochastic NPV was just about 3.41% lower when compared to the deterministic case, while the difference in ore tonnage was negligible. This highlights the value of integrating multiple realizations to obtain a LOM plan that considers the variability and uncertainty at the current level of knowledge of the mineral deposit.

Figure 4-23 shows the caving envelope for the stochastic case with geological uncertainty at undercut elevation 605m. For displaying and comparison purposes, the grades shown are the expected copper grade from all the geology realizations, and the deposit model is the deterministic kriged model.

The undercut opening sequence is presented in Figure 4-24. The sequence varies from the deterministic case to accommodate the variability in grade throughout the multiple realizations. The total area for the stochastic case is 106,800 m^2, a larger footprint.

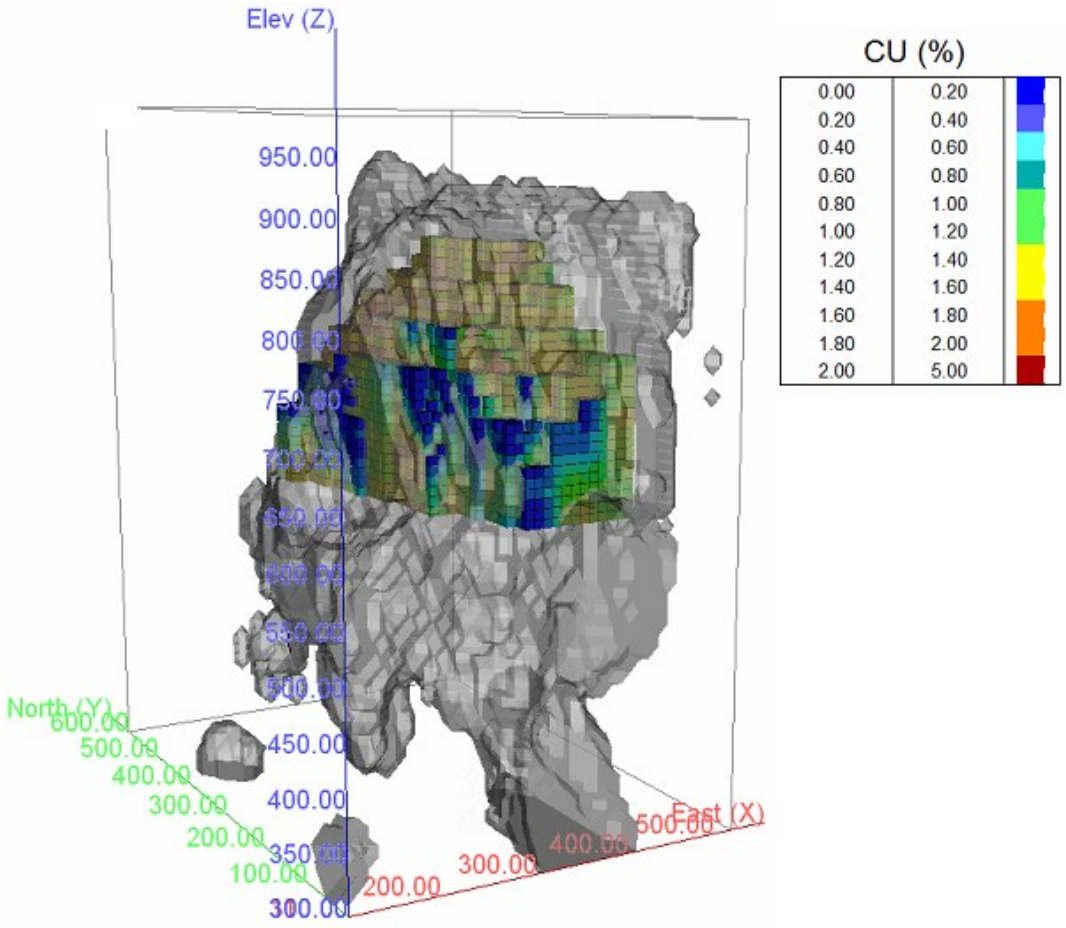


Figure 4-23. Block caving reserves envelope for the stochastic case with geological uncertainty at undercut elevation 605m.

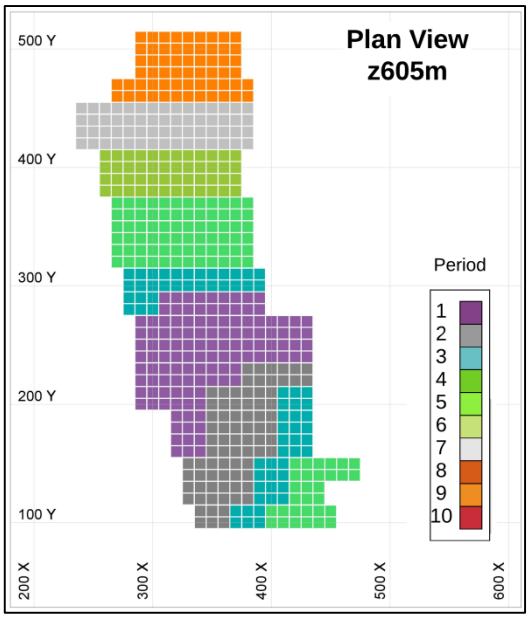


Figure 4-24. Undercut sequence for the stochastic case with geology uncertainty.

Figure 4-25 shows the production schedule for the stochastic case with geological uncertainty. The ore tonnage production, at a total 26.4 Mt meets the production targets established including the ramp-up and ramp-down periods. The average grade is shown along with the defined bounds. The average production grade differs from the deterministic case in that it follows a more balanced schedule.

The extraction heights of the caving envelope on a period basis are shown in Figure 4-26. This represents the caving profile throughout the LOM of the project. The caving sequence maximizes the NPV while minimizing the deviations incurred due to the geological uncertainty characterized throughout the multiple realizations.

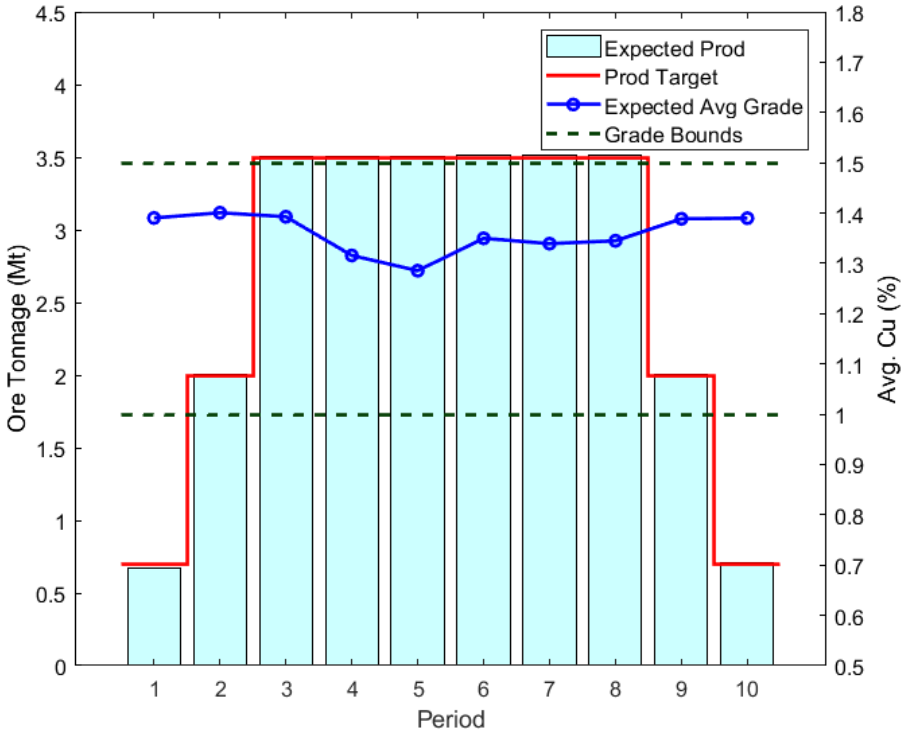


Figure 4-25. Production schedule for the stochastic case with geological uncertainty.

The cumulative NPV and discounted cash flows for the LOM plan are shown in Figure 4-27 and Figure 4-28 respectively. As mentioned above, the expected NPV for the project under geological uncertainty was found at 672.726 M$ with the 10th percentile of its distribution at 623.632 M$ and the 90th percentile at 712.376 M$. The overall pattern of the economics for the project were similar to the one observed at the deterministic case, with the adjustment in the sequence to accommodate the variability in ore tonnage and average production grade as the main reason for the lower achieved economic value.

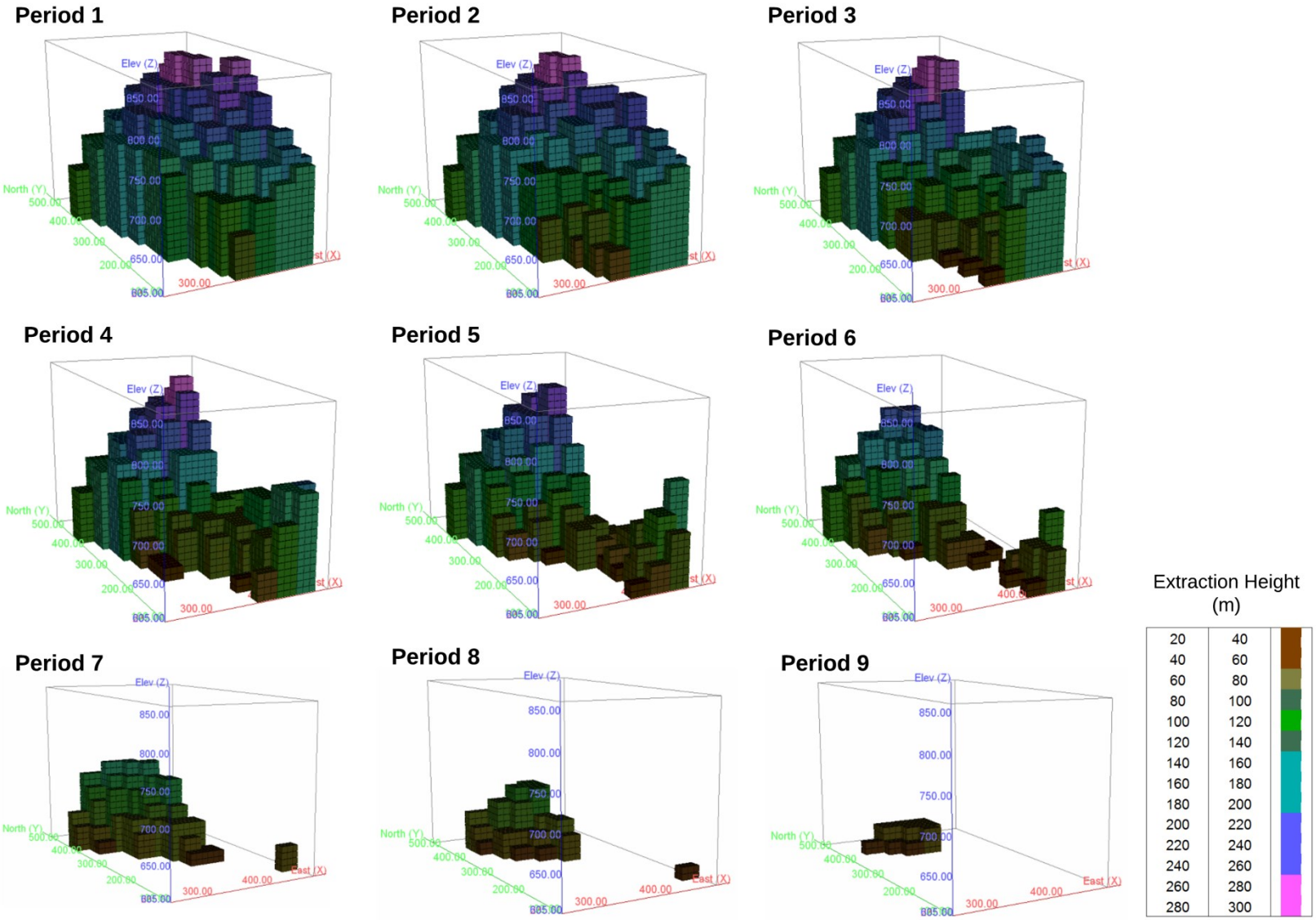


Figure 4-26. Extraction heights for the caving project by period for the stochastic case with geological uncertainty.

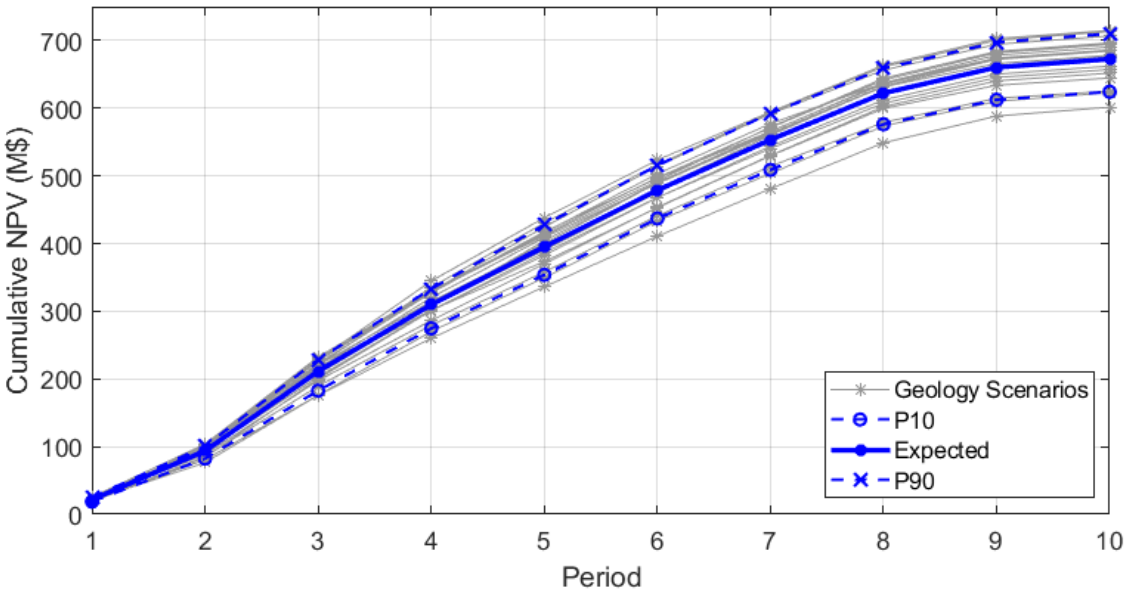


Figure 4-27. Cumulative NPV over the LOM for the stochastic case with geological uncertainty.

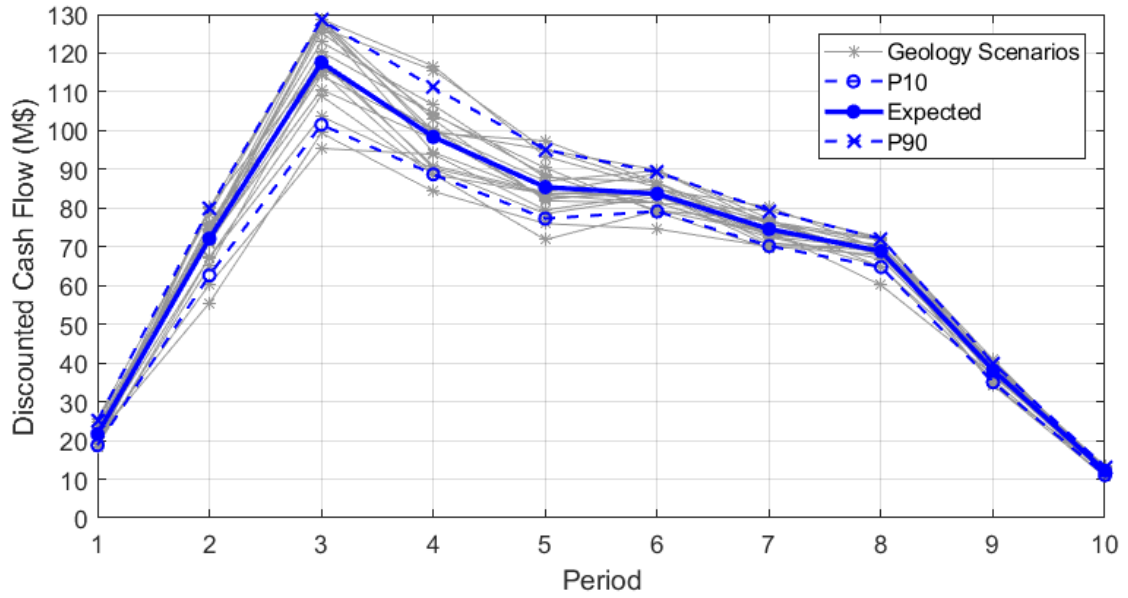


Figure 4-28. Cash Flows over the LOM for the stochastic case with geological uncertainty.

The average production grade for the multiple geological scenarios is presented in Figure 4-29. All realizations are kept within the boundaries throughout the whole LOM as they are explicitly included in the optimization process.

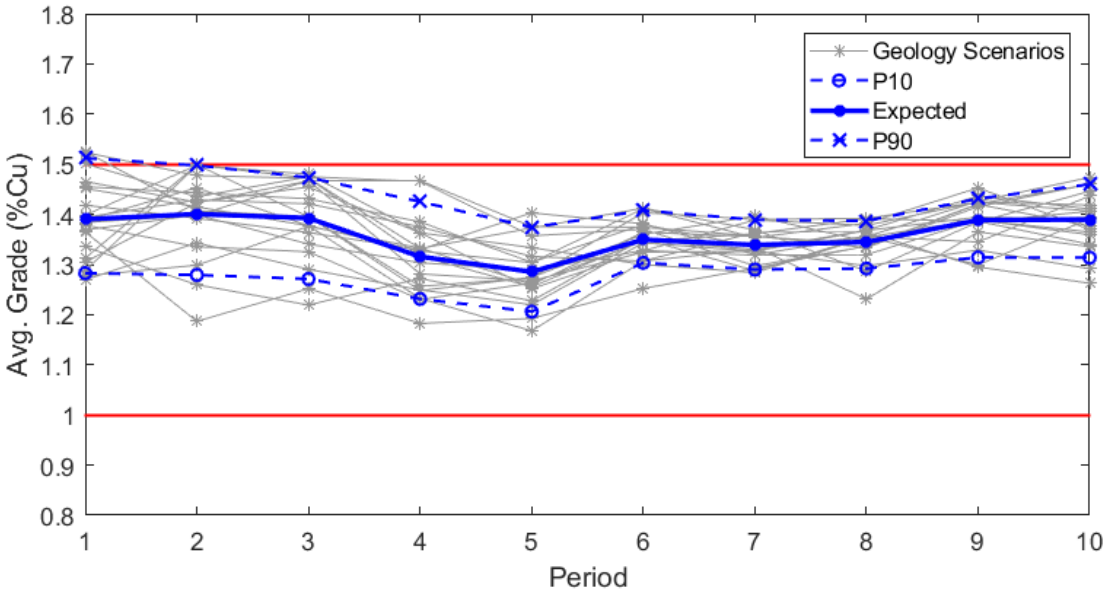


Figure 4-29. Average production grade for the stochastic case with geology uncertainty.

4.5.1 Geological and Material Flow Risk Profile

The stochastic mine plan obtained for the geological uncertainty case was tested against the joint geological and material flow scenarios. This allows to have some insights on the impact of the material flow and mixing uncertainty on the projects economic and operational forecasts.

Figure 4-30 and Figure 4-31 show the uncertainty profile in the cumulative NPV and cash flows. The gray lines represent the response of each geological and flow realization to the stochastic mine sequence under only geological uncertainty, summarized in the expected value, 10th and 90th percentile.

The expected NPV obtained at the end of the mine life for the geological stochastic LOM plan, over the joint geological and material flow scenarios was 607.828 M$. The incorporation of material flow and the resulting dilution would result in an expected NPV 9.64% less than the one reported with the stochastic mine plan under solely geological uncertainty. The reported NPV for the geology uncertainty was well over the 90th percentile of the NPV distribution for the joint geology and flow scenarios, which would result in a very low probability of actually achieving it if dilution was considered based on the parameters used to model it.

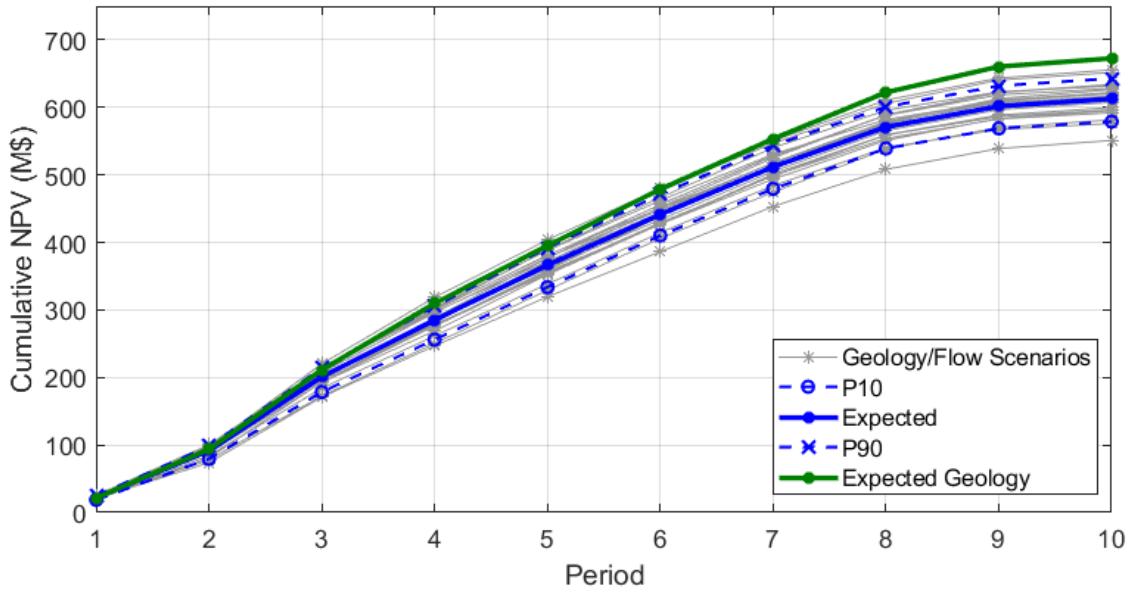


Figure 4-30. Cumulative NPV uncertainty profile for the stochastic mine plan with geological uncertainty over the joint geological and mixing scenarios.

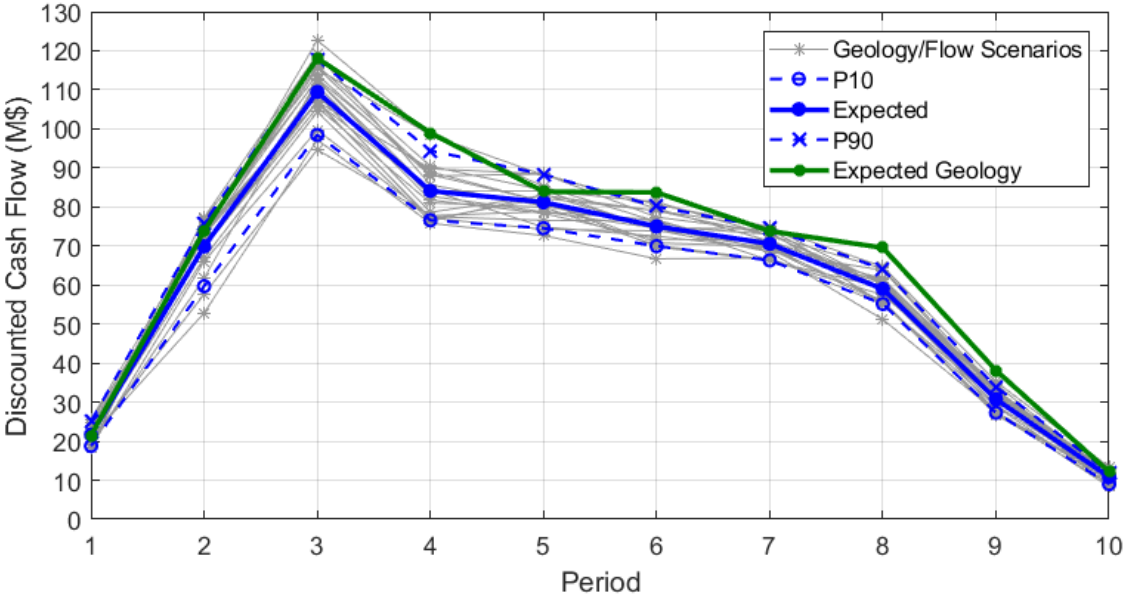


Figure 4-31. Cash flow uncertainty profile for the stochastic mine plan with geological uncertainty over the joint geological and mixing scenarios.

The geology and flow uncertainty profile for the average production grade generated by the stochastic LOM plan under geological uncertainty is shown in Figure 4-32. The average grades are kept within the boundaries for the whole LOM; however the forecasted values differ from the expected ones and are well over the 90th percentile of the distribution resulting from the incorporation of material flow and mixing.

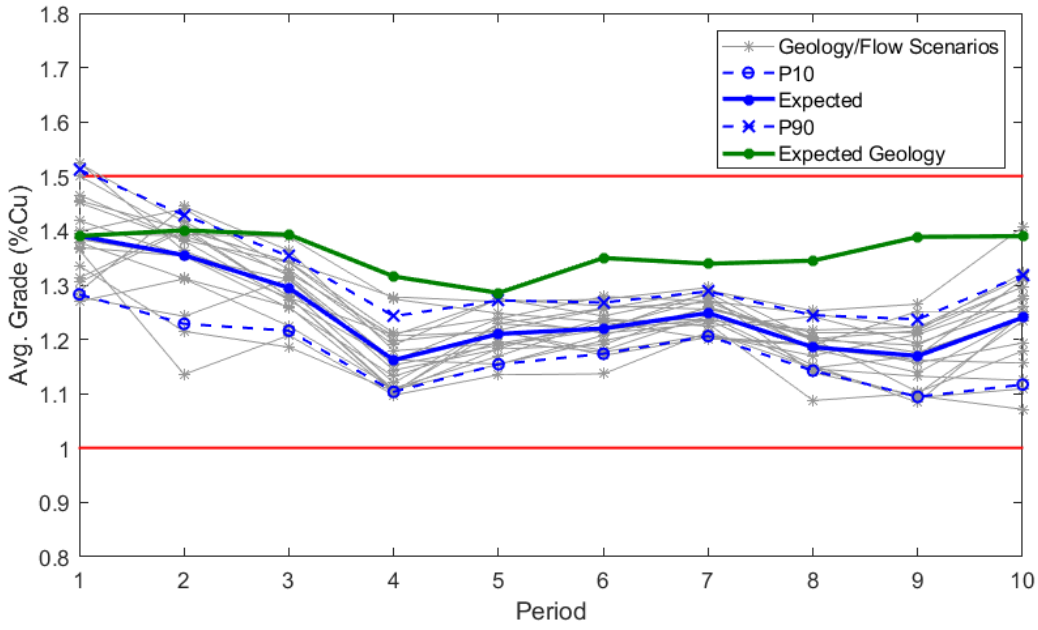


Figure 4-32. Average production grade uncertainty profile for the deterministic mine plan over the geological and flow scenarios.

4.6 Stochastic LOM Schedule with Geological and Flow Uncertainty

This case incorporates the geological and material flow scenarios directly into the optimization framework, to obtain a LOM plan that accounts for both sources of uncertainty. The model was used in the range of undercut elevations between 595 and 645m that was defined as the highest valued caves in the deterministic case, since the cave with the highest NPV can be expected to be within this range, and running the stochastic formulation on the full resource model would be computationally expensive. Table 4-7 shows the NPV and ore tonnage for this range of undercut elevations.

Table 4-7. NPV and ore tonnage for the stochastic optimization with geological and material flow uncertainty at different undercut elevations.

	645m	635m	625m	615m	605m	595m
NPV (M$)	589.290	603.481	596.863	609.694	615.908	591.402
Tonnage (Mt)	24.673	26.215	26.391	26.411	26.408	24.761

For the stochastic optimization with geological and flow uncertainty the undercut elevation that yielded the highest NPV was 605m at 615.908 M$. The most profitable cave when considering both geological and flow uncertainty was located at the same undercut elevation than the one

obtained with the stochastic case under only geological uncertainty, and 30m below the one determined with the deterministic scenario. The stochastic NPV with geological and flow uncertainty is 8.44% lower than the stochastic version with only geological uncertainty, and 11.56% lower than the reported with the deterministic LOM plan. However, this LOM incorporates potential dilution scenarios and adjusts for its impact on the expected economic value of the extracted ore.

Figure 4-33 shows the caving envelope for the stochastic case with geological and flow uncertainty at undercut elevation 605m. For displaying and comparison purposes, the grades shown are the expected copper grade from all the geology realizations, and the deposit model is the deterministic kriged model.

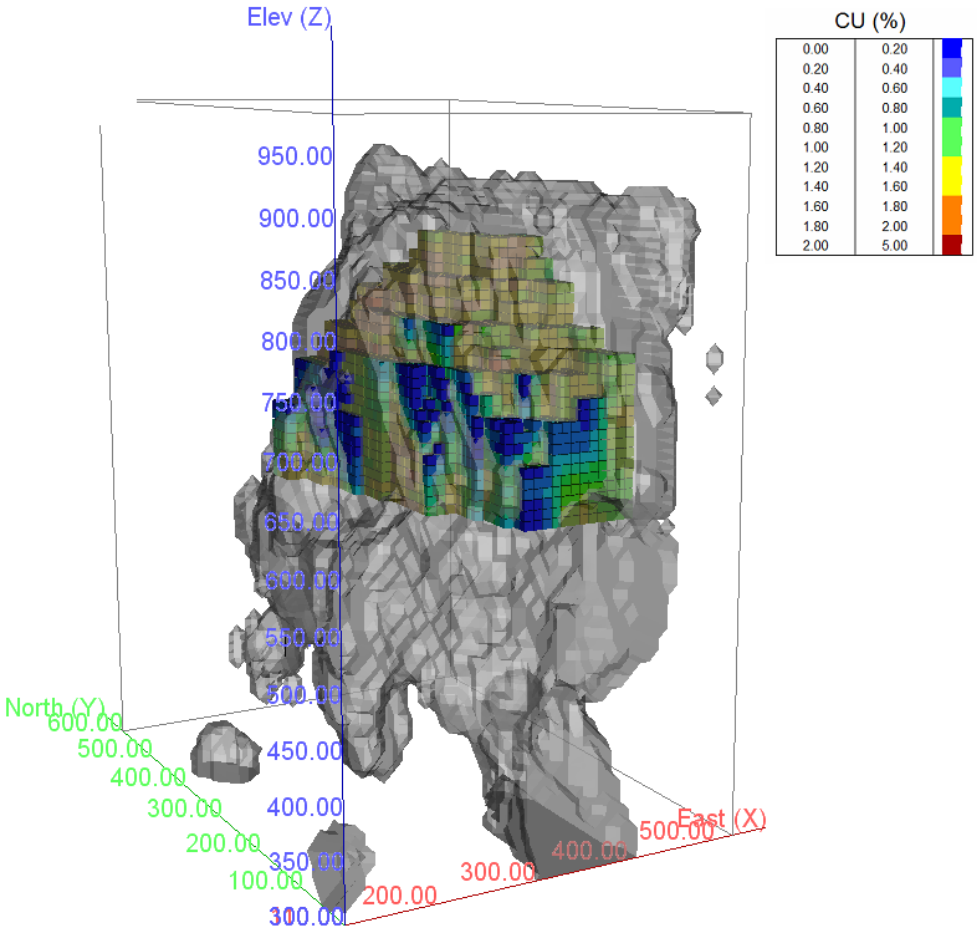


Figure 4-33. Block caving reserves envelope for the stochastic case with geological and flow uncertainty at undercut elevation 605m.

The undercut opening sequence is presented in Figure 4-34. The sequence varies from both the deterministic and stochastic case to accommodate for the variability in grade throughout the

multiple realizations accounting for the potential of mixing during flow. The total footprint area for the stochastic case with geology and flow uncertainty is 109,200 m².

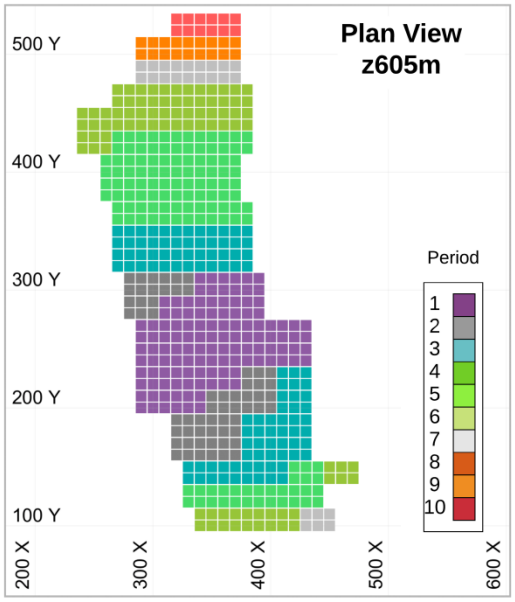


Figure 4-34. Undercut sequence for the stochastic case with geology and flow uncertainty.

Figure 4-35 shows the production schedule for the stochastic case with geological and flow uncertainty. The ore tonnage production, at a total 26.408 Mt meets the production targets established including the ramp-up and ramp-down periods. The average grade is shown along with the defined bounds. The overall average production grade decreases throughout the mine life, as the mining advancement direction approaches the ore/waste boundaries within the different realizations, diluting ore reserves.

The extraction heights of the caving envelope on a period basis are shown in Figure 4-36. This represents the caving profile throughout the LOM of the project. The caving sequence maximizes the NPV while minimizing the deviations incurred due to the geological and mixing uncertainty characterized throughout the multiple realizations.

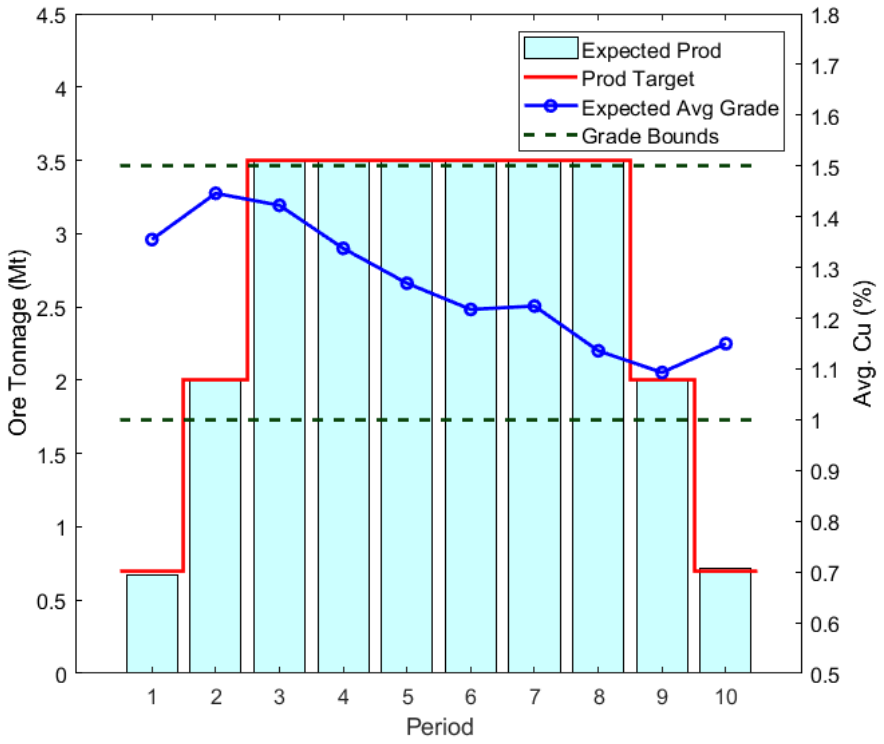


Figure 4-35. Production schedule for the stochastic case with geological and flow uncertainty.

The cumulative NPV and discounted cash flows for the LOM plan are showed in Figure 4-37 and Figure 4-38 respectively. As mentioned above, the expected NPV for the project under geological uncertainty was found at 615.908 M$ with the 10th percentile of its distribution at 577.813 M$ and the 90th percentile at 629.316 M$.

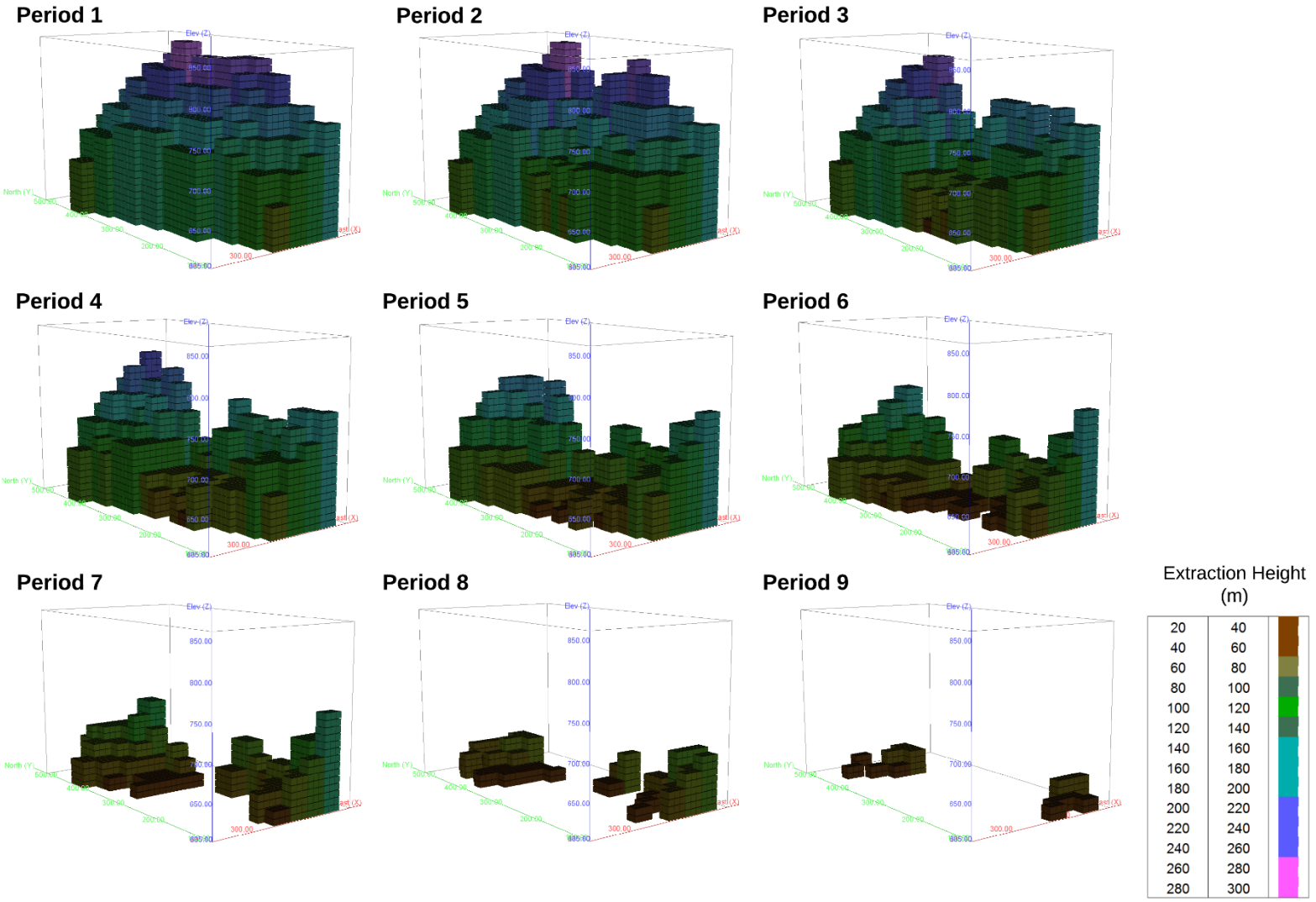


Figure 4-36. Extraction heights for the caving project by period for the stochastic case with geological and flow uncertainty.

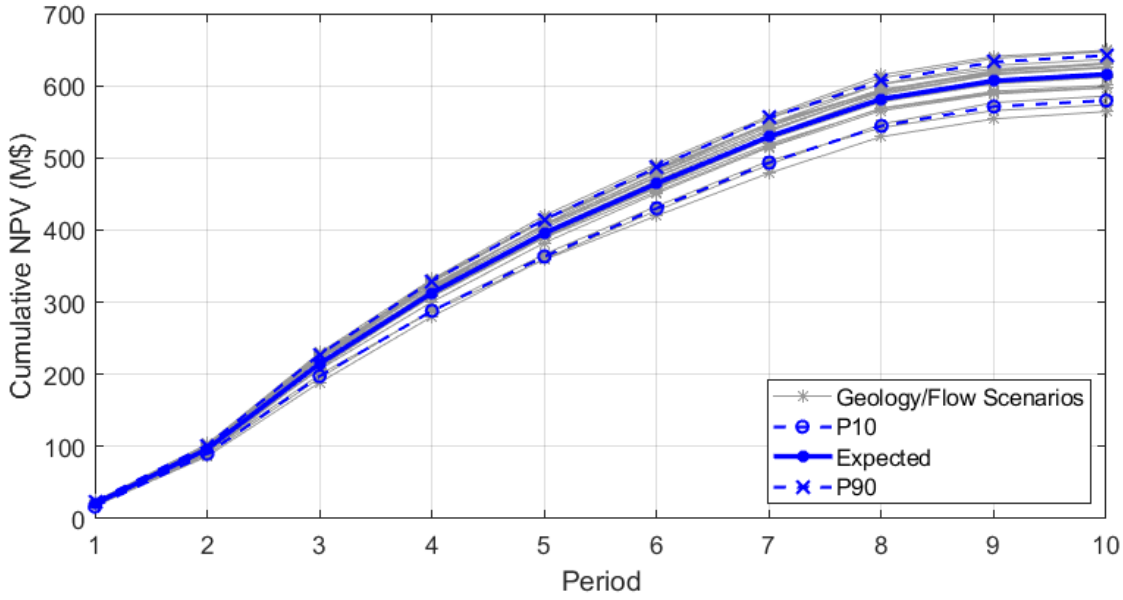


Figure 4-37. Cumulative NPV over the LOM for the stochastic case with geological and flow uncertainty.

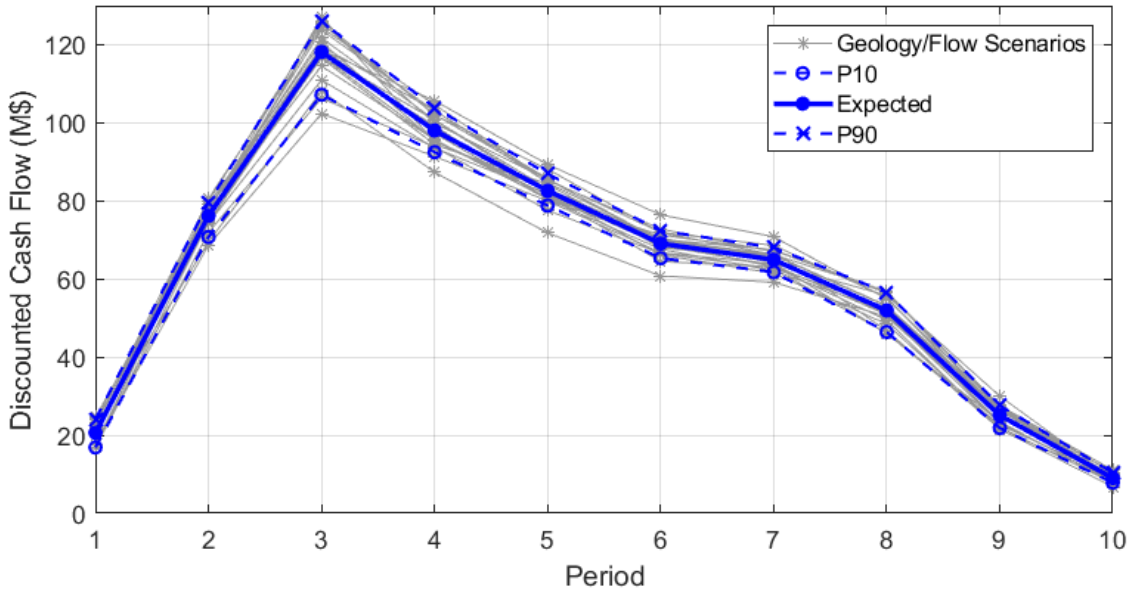


Figure 4-38. Cash Flows over the LOM for the stochastic case with geological uncertainty.

The average production grade throughout the LOM for the multiple geological and flow scenarios is presented in Figure 4-39. All realizations were kept within the boundaries throughout the whole LOM as they were explicitly included in the optimization process.

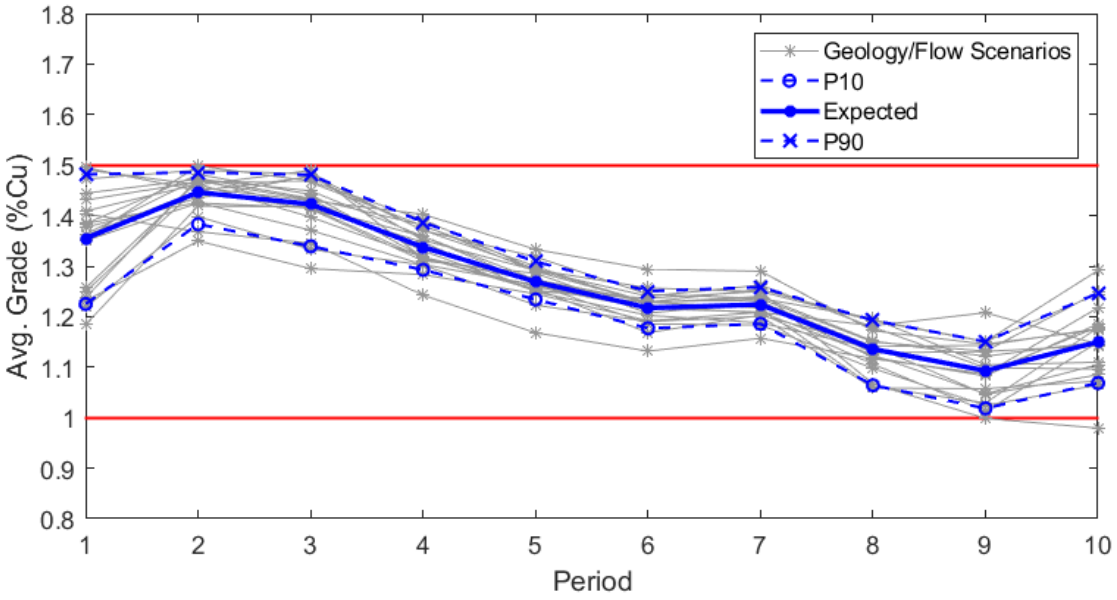


Figure 4-39. Average production grade for the stochastic case with geology uncertainty.

4.7 Comparison of deterministic and stochastic LOM Schedules

The incorporation of geological and material flow uncertainty lead to different LOM plans and economic and operational forecasts for the potential caving project in the case study. One of the main differences lies in the undercut elevation placement, where both stochastic alternatives suggest 605m as the highest profit undercut which is 30m deeper than the 635m obtained by the deterministic model. The placement of the undercut level is not a flexible decision and largely constraints the mining reserves in caving as it dictates the footprint and economic envelope for the cave. In this particular case, incorporating uncertainty allows the decision-makers to make a more informed decision.

Figure 4-40 shows a comparison between the NPV distribution parameters including the expected, 10th and 90th percentiles for the response of the mine plans obtained from the deterministic and both stochastic models to the uncertainty scenarios.

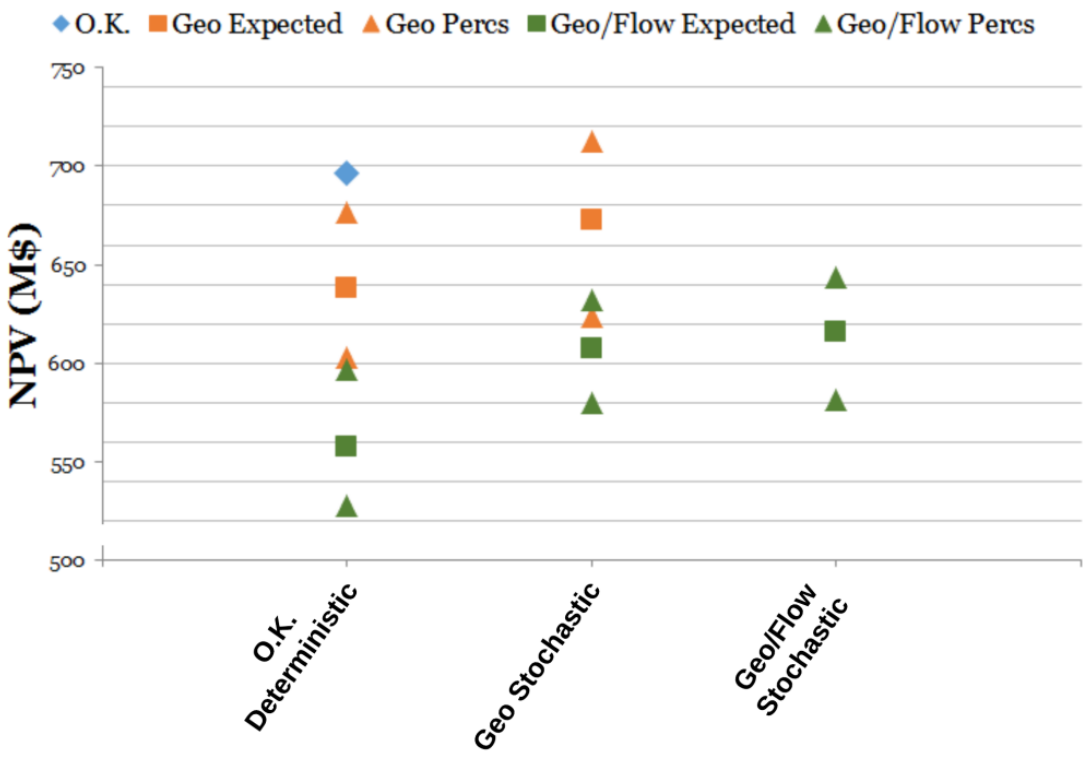


Figure 4-40. NPV distribution comparison between the deterministic and stochastic cases.

The deterministic model with the ordinary kriging estimated orebody provides only a single NPV value that does not incorporate explicitly any source of uncertainty. In this particular application in block caving, the deterministic case yielded the largest NPV forecast, which was likely to be unrealistic. The expected NPV for the deterministic mine plan once geological uncertainties were considered at the current level of knowledge of the deposit was 8.35% lower, and 19.88% lower when material flow and mixing uncertainty was evaluated. Moreover, the NPV for the deterministic mine sequence was larger than the 90th percentiles of the distribution of its response to the uncertainty scenarios, which would lead to a very low probability of actually achieving it.

The incorporation of the geological scenarios in the optimization process led to an expected NPV of 672.726 M$, which was just about 3.41% lower than the reported NPV with the deterministic kriging model. However, it was 5.11% higher than the expected response from the NPV distribution of the deterministic mine plan to the geological uncertainty scenarios. The optimization model, in this case, adjusted the sequence to maximize the expected value of the project accounting for the variability throughout all the geological realizations, providing a more robust estimate and LOM plan in the presence of grade and rock type uncertainty. The geological uncertainty based schedule can be evaluated over the geology and material flow

scenarios to get some insight in the impact of the potential mixing on the economics of the project. The expected NPV due to the geological and material flow scenarios was 9.41% lower than that considering only geological uncertainty. Mixing can potentially have a significant impact on the projects economics if not acknowledged.

The stochastic optimization using the both geological and material flow uncertainties yielded a NPV of 615.908 M$, which was 11.56% lower than the reported by the deterministic mine plan, and 8.44% lower than the reported by the stochastic mine plan considering only geological uncertainty. However, it was on overall higher than the expected NPV from the previous mine plans response to the joint geological and material flow scenarios. The mixing model adapted in this research with the concept of the cone of movement at a block scale support, allowed the potential inclusion of waste and dilution especially at the columns closer to the boundaries of the ore deposit, which would result in lower grades and therefore a lower economic potential. Although the expected NPV from the mine plan that incorporated the joint geology and flow uncertainty was lower than the previous case, it highlighted the impact of potential mixing in the economics of a block caving mine and the importance of incorporating it into the LOM planning workflow explicitly.

While literature vastly reports that stochastic optimization provides larger economic values for LOM plans when compared to deterministic methods in open-pit mining, block caving stochastic mine planning cases are very limited. In Dirkx et al. (2018), the deterministic reported NPV was also lower than both the obtained for the mine plan through the geological and geological and operational uncertainty based optimization. However it would be an unreliable forecast, and in reality it could be expected to be much lower. This could be due to the low flexibility of block caving, which potential alternative sequences are much constrained by the advancement directions and vertical extraction scheme.

Table 4-8 shows a comparison of the results for the three alternative cases, including NPV, ore tonnages, undercut elevation, footprint area and computing time required to run the model. Spaces left in blank represent parameters not applicable for the particular method.

From the footprint area, it can be seen that the successive incorporation of geological and flow uncertainty led to a cave with a larger footprint. The ore tonnage however remained at a similar output with very slight variations between each method. The stochastic based schedules favored the opening of more columns (larger footprint) at the tradeoff of the lower height of draws. Since the cost of opening each PU (column) was already incorporated in the model, the larger footprint paid for its development.

Table 4-8. Comparison of results for case (1): deterministic schedule, case (2): stochastic schedule with geological uncertainty and case (3): stochastic schedule with both geological and material flow uncertainty. Spaces in blank refer to parameters not applicable to that method.

						Geological Uncertainty			Geological/Flow Uncertainty		
	Undercut Elev. (m)	Footprint Area (m2)	NPV (M$)	Ore Ton (Mt)	Comp. Time (min)	E[NPV] (M$)	E[Ore Ton] (Mt)	Footprint Area (m^2)	E[NPV] (M$)	E[Ore Ton] (Mt)	Footprint Area (m^2)
Case (1)	635	92,400	696.49	26.37	12.31	638.33	26.37	92,400	557.99	26.37	92,400
Case (2)	605				332.4	672.73	26.45	106,800	607.83	26.45	106,800
Case (3)	605				985.3				615.91	26.41	109,200

Having a larger footprint represents an advantage as increased knowledge of the deposit in later stages could unlock additional value. Also, this could be an indication that the stochastic schedule can potentially unlock larger mineral reserves and therefore value, for which a longer time period or larger production rates should be evaluated.

A major point of interest is the computing time required to run the different models. The deterministic model at the most profitable undercut elevations takes about 12.31 minutes to run and obtain a solution. However, the incorporation of geology uncertainty in the stochastic case increases the computing time required to 332.4 min or 5.54 hours to run, about 27 times more than the deterministic case. The stochastic case with both geology and flow uncertainty takes about 16.42 hours to run, about 80 times more than the deterministic case and three times more than the stochastic with only geology uncertainty.

The computing times are obtained after the solution strategy based on the early start heuristics to reduce the number of variables and the sliding time windows metaheuristics to solve the problem. This poses a potential challenge in the application of these models as the computing times can be probative, research in the development of alternative strategies to obtain or approximate a solution is needed.

4.8 Summary and Conclusion

This chapter presented the application of the stochastic model development for the LOM planning of block caving mines under geological and material flow uncertainty. The geostatistical framework for the geological simulations was detailed, which used SIS to generate the rock type realizations and SGS for grade simulations which are later merged.

A cone of movement concept was used to generate the material flow scenarios, adapted on the concept presented by Khodayari (2018). Based on horizontal displacements (m) and a vertical slip angle (degrees), the movement of material as ore is drawn can follow multiple scenarios. In this research, the cone of movement concept was adapted to a block scale support. For each MU, there is a cone defined by the mentioned parameters that contains the individual blocks (from itself, other MU and waste in the boundaries of the orebody) that could potentially flow into it as it is extracted. A random sampling of these candidate blocks is carried out to “fill” each MU and update its ore tonnage and average grade.

The operational and economic parameters used for the case study were defined to represent an average block caving mine based on reported data from operations around the world. The LOM

schedule was generated for the deterministic case and for the stochastic case for both only geological uncertainty and joint geological and material flow uncertainty.

The most profitable undercut elevation for the deterministic case was found at 635m, while both stochastic alternatives found that 605m yielded the best expected NPV. For the deterministic case the NPV was found at 696.49 M$, however when its LOM plan is evaluated over the geological scenarios and geological and flow scenarios the expected NPVs were found at 8.35% and 19.88% lower respectively. Moreover, the deterministic mine plan NPV is higher than the 90th percentile of its distribution over the uncertainty scenarios, which means there is little probability of actually achieving it.

The stochastic optimization with only geological uncertainty yielded an expected NPV of 672.73 M$, while the expected NPV of the joint geological and material flow optimization model generated a mine plan with an expected NPV of 615.91 M$. The incorporation of uncertainties in the optimization process generates LOM plans with reduced NPVs in relation to the deterministic case, however, it provides more reliable estimates and a sequence adapted to deal with the potential variability within the orebody. The expected NPVs obtained in the successive stochastic optimization cases are higher than the response of the deterministic LOM schedule to the uncertainty scenarios.

The lower economic values can also be due to the low flexibility of block caving mines as the potential alternative sequences are largely constrained by the mining advancement direction and vertical extraction scheme.

A larger footprint is also observed on the stochastic model in comparison to the deterministic case. Since the ore tonnage variation is negligible between each method, the stochastic models favor the opening of larger undercut areas on lower height of draws.

The computing times required to solve the stochastic alternatives are significantly higher than the required for the deterministic case. The stochastic schedule with geological uncertainty requires a computing time of 5.54 hours and the joint geological and material flow uncertainty a total of about 16.42 hours, while the deterministic model takes about 12.31 minutes. This could be a prohibitive constraint in large projects, and further research should be aimed in finding more efficient solution approaches.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

Chapter 5 provides the contribution and significance of this research, as well as the conclusions reached and some recommendations for further studies.

5.1 Summary of Research

This research presents a mathematical programming model that optimizes the LOM plan and defines the boundary of a block caving mine, from a deposit block model scale, under geological and material flow uncertainty.

The optimization framework works over two steps: it initially aggregates the individual blocks into production units based on desired drawpoint spacing, representing the draw columns, and mining units based on the minimum draw rate, representing the slices that are commonly used in block caving mine planning. The mining units then become the basic scheduling unit for the stochastic integer programming scheduling model. Uncertainty is characterized by the development of multiple numerical deposit simulations. Geological simulations are developed using geostatistical simulation techniques, with sequential indicator simulation for rock types and sequential Gaussian simulation for grades. Material flow uncertainty is integrated by the concept of a cone of movement. As each mining unit is extracted, it leaves a void that can be filled by any fraction of the material on its surroundings based on the flow properties of the broken rock mass. A cone, based on potential horizontal displacement and vertical slip angle of the broken rock mass is used to generate grade and tonnage mixing scenarios for each mining unit. The cone is placed at the bottom of each mining unit, and a random sample of the blocks from the deposit model that are contained within it, "filling" the mining unit, is used to update its grade and tonnage. This allows for scenarios where each mining unit material could potentially be part of fractions of adjacent units as well as waste blocks at the orebody accounting for dilution.

The stochastic mathematical model takes as an input the set of geological and material flow simulations to generate a single best schedule that maximizes the expected economic value from the uncertainty sources, while minimizes the deviations incurred in production and average grade targets due to the variability between the potential scenarios. The operational constraints considered in the model include mining capacity targets, average production grade, minimum and maximum heights of draw, minimum and maximum vertical draw rates, undercut development rate, maximum adjacent relative height of draw, mining precedence both horizontal and vertical, and mineral reserves. The model was tested on a case study with parameters defined to be representative of current practices in caving operations worldwide.

5.2 Conclusions

The main conclusions of this research are summarized as follows:

- A mathematical programming framework was developed to generate a LOM schedule for block caving mines from a block model scale. The framework starts with the aggregation of the individual blocks into PU and MU representative of the slice and column models used in block caving mine planning, and then scheduling based on common operational constraints. This provides a flexible tool that produces results including economic and operational KPIs on period basis from the production schedule as well as defining the optimal mining boundaries.
- Geological uncertainty is modeled using geostatistical simulation techniques while the material flow uncertainty is modeled using the concept of a cone of movement based on the definition of potential horizontal displacement and vertical slip angle to characterize the movement of the broken rock mass. It provides a simple procedure to introduce 3D mixing as a set of scenarios.
- A deterministic based LOM plan yields economic forecasts that when evaluated over multiple geological and material flow scenarios show a significant overestimation with little probability of actually being achievable. For the case study, the deterministic mine plan would lead to an expected NPV 8.35% and 19.88% lower than the reported one when evaluated under geological and geological and material flow uncertainty. Moreover, the average production grade for the deterministic based schedule would lead to operational problems when evaluated over uncertainty scenarios as it drops below the desired minimum target at later periods. This highlights the problems associated to a LOM plan developed using a single estimated model.
- Stochastic based LOM plans incorporate explicitly the different uncertainty simulations to maximize the expected value rather than a single estimate. Also, it allows for better control of the operational parameters such as average production grade over the multiple realizations. For the case study, the geology uncertainty based schedule yielded an expected NPV of 672.73 M$, and the geology and material flow uncertainty based schedule an expected NPV of 615.91 M$. The stochastic optimization generates lower NPVs than the deterministic case; however these account for the multiple potential scenarios and therefore are more reliable forecasts. The lower NPVs could be due to the low flexibility in block caving mines. Also, the expected NPV obtained from the stochastic schedules is higher than the expected NPV from the deterministic mine plan when evaluated over the uncertainty scenarios.

- The stochastic based schedules generate a larger footprint area than the deterministic case. Since the production rates and life of mine is kept unchanged for the scenarios presented, it could be expected that additional reserves can be unlocked with the stochastic schedule at a larger life of mine or production rate.
- The computing time required for the stochastic based optimization tool is significantly higher than that required for the deterministic case. This could become a prohibitive constraint in larger operations, which should be addressed.

5.3 Contributions of the Research

The main contribution of this research lies in an alternative method to define block caving LOM production schedules and boundaries that explicitly incorporate geological and material flow uncertainty. The results of this thesis can be compiled in prototype software for testing on caving projects and assist decision-makers.

Material flow has been pointed out as a major challenge in the forecasting of the operational and economic output of a caving project. This research presents a framework in which geological scenarios are generated through geostatistical simulation, and material flow scenarios with the concept of a cone of movement. A mathematical programming model is then developed that takes as an input a set of geology/flow simulations to generate a LOM plan that explicitly incorporates the mentioned sources of uncertainties.

The main application of the presented technique would be in prefeasibility stage of the projects, where an economic and operational valuation is required for a deposit with a little level of knowledge. The method allows for operational caving constraints defined in simple expressions and provides uncertainty based LOM plans and economics estimates representative of a block caving mine.

5.4 Recommendations for Future Research

The following suggestions for future research address the limitations and challenges presented during the development of this thesis, as well as new potential research areas.

- Research into a more detailed mixing concept for the development of the material flow scenarios. However, the model is aimed to work at a block model scale and for prefeasibility evaluations before detailed engineering studies and the available mixing algorithms are all aimed at an operational level. Studying the possibility to scale back some of these mixing methods to generate a valid representative version

at this level of detail would be of interest for the stochastic optimization of block caving mines.

- The research incorporates the mixing uncertainty as a set of scenarios, therefore it does not interact or responds with the scheduling decisions. Exploring the possibility of developing a tool which can optimize the schedule in a dynamic way with the material flow would be of interest.
- Incorporate additional sources of uncertainty such as price and operational parameters to provide a more robust mining schedule. At the moment, the formulation allows for geological and material flow uncertainty only and its impact on the economic value of the project, and ore production and average grade targets.
- Extend the model to allow for multiple production lifts scenarios. Although multiple lifts could be considered as different problems and the model here presented could be used multiple times on the same ore deposit model, a framework that allows for the optimization of the placement and envelope of multiple production lift operations would be of interest for larger projects.
- Develop more efficient algorithms and solution approaches to solve the optimization problem. The time required to compute a solution for the geological and material flow stochastic schedule can be prohibitive. Metaheuristic techniques or other computing techniques could be useful to find an approximate solution that is good enough at faster processing times.

BIBLIOGRAPHY

Ahmed, H., Scoble, M., & Dunbar, S. (2014). A comparison between Offset Herringbone and El Teniente underground cave mining extraction layouts using a discrete event simulation. *International Journal of Mining, Reclamation and Environment*, *30*(2), 71–91.

Alfaro, A., & Saavedra, J. (2004). Predictive models for gravitational flow. In *Proceedings of MassMin 2004* (p. 179).

Alford, C. (1995). Optimization in underground mine design. In *Proceedings of the 25th International Symposium on the Application of Computers and Operations Research in the Mineral Industry (APCOM)* (pp. 729–739). Brisbane, Australia: AusIMM.

Alonso-Ayuso, A., Carvallo, F., Escudero, L., Guignard, M., Pi, J., Puranmalka, R., & Weintraub, A. (2014). Medium range optimization of copper extraction planning under uncertainty in future copper prices. *2European Journal of Operational Research*, *233*(3), 711–726.

Alvial, J. (1992). Analysis of the extraction at El Teniente 4Sur LHD. In *Proceedings of MassMin 1992*. The South African Institute of Mining and Metallurgy.

Annavarapu, S. (2019). *Estimating Primary Fragment Size Distributions from Drill Hole Data*. University of Arizona.

Askari-Nasab, H., Pourrahimian, Y., Ben-Awuah, E., & Kalantari, S. (2011). Mixed integer linear programming formulations for open-pit production scheduling. *Journal of Mining Science, 47*(338).

Bieniawski, Z. (1976). Rock mass classification in rock engineering. In *Proceedings for exploration for rock engineering* (pp. 97–106).

Brannon, C., Carlson, G., & Casten, T. (2011). Block Caving. In P. Darling (Ed.), *SME Mining Engineering Handbook* (Third Ed., pp. 1437–1451). Society for Mining, Metallurgy and Exploration.

Brown, E. T. (2007). *Block Caving Geomechanics*. Julius Kruttschnitt Mineral Research Centre, The University of Queensland.

Brunton, I., Sharrock, G., & Lett, J. (2012). Full scale near-field flow behavior at the Ridgeway

Deeps Block cave mine. In *Proceedings of MassMin 2012*. Montreal.

Castro, R. (2006). *Study of mechanisms of gravity flow for block caving*. The University of Queensland.

Castro, R., Gonzalez, F., & Arancibia, E. (2009). Development of a gravity flow numerical model for the evaluation of drawpoint spaceing for block/panel caving. *Journal of the South African Institute of Mining and Metallurgy*, *109*, 393–400.

Castro, R. L., & Paredes, P. S. (2014). Empirical observations of dilution in panel caving. *Journal of the South African Institute of Mining and Metallurgy2*, *114*.

Castro, R., Trueman, R., & Halim, A. (2007). A study of isolated draw zones in block caving mines by means of a large 3D physical model. *International Journal of Rock Mechanics and Mining Sciences*, *44*(6), 860–870. https://doi.org/10.1016/j.ijrmms.2007.01.001

Chanda, E. C. K. (1990). An application of integer programming and simulation to production planning for a stratiform ore body. *Mining Science and Technology*, *11*(2), 165–172. https://doi.org/10.1016/0167-9031(90)90318-M

Cullenbine, C., Kevin Wood, R., & Newman, A. (2011). A sliding time window heuristic for open-pit mine block sequencing. *Optimization Letters*, *5*, 365–377.

Cundall, P., Mukundakrishnan, B., & Lorig, L. (2000). *Cundall,. REBOP (Rapid Emulator Based on PFC3D) formulation and user's guide*.

Deserable, D. (2006). A Versatile two-dimensional cellular automata network for granular flow. *SIAM Journal on Applied Mathematics*, *62*(4), 1414–1436.

Deutsch, C., & Journel, A. (1997). *GSLIB: Geostatistical Software Library and User's Guide* (Second Ed.). Oxford University Press.

Deutsch, M., Gonzalez, E., & Williams, M. (2015). Using Simulation to Quantify Uncertainty in Ultimate Pit Limits and Inform Infrastructure Placement. In *SME Annual Meeting*.

Diering, T. (2000). PC-BC: A block cave design and draw control system. In *Proceedings MassMin 2000*. Brisbane, Australia: Australasian Institute of Mining and Metallurgy.

Diering, T. (2004). Computational considerations for production scheduling of block cave mines. In *Proceedings of MassMin 2004*. Santiago, Chile.

Diering, T., Richter, O., & Villa, D. (2010). Block cave production scheduling using PCBC. In *SME Annual Meeting*. Phoenix, AZ: Society for Mining, Metallurgy and Exploration.

Dimitrakopoulos, R. (2011). Stochastic optimization for strategic mine planning: A decade of developments. *Journal of Mining Science2*, *47*(2), 138–150.

Dimitrakopoulos, R., Farrelly, C., & Godoy, M. (2002). Moving forward from traditional optimization: grade uncertainty and risk effects in open-pit design. *MIning Technology2*, *111*(1), 82–88.

Dimitrakopoulos, R., & Ramazan, S. (2008). Stochastic integer programming for optimizing long term production schedules of open-pit mines: methods, application and value of stochastic solutions. *Transactions of the Institution of Mining and Metallurgy, Section A: Mining Technology*, *117*(4).

Dimitrakopoulos, Roussos, & Ramazan, S. (2008). Stochastic integer programming for optimising long term production schedules of open-pit mines: Methods, application and value of stochastic solutions. *Mining Technology*, *117*(4), 155–160.

Dirkx, R., Kazakidis, V., & Dimitrakopoulos, R. (2018). Stochastic optimisation of long-term block cave scheduling with hang-up and grade uncertainty. *International Journal of Mining, Reclamation and Environment*, *0930*, 1–18. https://doi.org/10.1080/17480930.2018.1432009

Elkington, T., Bates, L., & Richter, O. (2012). Block Caving Outline Optimisation. In *Proceedings MassMin 2012*. Sudbury, Canada: Canadian Institute of Mining, Metallurgy and Petroleum.

Esterhuizen, G., & Laubscher, D. (1992). A comparative evaluation of production level layouts. In *Proceedings of MassMin 19921* (pp. 63–70).

Fuentes, S., Villegas, F. (2014). Block caving using macro blocks. In R. Castro (Ed.), *Block Caving using macro blocks* (pp. 211–216). Santiago, Chile: University of Chile.

Garces, D., Viera, E., & Castro, R. (2016). Gravity flow full scale tests at Esmeralda Mine's Block 2, El Teniente. In *Proceedings of MassMin 2016*. Sydney.

Godoy, M. (2016). *The Effective management of geologic risk in mining: Mine planning under geologic uncertainty*. LAP LAMBERT Academic Publishing.

Goovaerts, P. (1997). *Geostatistics for natural resources evaluation*. Oxford University Press.

Guest, A., G., V. H., & Von Johannides, A. (2000). An application of linear programming for block cave draw control. In *Proceedings of MassMin 20002*. Brisbane, Australia.

Halim, A. (2006). *Study of the influence of interactive draw upon drawpoint spacing in*

block and sublevel caving mines. The University of Queensland.

Hustrulid, W., Kuchta, M., & Martin, R. (2013). *Open-pit mine planning and design* (3rd Editio). CRC Press.

Jakubec, J., & Laubscher, D. (2000). The MRMR Rock Mass Rating Classification System in Mining Practice. In *Proceedings of MassMin 2000* (pp. 413–421).

Janelid, I. (1966). Sublevel Caving. *International Journal of Rock Mechanics and Mining Science1*, 129–153.

Janelid, I. (1975). Study of the gravity flow process in sublevel caving. In *Proceedings of Sublevel Caving Symposium*. Stockolm, Sweden.

Khodayari, F. (2018). *Incorporation of 3-D mixing in long-term production scheduling optimization for block caving mines*. University of Alberta.

Khodayari, F., & Pourrahimian, Y. (2014). Determination of the best height of draw in block cave sequencing optimization. In *3rd International Symposium on Block and Sublevel Caving*. Santiago, Chile.

Khodayari, F, & Pourrahimian, Y. (2019). Long-term production scheduling optimization and 3D material mixing analysis for block caving mines. *Mining Technology Transactions*, *6668*. https://doi.org/10.1080/25726668.2018.1563742

Khodayari, Firouz, & Pourrahimian, Y. (2015). Determination of development precedence for drawpoints in bloc-caving mining. In *5th International Symposium Mineral Resources and Mine Development*. Aachen, Germany.

Koushavand, B. (2014). *Long-term mine planning in presence of grade uncertainty*. University of Alberta.

Kvapil, R. (1960). Flow of Granular Materials. *Gosgortekhizdat-Moscow*.

Kvapil, R. (1964). Problems of gravity flow of bulk materials. *Aufbereitungs-Tecnik*, *3*, 139–146.

Kvapil, R. (1965). Gravity flow of granular materials in Hoppers and Bins. *International Journal of Rock Mechanics and Mining Science1*, *2*, 35–41; 277–304.

Kvapil, R. (1982). The mechanics and design of sublevel caving systems. In W. Hustrulid (Ed.), *Underground Mining Methods Handbook* (pp. 880–897). Littleton, USA: American Institute of Mining, Metallurgical and Petroleum Engineers.

Lamghari, A., & Dimitrakopoulos, R. (2016). Progressive hedging applied as a metaheuristic to schedule production in open-pit mines accounting for reserve uncertainty. *2European Journal of Operational Research*, *253*(3), 843–855.

Laubscher, D. (1990). A Geomechanics Classification System for the Rating of Rock Mass in Mine Design. *Journal of the South African Institute of Mining and Metallurgy*, *90*(10), 257–271.

Laubscher, D. (1994). Cave mining-the state of the art. *The Journal of the South African Institute of Mining and Metallurgy*, 279–293.

Laubscher, D. (2000). *A practical manual on block caving*. Brisbane, Australia: International Caving Study, JKMRC and Itasca Consulting Group, Inc.

Laubscher, D., Guest, A., Jakubec, J., & Chitombo, G. (2017). *Guidelines on Caving Mining Methods - The Underlying Concepts*. Brisbane, Australia: W.H. Bryan Mining and Geology Research Centre.

Laubscher, D. H. (2011). Cave Mining. In P. Darling (Ed.), *SME Mining Engineering Handbook* (Third Edit, pp. 1385–1397). Society for Mining, Metallurgy and Exploration.

Leite, A., & Dimitrakopoulos, R. (2007). A stochastic optimization model for open-pit mine planning: Application and risk analysis at a copper deposit. *Transactions of Mining and Metallurgy: Mining Technology*, *116*(3).

Liu, Y. (2016). *Characterisation of Block Cave mining secondary fragmentation*. University of British Columbia.

Malaki, S., Khodayari, F., Pourrahimian, Y., & Liu, W. (2017). An application of mathematical programming and sequential Gaussian simulation for block cave production scheduling. In M. Hudyma & Y. Potvin (Eds.), *First International Conference on Underground Mining Technology, 2017 11-13 October, Sudbury* (pp. 323–337). Perth: Australian Centre for Geomechanics. Retrieved from https://papers.acg.uwa.edu.au/p/1710_25_Pourrahimian/#.Wl_kiQAZh-8.mendeley

Marano, G. (1980). The interaction between adjoining drawpoints in free flowing materials and its application to mining. *Chamber of Mines Journal, Zimbabwe*, *22*, 25–32.

Newman, A., Rubio, E., Caro, R., Weintraub, A., & Eurek, K. (2010). A review of operations research in mine planning. *Interfaces*, *40*(3), 222–245.

Nezhadshahmohammad, F., Aghababaei, H., & Pourrahimian, Y. (2017). Conditional draw

control system in block-cave production scheduling using mathematical programming. *International Journal of Mining, Reclamation and Environment*, *0930*, 1–24. https://doi.org/10.1080/17480930.2017.1385155

Nezhadshahmohammad, F., Pourrahimian, Y., & Aghababaei, H. (2018). Presentation of a multi-index clustering technique for the mathematical programming of block-cave scheduling. *International Journal of Mining Science and Technology2*, *28*, 941–950.

Parkinson, A. (2012). *Essays on sequence optimization in block caving mining and inventory policies with two delivery sizes*. The University of British Columnbia.

Pierce, M. (2010). *A model for gravity flow of fragmented rock in block caving mines*. The University of Queensland.

Pourrahimian, Y., Askari Nasab, H., & Tannant, D. (2013). A multi-step approach for block-cave production scheduling optimization. *International Journal of Mining Science and Technology*, *23*(5), 739–750. https://doi.org/10.1016/j.ijmst.2013.08.019

Rahal, D., Smith, M., Von Hout, G., & Von Johannides, A. (2003). The use of mixed integer linear programming for long-term scheduling in block caving mines. *Application of Computers and Operations Research in the Minerals Industries*, 123–132. Retrieved from http://saimm.org.za/Conferences/Apcom2003/123-Rahal.pdf

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

Rimele, M., Dimitrakopoulos, R., & Gamache, M. (2018). A stochastic optimization method with in-pit waste and tailings disposal for open-pit life-of-mine production planning. *Resources Policy*, *57*, 112–121.

Rodriguez, M. (2018). *Optimizacion del plan de produccion conjunto cielo abierto - subterranea [Optimization of the production schedule for coupled open-pit and underground mining systems]*. University of Chile.

Rossi, M., & Deutsch, C. (2014). *Mineral Resource Estimation*. Springer Netherlands.

Rubio, E. (2002). *Long Term Planning of Block Caving Operations Using Mathematical Programming Tools*. University of British Columbia.

Rubio, E., Caceres, C., & Scoble, M. (2004). Towards an integrated approach to block cave planning. In *Proceedings of MassMin 2004* (pp. 128–134).

Sepúlveda, E., Dowd, P. A., & Xu, C. (2018). The optimisation of block caving production scheduling with geometallurgical uncertainty–a multi-objective approach. *Mining Technology: Transactions of the Institute of Mining and Metallurgy*, *127*(3), 131–145. https://doi.org/10.1080/25726668.2018.1442648

Sinha, S. (2006). *Mathematical Programming: Theory and Methods*. Elsevier Science.

Smoljanovic, M. (2012). *Optimum sequencing of underground ore reserves for different mining systems*. University of Chile.

Song, X. (1989). Caving process simulation and optimal mining sequence at Tong Kuang Yu mine. In *21st International Symposium on Applications of Computers and Operations Research in the Mineral Industry*. Las Vegas, USA.

Trueman, R., Castro, R., & Halim, A. (2008). Study of multiple draw-zone interaction in block caving mines by means of a large 3D physical model. *International Journal of Rock Mechanics and Mining Sciences2*, *45*, 1044–1051.

Vargas, E., Morales, N., & Emery, X. (2014). Footprint and economic envelope calculation for Block/Panel caving mines under geological uncertainty. In R. Castro (Ed.), *Proceedings of the 3rd International Symposium on Blcok and Sublevel Caving* (pp. 449–456). Santiago, Chile: University of Chile.

Villa, D. (2014). Mine sequence optimization for block caving using concept of "best and worst case." In R. Castro (Ed.), *Proceedings of the 3rd International Symposium on Blcok and Sublevel Caving* (pp. 426–436). Santiago, Chile: University of Chile.

Weintraub, A., Pereira, M., & Schultz, X. (2008). A priori and a posteriori aggregation procedures to reduce model size in MIP mine planning models. *Electronic Notes in Discrete Mathematics*, *30*, 297–302.

Whittle, J. (1999). A decade of open-pit mine planning and optimisation - The craft of turning algorithms into packages. In *Proceedings of the 28th International Symposium on the Application of Computers and Operations Research in the Mineral Industry APCOM*. Golden, USA: Colorado School of Mines.

Williams, H. (2013). *Model building in mathematical programming* (5th Ed.). Wiley.

APPENDIX A

MATLAB Codes

Programming Description

This appendix includes the MATLAB codes developed for the implementation of the optimization framework presented in this research. The codes presented here include the data reading and preparation, which is done from Microsoft EXCEL files containing all the desired parameters, the construction and solving of the resulting model. Plotting functions were developed for personal use on the validation and vertification of the results, these are not included here. The code is presented in a series of scripts, which are later run sequentially on a main script for the aggregation step and scheduling step.

The main scripts and the general code presented here is for the full stochastic optimization framework (aggregation and scheduling steps) with geological and material flow uncertainty. Running the model on deterministic or only geological uncertainty cases would require taking off some of the functions presented here.

The input files required are three Microsoft EXCEL files: the block model file with all the information including blocks coordinates, indices and numerical values for all the geological realizations; the technical and operational parameters for the caving scenario; the production and average grade targets.

f1_ readPar

```
function f1_readPar(flRoot)

    [parameters,roots]=xlsread(flRoot);


params=struct('bHeight',parameters(1),'minHeight',parameters(2),'maxHeight'
,parameters(3),...
        'xIndexCol',parameters(4),'yIndexCol',parameters(5),...

'zIndexCol',parameters(6),'xCoordCol',parameters(7),'yCoordCol',parameters(
8),...

'zCoordCol',parameters(9),'gradeCol',parameters(10),'tonCol',parameters(11)
,'price',parameters(12),...

'sCost',parameters(13),'mCost',parameters(14),'pCost',parameters(15),'rec',
parameters(16),...

'disRate',parameters(17),'eRate',parameters(18),'S',parameters(19),'oreCode
Col',parameters(22),...

'oreCode',parameters(23),'minDR',parameters(24),'sizeX',parameters(25),'siz
eY',parameters(26),'ped',parameters(27),...
        'dilution',parameters(28));
    params.SimFolderRoot=roots{20,3};
    params.KrigBMRoot=roots{21,3};

     save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\params.mat','params')


end
```

f2_readSimulations

```
function f2_readSimulations

    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\params.mat')
    S=params.S;
    for s = 1:S
        blkFile = strcat(params.SimFolderRoot,'\Real',int2str(s),'.xlsx');
        data=xlsread(blkFile);
        noBlocks=length(data);
        if s == 1

blockModel(noBlocks)=struct('blockIndex',[],'xIndex',[],'yIndex',[],'zIndex
',[],...
            'xCoord',[],'yCoord',[],'zCoord',[],'grade1',[],'ton1',[]);
        else
            blockModel(noBlocks).(strcat('grade',int2str(s)))=[];
            blockModel(noBlocks).(strcat('ton',int2str(s)))=[];
        end

        for block=1:noBlocks
            blockModel(block).blockIndex=block;
            blockModel(block).xIndex=data(block,params.xIndexCol);
            blockModel(block).yIndex=data(block,params.yIndexCol);
            blockModel(block).zIndex=data(block,params.zIndexCol);
            blockModel(block).xCoord=data(block,params.xCoordCol);
            blockModel(block).yCoord=data(block,params.yCoordCol);
            blockModel(block).zCoord=data(block,params.zCoordCol);

blockModel(block).(strcat('grade',int2str(s)))=data(block,params.gradeCol);

blockModel(block).(strcat('ton',int2str(s)))=data(block,params.tonCol);
            blockModel(block).oreCode=data(block,params.oreCodeCol);
        end
        fprintf('Finished Reading Block Model Realization %d\n',s)
    end
    gradeAvg = zeros(1,noBlocks);
    tonSum = zeros(1,noBlocks);
    for s=1:S
        gradeAvg =
gradeAvg+([blockModel.(strcat('grade',int2str(s)))].*[blockModel.(strcat('t
on',int2str(s)))]);
        tonSum = tonSum+([blockModel.(strcat('ton',int2str(s)))]);
    end
    temp = num2cell(gradeAvg./tonSum);
    [blockModel.Egrade] = temp{:};
    temp = num2cell(tonSum/S);
    [blockModel.Eton] = temp{:};
    dataKrig=xlsread(params.KrigBMRoot);
    noBlocks=length(dataKrig);
    for block=1:noBlocks
        blockModel(block).grade=dataKrig(block,params.gradeCol);
        blockModel(block).ton=dataKrig(block,params.tonCol);
    end
    fprintf('Finished reading Kriged Block Model')
```

```
    save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\blockModel.mat','blockModel')
end
```

f3_ucutModel

```
function f3_ucutModel(ucutLevel)

    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\blockModel.mat')
    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\params.mat')
    ucutModel=blockModel([blockModel.zIndex]<=ucutLevel);
    ore=ucutModel([ucutModel.Egrade]>0);
    minX=min([ore.xIndex]);
    maxX=max([ore.xIndex]);
    minY=min([ore.yIndex]);
    maxY=max([ore.yIndex]);
    minZ=min([ore.zIndex]);
    maxZ=max([ore.zIndex]);

    [ucutModel(:).ucutIndex]=deal(0);
    count=0;
    for i=1:length(ucutModel)
        count=count+1;
        ucutModel(i).ucutIndex=count;
    end

    level=max([ucutModel.zIndex]);
    ucut=ucutModel([ucutModel.zIndex]==level);
    ore=ucut([ucut.Egrade]>0);
    minX=min([ore.xIndex]);
    maxX=max([ore.xIndex]);
    minY=min([ore.yIndex]);
    maxY=max([ore.yIndex]);
    ucut=ucut([ucut.xIndex]>=(minX));
    ucut=ucut([ucut.xIndex]<=(maxX));
    ucut=ucut([ucut.yIndex]>=(minY));
    ucut=ucut([ucut.yIndex]<=(maxY));

    save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\ucutModel.mat','ucutModel')
    save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\ucut.mat','ucut')
end
```

f4_buildPU

```
function f4_buildPU

    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\ucut.mat')
    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\params.mat')
    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\ucutModel.mat')

    xSize=params.sizeX;
    ySize=params.sizeY;
    minX=min([ucut.xIndex]);
    maxX=max([ucut.xIndex]);
    minY=min([ucut.yIndex]);
    maxY=max([ucut.yIndex]);
    bSize=params.bHeight;
    noBlocksX=ceil(xSize/params.bHeight);
    noBlocksY=ceil(ySize/params.bHeight);
    ucutLevel=max([ucutModel.zIndex]);
    [ucut.layout]=deal(0);

    noBB=(((maxX-minX)+1)-(noBlocksX-1))*(((maxY-minY)+1)-(noBlocksY-1));


bigBlocks(noBB)=struct('x1',[],'x2',[],'y1',[],'y2',[],'xCoord1',[],'xCoord
2',[],'yCoord1',[],'yCoord2',[],'bBlockIndex',[],'oreTon',0,'metalContent',
0,'minHeight',0,'maxHeight',0,'vertCumValue',0);

    count=1;
    for block=1:length(ucut)
        if ((ucut(block).xIndex + noBlocksX)-1) > maxX ||
((ucut(block).yIndex + noBlocksY))-1 > maxY
            continue
        else

bigBlocks(count)=struct('x1',ucut(block).xIndex,'x2',ucut(block).xIndex+noB
locksX-1,'y1',ucut(block).yIndex,'y2',ucut(block).yIndex+noBlocksY-1,...
                'xCoord1',ucut(block).xCoord,'xCoord2',ucut(block).xCoord +
(noBlocksX*bSize),'yCoord1',ucut(block).yCoord,...
                'yCoord2',ucut(block).yCoord +
(noBlocksY*bSize),'bBlockIndex',count,'oreTon',0,'metalContent',0,'minHeigh
t',0,'maxHeight',0,'vertCumValue',0);
            count=count+1;
        end
    end
    [bigBlocks.columnMetalContent]=deal(0);
    [bigBlocks.columnHeight]=deal(0);

    for b=1:length(bigBlocks)
        if mod(b,200)==0
            fprintf('Finished calculating big block %d\n',b)
        end
        bBlock=bigBlocks(b);
```

```
        columns=ucutModel([ucutModel.xIndex]>=bBlock.x1 &
[ucutModel.xIndex]<=bBlock.x2 & [ucutModel.yIndex]>=bBlock.y1 &
[ucutModel.yIndex]<=bBlock.y2);
        columnHeight=0;
        level=ucutLevel;
        columnMetalContent=0;
        flag=0;
        while columnHeight+(flag*params.bHeight)<=params.maxHeight
            metalContent=0;
            columnTon=0;
            wasteTon=0;
            totTon=0;
            flag=0;
            while columnTon <=params.minDR
                blocks=columns([columns.zIndex]==level);
                if columnTon+sum([blocks.Eton])<=params.minDR
                    columnTon=columnTon+sum([blocks.Eton]);
                    valid=blocks([blocks.Egrade]>0);
                    waste=blocks([blocks.Egrade]==0);
                    wasteTon=wasteTon+sum([waste.Eton]);
                    totTon=totTon+sum([blocks.Eton]);

      metalContent=metalContent+sum([valid.Eton].*([valid.Egrade]/100));
                    flag=flag+1;
                    level=level-1;
                else
                    break
                end
            end
            if metalContent==0 || (wasteTon/totTon)*100 >= params.dilution
                break
            elseif columnHeight+(flag*params.bHeight)<=params.maxHeight
                columnHeight=columnHeight+(flag*params.bHeight);
                columnMetalContent=columnMetalContent+metalContent;
            end
        end
        if bigBlocks(b).yCoord1 > 100 & bigBlocks(b).xCoord1 >=180
            bigBlocks(b).columnMetalContent=columnMetalContent;
            bigBlocks(b).columnHeight=columnHeight;
            layoutBlocks=columns([columns.zIndex]==ucutLevel);
        end
        if bigBlocks(b).columnHeight > params.minHeight
            for i=1:length(layoutBlocks)

ucut([ucut.blockIndex]==layoutBlocks(i).blockIndex).layout=1;
            end
        end


    end
save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\bigBlocks.mat','bigBlocks')
    save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\ucut.mat','ucut')
end
```

f5_objFunctionCoeffAgg

```
function f5_objFunctionCoeffAgg

    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\bigBlocks.mat')
    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\params.mat')
    load('Elevation Optimization\Drawpoints COnfiguration
Opt\MatFiles\ucut.mat')

    noBB=length(bigBlocks);

    f2MetalContent=zeros(1,noBB);

    for b=1:noBB
        if bigBlocks(b).columnHeight >= params.minHeight
            f2MetalContent(b)=-bigBlocks(b).columnMetalContent;
        end
    end

    save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\f2MetalContent.mat','f2MetalContent')

end
```

f6_constAgg

```
function f6_constAgg

load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\bigBlocks.mat')
load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\ucut.mat')
load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\params.mat')

nBlocks=length(bigBlocks);
ore=ucut([ucut.layout]==1);
ore=ore([ore.xCoord]>=160);
ore=ore([ore.yCoord]>100);
nOre=length(ore);
noX=params.sizeX/params.bHeight;
noY=params.sizeY/params.bHeight;

c2_continuity=sparse(nOre,nBlocks);

for b=1:nOre
    x=ore(b).xIndex;
    y=ore(b).yIndex;
    bBlocks=bigBlocks([bigBlocks.x1]<=x);
    bBlocks=bBlocks([bBlocks.x1]>=(x-noX+1));
    bBlocks=bBlocks([bBlocks.y1]>=y);
    bBlocks=bBlocks([bBlocks.y1]<=(y+noY-1));
    for j=1:length(bBlocks)
            c2_continuity(b,bBlocks(j).bBlockIndex)=1;
    end
end

c2_rhs_continuity=ones(nOre,1);

save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\c2_continuity.mat','c2_continuity')
save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\c2_rhs_continuity.mat','c2_rhs_continuity')
end
```

f7_solveAgg

```
function f7_solveAgg

    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\Aeq.mat')
    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\beq.mat')
    load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\f2MetalContent.mat')

    Cplex.Param.mip.tolerances.mipgap=0.01;

    solution=cplexbilp(f2MetalContent,[],[],Aeq,beq,options);

    save('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\solution.mat','solution')
end
```

f7_Main_Agg

```
fprintf('Reading Data and Preparing Objects...\n')
f1_readPar ('Elevation Optimization\Drawpoints Configuration
Opt\Parameters\Parameters_Saha.xlsx')
fprintf('Finished Reading Parameters File\n')
f2_readSimulations
f3_ucutModel(level)
fprintf('FInished Reading Data and Preparing Objets\n\n')
fprintf('Building PU...\n')
f4_buildPU
fprintf('Building Optimization Problem...\n')
f5_objFunctionCoeffAgg
f6_constAgg
fprintf('Solving Optimization Problem...\n')
f7_solveAgg
fprintf('\nPost Processing Solution...\n')

end
```

f1a_readPar

```
function f1a_readPar(flRoot,flRoot2)

    parameters=xlsread(flRoot);
    miningCap=xlsread(flRoot2);

params=struct('price',parameters(1),'sCost',parameters(2),'mCost',parameter
s(3),...

'pCost',parameters(4),'rec',parameters(5),'bHeight',parameters(6),'disRate'
,parameters(7),...

'maxAdjUnits',parameters(8),'minHeight',parameters(9),'maxHeight',parameter
s(10),...

'minMinCap',parameters(11),'maxMinCap',parameters(12),'minDR',parameters(13
),...

'maxDR',parameters(14),'T',parameters(15),'xIndexCol',parameters(16),'yInde
xCol',parameters(17),...

'zIndexCol',parameters(18),'xCoordCol',parameters(19),'yCoordCol',parameter
s(20),...

'zCoordCol',parameters(21),'gradeCol',parameters(22),'tonCol',parameters(23
),...

'uDelCost',parameters(24),'uDelRate',parameters(25),'ped',parameters(26),'x
Start',parameters(27),...

'yStart',parameters(28),'xEnd1',parameters(29),'yEnd1',parameters(30),'xEnd
2',parameters(31),...

'yEnd2',parameters(32),'Vangle',parameters(33),'sizeX',parameters(34),'size
Y',parameters(35),...

'bSize',parameters(36),'S',parameters(37),'gRate',parameters(38),...

'tonOverCost',parameters(39),'tonUnderCost',parameters(40),'gradeOverCost',
parameters(41),...

'gradeUnderCost',parameters(42),'bSizeX',parameters(43),'bSizeY',parameters
(44),'HD',parameters(45),'VSA',parameters(46),...
        'HIZ',parameters(47));


     save('Elevation Optimization\MatFiles\params.mat','params')
     save('Elevation Optimization\MatFiles\miningCap.mat','miningCap')


end
```

f8_buildMU

```
function f8_buildMU

load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\bigBlocks.mat')
load('Elevation Optimization\Drawpoints Configuration
Opt\MatFiles\ucutModel.mat')
load('Elevation Optimization\MatFiles\params.mat')
sizeX=params.sizeX;
sizeY=params.sizeY;
ucutLevel=max([ucutModel.zIndex]);
ucutElevation=max([ucutModel.zCoord]);
limit=ucutElevation+params.maxHeight;
diffLevel=ceil((limit-ucutElevation)/params.bHeight);
minLevel=ucutLevel-diffLevel;
[ucutModel(:).columnIndex]=deal(0);
[ucutModel(:).miningUnitIndex]=deal(0);
[ucutModel(:).miningUnitTon]=deal(0);
[ucutModel(:).miningUnitGrade]=deal(0);
[ucutModel(:).miningUnitHeight]=deal(0);

noBlocks=(ceil(sizeX/params.bHeight))*(ceil(sizeY/params.bHeight));

ton=mean([ucutModel([ucutModel.Egrade]>0).Eton]);
height=ceil(params.minDR*1000/(ton*noBlocks));

nBBlocks=length(find([bigBlocks.solutionConstrained]));
bigBlocks=bigBlocks([bigBlocks.solutionConstrained]==1);
cCounter=0;

for b=1:length(bigBlocks)
        cCounter=cCounter+1;
        mCounter=0;
        bBlock=bigBlocks(b);
        columns=ucutModel([ucutModel.xIndex]>=bBlock.x1 &
[ucutModel.xIndex]<=bBlock.x2 & [ucutModel.yIndex]>=bBlock.y1 &
[ucutModel.yIndex]<=bBlock.y2);
        columnHeight=0;
        level=ucutLevel;
        columnMetalContent=0;
        flag=0;
        miningUnitIndex=0;
        while columnHeight+(flag*params.bHeight)<=params.maxHeight
            metalContent=0;
            columnTon=0;
            levelBot=level;
            flag=0;
            miningUnitIndex=miningUnitIndex+1;
            while columnTon <=params.minDR
                blocks=columns([columns.zIndex]==level);
                if columnTon+sum([blocks.Eton])<=params.minDR
                    columnTon=columnTon+sum([blocks.Eton]);

metalContent=metalContent+sum([blocks.Eton].*([blocks.Egrade]/100));
```

```
                    flag=flag+1;
                    level=level-1;
                else
                    break
                end
            end
            if metalContent==0
                break
            elseif columnHeight+(flag*params.bHeight)<=params.maxHeight
                mCounter=mCounter+1;
                columnHeight=columnHeight+(flag*params.bHeight);
                columnMetalContent=columnMetalContent+metalContent;
                columnGrade=(metalContent/columnTon)*100;

            end
            mBlocks=columns([columns.zIndex]>=level+1 &
[columns.zIndex]<=levelBot);
            for mB=1:length(mBlocks)

ucutModel(mBlocks(mB).ucutIndex).miningUnitIndex=miningUnitIndex;
                ucutModel(mBlocks(mB).ucutIndex).columnIndex=cCounter;
                ucutModel(mBlocks(mB).ucutIndex).miningUnitTon=columnTon;

ucutModel(mBlocks(mB).ucutIndex).miningUnitGrade=columnGrade;

ucutModel(mBlocks(mB).ucutIndex).miningUnitHeight=columnHeight;
            end
        end
end
save('Elevation Optimization\MatFiles\ucutModel.mat','ucutModel')
end
```

f9_updMU

```
function f9_updMU

load('Elevation Optimization\MatFiles\ucutModel.mat','ucutModel')
load('Elevation Optimization\MatFiles\params.mat','params')
units=ucutModel([ucutModel.columnIndex]>0);
S=params.S;
noColumns=max([ucutModel.columnIndex]);
count=0;

miningUnits(1)=struct('index',[],'x1',[],'x2',[],'y1',[],'y2',[],'xCoord1',
[],...

'xCoord2',[],'yCoord1',[],'yCoord2',[],'xMid',[],'yMid',[],'xCoordMid',[],.
..

'yCoordMid',[],'columnIndex',[],'miningUnitIndex',[],'zCoord',[],'miningUni
tETon',[],'miningUnitEGrade',[],...
            'miningUnitGrade',[],'miningUnitTon',[],'miningUnitHeight',[]);
for s = 1:S
    miningUnits(1).(strcat('miningUnitGrade',int2str(s)))=[];
    miningUnits(1).(strcat('miningUnitTon',int2str(s)))=[];
end
for column=1:noColumns

    columnUnits=units([units.columnIndex]==column);
    x1=min([columnUnits.xIndex]);
    y1=max([columnUnits.yIndex]);
    x2=max([columnUnits.xIndex]);
    y2=min([columnUnits.yIndex]);
    xCoord1=min([columnUnits.xCoord]);
    yCoord1=max([columnUnits.yCoord]);
    xCoord2=max([columnUnits.xCoord]);
    yCoord2=min([columnUnits.yCoord]);
    xMid=((x2-x1)/2)+x1;
    yMid=((y1-y2)/2)+y2;
    xCoordMid=(((xCoord2+params.bSize)-xCoord1)/2)+xCoord1;
    yCoordMid=((yCoord1-yCoord2)/2)+yCoord2;


    for unit=1:max([columnUnits.miningUnitIndex])
        count=count+1;
        mUnit = columnUnits([columnUnits.miningUnitIndex]==unit);
        Eton = sum([mUnit.Eton]);
        Egrade = sum([mUnit.Eton].*[mUnit.Egrade])/Eton;
        ton=sum([mUnit.ton]);
        grade=sum([mUnit.ton].*[mUnit.grade])/ton;
        miningUnits(count).miningUnitETon = Eton;
        miningUnits(count).miningUnitEGrade = Egrade;
        miningUnits(count).miningUnitTon=ton;
        miningUnits(count).miningUnitGrade=grade;
        miningUnits(count).miningUnitHeight=mUnit(1).miningUnitHeight;
        for s=1:S
            simTon=sum([mUnit.(strcat('ton',int2str(s)))]);
```

```
simGrade=sum([mUnit.(strcat('ton',int2str(s)))].*[mUnit.(strcat('grade',int
2str(s)))])/ton;
            miningUnits(count).index = count;
            miningUnits(count).x1 = x1;
            miningUnits(count).x2 = x2;
            miningUnits(count).y1 = y1;
            miningUnits(count).y2 = y2;
            miningUnits(count).xCoord1 = xCoord1;
            miningUnits(count).xCoord2 = xCoord2;
            miningUnits(count).yCoord1 = yCoord1;
            miningUnits(count).yCoord2 = yCoord2;
            miningUnits(count).xMid = xMid;
            miningUnits(count).yMid = yMid;
            miningUnits(count).xCoordMid = xCoordMid;
            miningUnits(count).yCoordMid = yCoordMid;
            miningUnits(count).columnIndex = column;
            miningUnits(count).miningUnitIndex = unit;
            miningUnits(count).zCoord = max([mUnit.zCoord]);
            miningUnits(count).(strcat('miningUnitTon',int2str(s))) =
simTon;
            miningUnits(count).(strcat('miningUnitGrade',int2str(s))) =
simGrade;
        end
    end
end
save('Elevation Optimization\MatFiles\miningUnits.mat','miningUnits')
end
```

f10_mixScenarios

```
function f_mixScenarios

    load('Elevation Optimization\MatFiles\params.mat')
    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\ucutModel.mat')

    columns=miningUnits([miningUnits.miningUnitIndex]==1);
    C=length(columns);
    U=length(miningUnits);
    [ucutModel.flag]=deal(0);
    zLevel=min([ucutModel.zCoord])-params.bHeight;

    for c=1:C
        columnModel=miningUnits([miningUnits.columnIndex]==c);
        U=length(columnModel);
        cX=columnModel(1).xCoordMid;
        cY=columnModel(1).yCoordMid;
        for u=1:U
            counter=0;
            if columnModel(u).miningUnitHeight <= params.HIZ
                continue
            else
                cZ=zLevel+columnModel(u-1).miningUnitHeight;
                h=params.HD*tand(params.VSA);
                x=[cX,cY,cZ];
                dir=[0,0,1];
                r=params.HD;
                blocksSpace=ucutModel([ucutModel.xCoord]>=cX-r-
(params.bSizeX/2) & [ucutModel.xCoord]<=cX+r-(params.bSizeX/2) &
[ucutModel.yCoord]>=cY-r+(params.bSizeY/2) &
[ucutModel.yCoord]<=cY+r+(params.bSizeY/2) &
[ucutModel.zCoord]>=cZ+(params.bHeight/2) &
[ucutModel.zCoord]<=cZ+h+(params.bHeight/2));
                blocksSpace=blocksSpace([blocksSpace.flag]==0);
                blockSet=ucutModel([ucutModel.columnIndex]==c &
[ucutModel.miningUnitIndex]==u);
                for u2=1:length(blocksSpace)

p=[blocksSpace(u2).xCoord+(params.bSizeX/2),blocksSpace(u2).yCoord-
(params.bSizeY/2),blocksSpace(u2).zCoord-(params.bHeight/2)];
                    cone_dist=dot(p-x,dir);
                    cone_radius=(cone_dist/h)*r;
                    orth_dist=length((p-x)-cone_dist*dir);
                    if orth_dist < cone_radius &&
~ismember(blocksSpace(u2).blockIndex,[blockSet.blockIndex])
                        blockSet(end+1)=blocksSpace(u2);
                    end
                end
                cumTon=0;
                f=fieldnames(blockSet)';
                f{2,1}={};
                blockSample=struct(f{:});
                for s=1:params.S
```

```
                    while
cumTon<=columnModel(u).(strcat('miningUnitTon',int2str(s)))
                        blockSet=blockSet([blockSet.flag]==0);
                        pos=randi(length(blockSet));
                        blockSample(end+1)=blockSet(pos);
                        blockSet(pos).flag=1;

cumTon=cumTon+blockSet(pos).(strcat('ton',int2str(s)));
                    end
                    mixTon=sum([blockSample.(strcat('ton',int2str(s)))]);

mixGrade=sum([blockSample.(strcat('grade',int2str(s)))].*[blockSample.(strc
at('ton',int2str(s)))])/mixTon;
                    miningUnits([miningUnits.columnIndex]==c &
[miningUnits.miningUnitIndex]==u).(strcat('miningUnitTon',int2str(s)))=mixT
on;
                    miningUnits([miningUnits.columnIndex]==c &
[miningUnits.miningUnitIndex]==u).(strcat('miningUnitGrade',int2str(s)))=mi
xGrade;
                end
            end
            counter=counter+1;
            if mod(counter,100)==0
                fprintf('\nFinished mixing MU %d',u)
            end
        end
    end
    save('Elevation Optimization\MatFiles\miningUnits.mat','miningUnits')
end
```

f11_objFunctionCoeffSch

```
function f11_objFunctionCoeffSch

    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\params.mat')
    T=params.T;
    S=params.S;
    f=zeros(1,(length(miningUnits)*params.T));
    dCost=164*params.uDelCost;
    devGrade=zeros(1,2*params.S*params.T);
    devTon=zeros(1,2*params.S*params.T);

    for t=1:params.T
        for unit=1:length(miningUnits)
            blockVals = zeros(1,params.S);
            for s=1:params.S

revenue=(miningUnits(unit).(strcat('miningUnitGrade',int2str(s)))/100) *
miningUnits(unit).(strcat('miningUnitTon',int2str(s))) * (params.rec/100) *
(params.price-params.sCost);

miningCost=miningUnits(unit).(strcat('miningUnitTon',int2str(s)))*(params.m
Cost);

processCost=miningUnits(unit).(strcat('miningUnitTon',int2str(s)))*(params.
pCost);
                if miningUnits(unit).miningUnitIndex == 1
                    miningCost=miningCost+dCost;
                end
                if revenue>processCost
                    profit=revenue-(miningCost+processCost);
                else
                    profit=-miningCost;
                end
                disProfit=profit/((1+(params.disRate/100))^t);
                blockVals(s)=disProfit;
            end
            f(unit+(length(miningUnits)*(t-1)))=mean(blockVals)*-1;
        end
    end
    for s=1:params.S
        for t=1:params.T
            devGrade(t+(params.T*(s-
1)))=(params.gradeOverCost/((1+(params.gRate/100))^t));
            devGrade((T*S)+t+(params.T*(s-
1)))=(params.gradeUnderCost/((1+(params.gRate/100))^t));
        end
    end
    for s=1:params.S
        for t=1:params.T
            devTon(t+(params.T*(s-
1)))=(params.tonOverCost/((1+(params.gRate/100))^t));
            devTon((T*S)+t+(params.T*(s-
1)))=(params.tonUnderCost/((1+(params.gRate/100))^t));
        end
```

```
    end
    f=[f,devTon,devGrade];

    save('Elevation Optimization\MatFiles\f.mat','f')
end
```

f12_constMiningTarget

```
function f12_constMiningTarget

    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\params.mat')
    load('Elevation Optimization\MatFiles\miningCap.mat')

    noUnits=length(miningUnits);
    T=params.T;
    S=params.S;

c1_targetCap=sparse(params.T*S,(noUnits*params.T)+(2*params.T*S)+(2*params.
T*S));

    for s=1:params.S
        for t=1:params.T
            for unit=1:noUnits
                c1_targetCap(t+(T*(s-1)),unit+(noUnits*(t-
1)))=miningUnits(unit).(strcat('miningUnitTon',int2str(s)));
            end
            c1_targetCap(t+(T*(s-1)),(T*noUnits)+t+(T*(s-1)))=-1;
            c1_targetCap(t+(T*(s-1)),(T*noUnits)+(T*S)+t+(T*(s-1)))=1;
        end
    end


    c1_rhs_targetCap=ones(t,1).*miningCap(2,:)'*1000;
    c1_rhs_targetCap=repmat(c1_rhs_targetCap,S,1);

    save('Elevation Optimization\MatFiles\c1_targetCap.mat','c1_targetCap')
    save('Elevation
Optimization\MatFiles\c1_rhs_targetCap.mat','c1_rhs_targetCap')

end
```

f13_constReserves

```
function f13_constReserves

    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\params.mat')

    noUnits=length(miningUnits);

    c2_reserves=sparse(noUnits,(params.T*noUnits));

    for unit=1:noUnits
        for t=1:params.T
            c2_reserves(unit,unit+(noUnits*(t-1)))=1;
        end
    end

    c2_rhs_reserves=ones(noUnits,1);

    save('Elevation Optimization\MatFiles\c2_reserves.mat','c2_reserves')
    save('Elevation
Optimization\MatFiles\c2_rhs_reserves.mat','c2_rhs_reserves')

end
```

f14_constVertPrec

```
function f14_constVertPrec

    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\params.mat')

    noColumns=max([miningUnits.columnIndex]);
    noUnits=length(miningUnits);

    c3_verticalPrecedence=sparse(noUnits*params.T,noUnits*params.T);

    for t=1:params.T
        for column=1:noColumns
            columnModel=miningUnits([miningUnits.columnIndex]==column);
            noUnitsCol=max([columnModel.miningUnitIndex]);
            for unit=1:noUnitsCol
                if unit==1
                    continue
                else

uIndex=columnModel([columnModel.miningUnitIndex]==unit).index;

precIndex=columnModel([columnModel.miningUnitIndex]==(unit-1)).index;
                    c3_verticalPrecedence(uIndex+(noUnits*(t-
1)),uIndex+(noUnits*(t-1)))=1;
                    if t>=2
                        c3_verticalPrecedence(uIndex+(noUnits*(t-
1)),precIndex+(noUnits*(t-1)))=-1;
                        c3_verticalPrecedence(uIndex+(noUnits*(t-
1)),precIndex+(noUnits*(t-2)))=-1;
                    end
                end
            end
        end
    end

    [r,c]=size(c3_verticalPrecedence);

    c3_rhs_verticalPrecedence=zeros(r,1);

    save('Elevation
Optimization\MatFiles\c3_verticalPrecedence','c3_verticalPrecedence')
    save('Elevation
Optimization\MatFiles\c3_rhs_verticalPrecedence','c3_rhs_verticalPrecedence
')

end
```

f15_constMaxDrawRate

```
function f15_constMaxDrawRate

    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\params.mat')

    noUnits=length(miningUnits);
    noColumns=max([miningUnits.columnIndex]);

    c4_maxDR=sparse(params.T*noColumns,noUnits*params.T);

    for t=1:params.T
        for column=1:noColumns
            columnModel=miningUnits([miningUnits.columnIndex]==column);
            noUnitsCol=max([columnModel.miningUnitIndex]);
            for unit=1:noUnitsCol

uIndex=columnModel([columnModel.miningUnitIndex]==unit).index;
                c4_maxDR(column + noColumns*(t-1),uIndex + noUnits*(t-1)) =
columnModel([columnModel.miningUnitIndex]==unit).miningUnitETon;
            end
        end
    end

    c4_rhs_maxDR=ones(params.T*noColumns,1)*params.maxDR;

    save('Elevation Optimization\MatFiles\c4_maxDR.mat','c4_maxDR')
    save('Elevation Optimization\MatFiles\c4_rhs_maxDR.mat','c4_rhs_maxDR')
end
```

f16_buildHorPrec

```
function f16_buildHorPrec

load('Elevation Optimization\MatFiles\miningUnits.mat','miningUnits')
load('Elevation Optimization\MatFiles\params.mat','params')

columns=miningUnits([miningUnits.miningUnitIndex]==1);
N=length(columns);
T=params.T;

Dir1=[xStart,yStart;xEnd1,yEnd1];
Dir2=[xStart,yStart;xEnd2,yEnd2];
VShapedAngle = Vangle;
StPoint1 = [Dir1(1,1), Dir1(1,2)];
EnPoint1 = [Dir1(2,1), Dir1(2,2)];
StPoint2 = [Dir2(1,1), Dir2(1,2)];
EnPoint2 = [Dir2(2,1), Dir2(2,2)];
StepDis = 1;
PlotStepSize = 5;
PlotCounter = 0;
Dir1LineSlop = (EnPoint1(2)-StPoint1(2))/(EnPoint1(1)-StPoint1(1));
Dir2LineSlop = (EnPoint2(2)-StPoint2(2))/(EnPoint2(1)-StPoint2(1));
Dir1LineLegnth = sqrt(((EnPoint1(1)-StPoint1(1))^2)+((EnPoint1(2)-
StPoint1(2))^2));
Dir2LineLegnth = sqrt(((EnPoint2(1)-StPoint2(1))^2)+((EnPoint2(2)-
StPoint2(2))^2));
PrecVShape = zeros(N,2);
PrecVShape (1:N,1) = 1:N;
PrecCounter = 1;
adjPrecedence=zeros(N);

while PrecCounter <= N
    for iloop = 1:N


        if StPoint1(1) == EnPoint1(1) %Vertical Direction (when X_Start ==
X_End)
            VShape1Point2 = [StPoint1(1) + StepDis*tand(VShapedAngle/2),
StPoint1(2)];
            VShape1Point3 = [StPoint1(1) - StepDis*tand(VShapedAngle/2),
StPoint1(2)];
            if StPoint1(2) < EnPoint1(2)
                VShape1Point1 = [StPoint1(1), StPoint1(2) + StepDis];
            else
                VShape1Point1 = [StPoint1(1), StPoint1(2)-StepDis];
            end
        elseif StPoint1(2) == EnPoint1(2) %Horizontal Direction (when
Y_Start == Y_End)
            VShape1Point2 = [StPoint1(1), StPoint1(2) +
StepDis*tand(VShapedAngle/2)];
            VShape1Point3 = [StPoint1(1), StPoint1(2) -
StepDis*tand(VShapedAngle/2)];
            if StPoint1(1) < EnPoint1(1)
                VShape1Point1 = [StPoint1(1) + StepDis, StPoint1(2)];
```

```
            else
                VShape1Point1 = [StPoint1(1) - StepDis, StPoint1(2)];
            end
        else

        XVpointDir1Point1 = (StepDis/sqrt(1+(Dir1LineSlop^2)))+StPoint1(1);
% X coordination of the point in whch the V shape and the Direction Line
intersect (first situation which means considering + sign for the square
root calculations of the coordinates)
        YVpointDir1Point1 =
Dir1LineSlop*(StepDis/sqrt(1+(Dir1LineSlop^2)))+StPoint1(2); % Y
coordination of the point in whch the V shape and the Direction Line
intersect (first situation which means considering + sign for the square
root calculations of the coordinates)

        XVpointDir2Point1 = (StepDis/sqrt(1+(Dir2LineSlop^2)))+StPoint2(1);
% X coordination of the point in whch the V shape and the Direction Line
intersect (first situation which means considering + sign for the square
root calculations of the coordinates)
        YVpointDir2Point1 =
Dir2LineSlop*(StepDis/sqrt(1+(Dir2LineSlop^2)))+StPoint1(2); % Y
coordination of the point in whch the V shape and the Direction Line
intersect (first situation which means considering + sign for the square
root calculations of the coordinates)

        XVpointDir1Point2 = -
(StepDis/sqrt(1+(Dir1LineSlop^2)))+StPoint1(1); % X coordination of the
point in whch the V shape and the Direction Line intersect (second
situation which means considering - sign for the square root calculations
of the coordinates)
        YVpointDir1Point2 = -
Dir1LineSlop*(StepDis/sqrt(1+(Dir1LineSlop^2)))+StPoint1(2); % Y
coordination of the point in whch the V shape and the Direction Line
intersect (Second situation which means considering - sign for the square
root calculations of the coordinates)

        XVpointDir2Point2 = -
(StepDis/sqrt(1+(Dir2LineSlop^2)))+StPoint2(1); % X coordination of the
point in whch the V shape and the Direction Line intersect (second
situation which means considering - sign for the square root calculations
of the coordinates)
        YVpointDir2Point2 = -
Dir2LineSlop*(StepDis/sqrt(1+(Dir2LineSlop^2)))+StPoint2(2); % Y
coordination of the point in whch the V shape and the Direction Line
intersect (Second situation which means considering - sign for the square
root calculations of the coordinates)

        XVpointPerLine1Point1 =
((StepDis*tand(VShapedAngle/2))/(sqrt(1+(1/(Dir1LineSlop^2)))))+StPoint1(1)
; % X coordination of the point in whch the V shape and the Perpendicular
line (to the Direction Line) intersect (first point)
        YVpointPerLine1Point1 = (-
1/Dir1LineSlop)*(((StepDis*tand(VShapedAngle/2))/(sqrt(1+(1/(Dir1LineSlop^2
))))))+StPoint1(2); % Y coordination of the point in whch the V shape and
the Perpendicular line (to the Direction Line) intersect (first point)
```

```
        XVpointPerLine2Point1 =
((StepDis*tand(VShapedAngle/2))/(sqrt(1+(1/(Dir2LineSlop^2)))))+StPoint2(1)
; % X coordination of the point in whch the V shape and the Perpendicular
line (to the Direction Line) intersect (first point)
        YVpointPerLine2Point1 = (-
1/Dir2LineSlop)*(((StepDis*tand(VShapedAngle/2))/(sqrt(1+(1/(Dir2LineSlop^2
))))))+StPoint2(2); % Y coordination of the point in whch the V shape and
the Perpendicular line (to the Direction Line) intersect (first point)

        XVpointPerLine1Point2 = -
((StepDis*tand(VShapedAngle/2))/(sqrt(1+(1/(Dir1LineSlop^2)))))+StPoint1(1)
; % X coordination of the point in whch the V shape and the Perpendicular
line (to the Direction Line) intersect (second point)
        YVpointPerLine1Point2 = (-1/Dir1LineSlop)*(-
((StepDis*tand(VShapedAngle/2))/(sqrt(1+(1/(Dir1LineSlop^2))))))+StPoint1(2
); % Y coordination of the point in whch the V shape and the Perpendicular
line (to the Direction Line) intersect (second point)

        XVpointPerLine2Point2 = -
((StepDis*tand(VShapedAngle/2))/(sqrt(1+(1/(Dir2LineSlop^2)))))+StPoint2(1)
; % X coordination of the point in whch the V shape and the Perpendicular
line (to the Direction Line) intersect (second point)
        YVpointPerLine2Point2 = (-1/Dir2LineSlop)*(-
((StepDis*tand(VShapedAngle/2))/(sqrt(1+(1/(Dir2LineSlop^2))))))+StPoint2(2
); % Y coordination of the point in whch the V shape and the Perpendicular
line (to the Direction Line) intersect (second point)

        %Distance Calculator
        Dis1VpointToVpoint1 = sqrt(((EnPoint1(1)-
XVpointDir1Point1)^2)+((EnPoint1(2)-YVpointDir1Point1)^2)); %Calculating
the distance between the point of intersection of V shape (and Direction
Line) and end point in first situation
        Dis1VpointToVpoint2 = sqrt(((EnPoint1(1)-
XVpointDir1Point2)^2)+((EnPoint1(2)-YVpointDir1Point2)^2)); %Calculating
the distance between the point of intersection of V shape (and Direction
Line) and end point in second situation

        Dis2VpointToVpoint1 = sqrt(((EnPoint1(1)-
XVpointDir2Point1)^2)+((EnPoint2(2)-YVpointDir2Point1)^2)); %Calculating
the distance between the point of intersection of V shape (and Direction
Line) and end point in first situation
        Dis2VpointToVpoint2 = sqrt(((EnPoint1(1)-
XVpointDir2Point2)^2)+((EnPoint2(2)-YVpointDir2Point2)^2)); %Calculating
the distance between the point of intersection of V shape (and Direction
Line) and end point in second situation

            if Dis1VpointToVpoint1 < Dis1VpointToVpoint2 %Comparing the two
distances to find the lower one and pick that as the correct point for the
triangle of the V shape
                VShape1Point1 = [XVpointDir1Point1,YVpointDir1Point1]; %if
dis1 < dis2 then dis1 is the correct point
            else
                VShape1Point1 = [XVpointDir1Point2,YVpointDir1Point2]; %if
dis1 > dis2 then dis2 is the correct point
            end
```

```
        VShape1Point2 = [XVpointPerLine1Point1,YVpointPerLine1Point1];
%Second point of the V shape (triangle) which is the first intersection of
the V Shape and the Perpendicular Line (Perpendicular to the Direction
Line)
        VShape1Point3 = [XVpointPerLine1Point2,YVpointPerLine1Point2];
%Third point of the V shape (triangle) which is the second intersection of
the V Shape and the Perpendicular Line (Perpendicular to the Direction
Line)
 %22222222222222222222
            if Dis2VpointToVpoint1 < Dis2VpointToVpoint2 %Comparing the two
distances to find the lower one and pick that as the correct point for the
triangle of the V shape
                VShape2Point1 = [XVpointDir2Point1,YVpointDir2Point1]; %if
dis1 < dis2 then dis1 is the correct point
            else
                VShape2Point1 = [XVpointDir2Point2,YVpointDir2Point2]; %if
dis1 > dis2 then dis2 is the correct point
            end
        VShape2Point2 = [XVpointPerLine2Point1,YVpointPerLine2Point1];
%Second point of the V shape (triangle) which is the first intersection of
the V Shape and the Perpendicular Line (Perpendicular to the Direction
Line)
        VShape2Point3 = [XVpointPerLine2Point2,YVpointPerLine2Point2];
%Third point of the V shape (triangle) which is the second intersection of
the V Shape and the Perpendicular Line (Perpendicular to the Direction
Line)
        %for plotting purposes to be used in PlotDPS_starting_Periods.m
function
        if StepDis > (Dir1LineLegnth/15) & StepDis < (Dir1LineLegnth/10)
            Triangle1 = [VShape1Point1;VShape1Point2;VShape1Point3];
        else
        end
        if StepDis > (Dir2LineLegnth/15) & StepDis < (Dir2LineLegnth/10)
            Triangle2 = [VShape2Point1;VShape2Point2;VShape2Point3];
        else
        end
%-----------------------------
        end
        xv = [VShape1Point1(1,1), VShape1Point2(1,1), VShape1Point3(1,1),
VShape2Point1(1,1), VShape2Point2(1,1),
VShape2Point3(1,1),VShape1Point1(1,1), VShape1Point3(1,1)];
        yv = [VShape1Point1(1,2), VShape1Point2(1,2), VShape1Point3(1,2),
VShape2Point1(1,2), VShape2Point2(1,2),
VShape2Point3(1,2),VShape1Point1(1,2), VShape1Point3(1,2)];
        xvall(StepDis,:) = xv;
        yvall(StepDis,:) = yv;
        [in,on] =
inpolygon(columns(iloop).xCoordMid,columns(iloop).yCoordMid,xv,yv);

        %Check if the drawpoint falls into the V shape (triangle) or not
        if (in==1 | on==1) && (PrecVShape(iloop, 2) == 0)
            if iloop==140
            disp('Hi')
            end
            PrecVShape(iloop, 2) = PrecCounter;
            PrecCounter = PrecCounter + 1;
            xcCenter=columns(iloop).xCoordMid;
```

```
            ycCenter=columns(iloop).yCoordMid;
            radius=max(params.sizeX,params.sizeY);
            theta=0:0.01:2*pi;
            xCircle=radius*cos(theta)+xcCenter;
            yCircle=radius*sin(theta)+ycCenter;
            for dp=1:N
                if dp==iloop
                    continue
                else
                if
inpolygon(columns(dp).xCoordMid,columns(dp).yCoordMid,xv,yv)&&inpolygon(col
umns(dp).xCoordMid,columns(dp).yCoordMid,xCircle,yCircle)
                    adjPrecedence(iloop,dp)=1;
                end
                end
            end
       else
        end
    end

    StepDis = StepDis + 1;

    if StepDis == PlotStepSize && PlotCounter <= 10;
            PlotCounter = PlotCounter+1;
            plot(xv,yv);
            hold on
            PlotStepSize = PlotStepSize + 40;
            MiningDirectionPlotDataX(PlotCounter,:) = xv;
            MiningDirectionPlotDataY(PlotCounter,:) = yv;
            else
    end
end
for dp = 1:N
    DpPrec = PrecVShape(dp,2);
    counter = 1; %Counting the predecessors of "dp"

    for AdjCount = 1:N %Loop for the Adjacent drawpoints of "dp"....for dp
1 is 3
        AdjPrec = PrecVShape(AdjCount,2);

        if (DpPrec == 1)
           Prec(dp,1) = dp;
        elseif (AdjPrec + 1 == DpPrec)
           Prec(dp,1) = AdjCount;
            counter = counter+1;

        else

        end
    end

end

[lastvals,idx]=sort(PrecVShape(:,end));
sortedPrec=PrecVShape(idx,:);
```

```
horPrecedence=zeros(N);

for c=2:N
    horPrecedence(sortedPrec(c,1),sortedPrec(c-1,1))=1;
end


save('Elevation Optimization\MatFiles\horPrecedence.mat','horPrecedence');
save('Elevation Optimization\MatFiles\adjPrecedence.mat','adjPrecedence');
end
```

f17_constHorPrec

```
function f17_constHorPrec

load('Elevation Optimization\MatFiles\miningUnits.mat')
load('Elevation Optimization\MatFiles\adjPrecedence.mat')
load('Elevation Optimization\MatFiles\params.mat')

ucutUnits=miningUnits([miningUnits.miningUnitIndex]==1);
noUnits=length(miningUnits);
noColumns=length(ucutUnits);

c5_horPrecedence=sparse(noUnits*params.T,noUnits*params.T);

for t=1:params.T
    for column=1:noColumns
            prec=find(adjPrecedence(column,:));
            if prec
                uIndex=ucutUnits(column).index;
                c5_horPrecedence(uIndex+(noUnits*(t-1)),uIndex+(noUnits*(t-
1)))=sum(adjPrecedence(column,:));
                for t2=1:t
                    for i = 1:length(prec)
                        precIndex=ucutUnits(prec(i)).index;
                        c5_horPrecedence(uIndex+(noUnits*(t-
1)),precIndex+(noUnits*(t2-1)))=-1;
                    end
                end
            end
    end
end
```

f18_constUndercutRate

```
function f18_constUndercutRate

load('Elevation Optimization\MatFiles\params.mat')
load('Elevation Optimization\MatFiles\miningUnits.mat')

base=miningUnits([miningUnits.miningUnitIndex]==1);
c6_uCutDelRate=sparse(params.T,length(miningUnits)*params.T);

area=((base(1).x2 - base(1).x1 + 1)*params.bHeight)*((base(1).y1 -
base(1).y2 + 1)*params.bHeight);

for t=1:params.T
    for unit=1:length(base)
        c6_uCutDelRate(t,base(unit).index + (length(miningUnits)*(t-
1)))=area;
    end
end

c6_rhs_uCutDelRate=ones(params.T,1)*params.uDelRate;

save('Elevation Optimization\MatFiles\c6_uCutDelRate.mat','c6_uCutDelRate')
save('Elevation
Optimization\MatFiles\c6_rhs_uCutDelRate.mat','c6_rhs_uCutDelRate')

end
```

f19_constDrawControl

```
function f19_constDrawControl

load('Elevation Optimization\MatFiles\miningUnits.mat')
load('Elevation Optimization\MatFiles\params.mat','params')
load('Elevation
Optimization\MatFiles\adjPrecedenceSlope.mat','adjPrecedenceSlope')

maxAdjUnits=params.maxAdjUnits;
ucutUnits=miningUnits([miningUnits.miningUnitIndex]==1);
noColumns=max([miningUnits.columnIndex]);
noUnits=length(miningUnits);
noEq=length(find(adjPrecedenceSlope(:,:)));

c7_drawControlUB=sparse(noEq*params.T,noUnits*params.T);
c7_drawControlLB=sparse(noEq*params.T,noUnits*params.T);

distances=zeros(noColumns,1);
for c=1:noColumns
    distances(c,1)=sqrt((ucutUnits(c).xCoordMid-params.xStart)^2 +
(ucutUnits(c).yCoordMid-params.yStart)^2);
end

[minD,minDindex]=min(distances);

start=ucutUnits(minDindex).columnIndex;

for t=1:params.T
    count=1;
    for column=1:noColumns
        if column==start
            continue
        else
            noPrec=length(find(adjPrecedenceSlope(column,:)));
            columnPrec=find(adjPrecedenceSlope(column,:));

            for eq=1:noPrec

columnModel2=miningUnits([miningUnits.columnIndex]==column);
                precColumn=columnPrec(eq);
                noUnitsCol2=max([columnModel2.miningUnitIndex]);

columnModel1=miningUnits([miningUnits.columnIndex]==precColumn);
                noUnitsCol1=max([columnModel1.miningUnitIndex]);
                for unit=1:noUnitsCol1

uIndex=columnModel1([columnModel1.miningUnitIndex]==unit).index;
                    c7_drawControlUB(count+(noEq*(t-1)),uIndex+(noUnits*(t-
1)))=1;
                    c7_drawControlLB(count+(noEq*(t-1)),uIndex+(noUnits*(t-
1)))=-1;
                end
                for unit=1:noUnitsCol2
```

```
uIndex=columnModel2([columnModel2.miningUnitIndex]==unit).index;
                    c7_drawControlUB(count+(noEq*(t-1)),uIndex+(noUnits*(t-
1)))=-1;
                    c7_drawControlLB(count+(noEq*(t-1)),uIndex+(noUnits*(t-
1)))=1;
                end
                for t2=1:t
                    for unit=1:noUnitsCol1

uIndex=columnModel1([columnModel1.miningUnitIndex]==unit).index;
                        c7_drawControlUB(count+(noEq*(t-
1)),uIndex+(noUnits*(t2-1)))=1;
                        c7_drawControlLB(count+(noEq*(t-
1)),uIndex+(noUnits*(t2-1)))=-1;
                    end
                    for unit=1:noUnitsCol2

uIndex=columnModel2([columnModel2.miningUnitIndex]==unit).index;
                        c7_drawControlUB(count+(noEq*(t-
1)),uIndex+(noUnits*(t2-1)))=-1;
                        c7_drawControlLB(count+(noEq*(t-
1)),uIndex+(noUnits*(t2-1)))=1;
                    end
                end


                count=count+1;
            end
        end
    end
end

c7_rhs_drawControlUB=ones(noEq*params.T,1)*maxAdjUnits;
c7_rhs_drawControlLB=ones(noEq*params.T,1)*maxAdjUnits;

save('Elevation
Optimization\MatFiles\c7_drawControlUB.mat','c7_drawControlUB')
save('Elevation
Optimization\MatFiles\c7_drawControlLB.mat','c7_drawControlLB')
save('Elevation
Optimization\MatFiles\c7_rhs_drawControlUB.mat','c7_rhs_drawControlUB')
save('Elevation
Optimization\MatFiles\c7_rhs_drawControlLB.mat','c7_rhs_drawControlLB')

end
```

f20_constGradeBounds

```
function f20_constGradeBounds

    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\params.mat')
    load('Elevation Optimization\MatFiles\miningCap.mat')


    U=length(miningUnits);
    T=params.T;
    S=params.S;
    C=max([miningUnits.columnIndex]);

    c8_gradeQUB=sparse(T*S,(U*T)+(4*T*S));
    c8_gradeQLB=sparse(T*S,(U*T)+(4*T*S));

    for s=1:S
        for t=1:T
            for u=1:U
                c8_gradeQUB(t+(T*(s-1)),u+(U*(t-
1)))=((miningUnits(u).(strcat('miningUnitGrade',int2str(s)))-
miningCap(4,t))/100)*miningUnits(u).(strcat('miningUnitTon',int2str(s)));
                c8_gradeQLB(t+(T*(s-1)),u+(U*(t-1)))=((-
miningUnits(u).(strcat('miningUnitGrade',int2str(s)))+miningCap(5,t))/100)*
miningUnits(u).(strcat('miningUnitTon',int2str(s)));
            end
            c8_gradeQUB(t+(T*(s-1)),(U*T)+(2*T*S)+t+(T*(s-1)))=-1;
            c8_gradeQLB(t+(T*(s-1)),(U*T)+(2*T*S)+(T*S)+t+(T*(s-1)))=-1;
        end
    end

    c8_rhs_gradeQUB=zeros(T*S,1);
    c8_rhs_gradeQLB=zeros(T*S,1);

    save('Elevation Optimization\MatFiles\c8_gradeQUB.mat','c8_gradeQUB')
    save('Elevation
Optimization\MatFiles\c8_rhs_gradeQUB.mat','c8_rhs_gradeQUB')
    save('Elevation Optimization\MatFiles\c8_gradeQLB.mat','c8_gradeQLB')
    save('Elevation
Optimization\MatFiles\c8_rhs_gradeQLB.mat','c8_rhs_gradeQLB')

end
```

f21_earlyStart

```
function f21_earlyStart

    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\ucutModel.mat')
    load('Elevation Optimization\MatFiles\params.mat')
    load('Elevation Optimization\MatFiles\f.mat')
    load('Elevation Optimization\MatFiles\miningCap.mat')
    load('Elevation Optimization\MatFiles\adjPrecedence.mat')

    U = length(miningUnits);
    C = max([miningUnits.columnIndex]);
    T = params.T;

    [ucutModel.earlyColumnStart]=deal(0);
    [miningUnits.earlyColumnStart]=deal(1);
    [miningUnits.earlyUnitStart]=deal(1);

    columns = miningUnits([miningUnits.miningUnitIndex]==1);

    seq = cell(length(columns));


noBlocks=(ceil(params.sizeX/params.bHeight))*(ceil(params.sizeY/params.bHei
ght));
    ton=mean([ucutModel([ucutModel.grade]>0).ton]);
    unitHeight=(ceil(params.minDR*1000/(ton*noBlocks)))*params.bHeight;

    for c=1:C
        temp = find(adjPrecedence(c,:));
        seq{c,1} = temp;
        count=1;
        while ~isempty(temp)
            temp = [];
            for i = 1:length(seq{c,count})
                seqAdj = find(adjPrecedence(seq{c,count}(i),:));
                tempAdj = [];
                    for j = 1:length(seqAdj)
                        if ismember(seqAdj(j),cell2mat(seq(c,:)))
                            continue
                        else
                            tempAdj = [tempAdj seqAdj(j)];
                        end
                    end
                    temp = [temp tempAdj];
            end
            temp = unique(temp);
            count = count+1;
            seq{c,count}=temp;
        end
    end

    for c=1:C
        num = length(find(~cellfun('isempty',seq(c,:))));
```

```
        colSeq = seq(c,1:num);
        height = zeros(1,num);
        seqTon = zeros(1,num);
        for i = 1:num
            if i*params.maxAdjUnits*unitHeight > params.maxHeight
                height(i) = params.maxHeight;
            else
                height(i) = i*params.maxAdjUnits;
            end
        end
        for i = 1:num
            ton = 0;
            for n = 1:length(colSeq{i})
                colTon =
sum([miningUnits([miningUnits.columnIndex]==colSeq{i}(n)).miningUnitTon]);
                ton = ton +
colTon*(height(i)/(length(miningUnits([miningUnits.columnIndex]==colSeq{i}(
n)))*unitHeight));
            end
            seqTon(i) = ton;
        end
        totalTon = sum(seqTon);
        prodTon = 0;
        p = 1;
        while (prodTon+(miningCap(3,p)*1000))<=totalTon && p<=T-1
            prodTon = prodTon + (miningCap(3,p)*1000);
            p = p+1;
        end
        columnModel = miningUnits([miningUnits.columnIndex]==c);
        for u = 1:length(columnModel)
            if p == 1
                miningUnits(columnModel(u).index).earlyColumnStart = p;

            else
                miningUnits(columnModel(u).index).earlyColumnStart = p-1;
            end
        end

        columnModelUnits = miningUnits([miningUnits.columnIndex]==c);

        ucutColumnModel = ucutModel([ucutModel.columnIndex]==c);
        for j = 1:length(ucutColumnModel)
            ucutModel(ucutColumnModel(j).ucutIndex).earlyColumnStart =
columnModelUnits(1).earlyColumnStart;
        end

        columnModelUnits = miningUnits([miningUnits.columnIndex]==c);

        dRate = 0;
        earlyStart = columnModelUnits(1).earlyColumnStart;
        dCount = 0;
        columnTon = [columnModelUnits.miningUnitTon];
        columnCumTon = cumsum(columnTon);
        cTon = sum(columnTon);
        miningUnits(columnModelUnits(1).index).earlyUnitStart = earlyStart;
```

```
        while dRate < cTon
            if (dRate+(params.maxDR*1000)) >= cTon
                dRate = cTon;
            else
                dRate = dRate + (params.maxDR*1000);
            end
            for i = 2:length(columnCumTon)
                if dRate <= columnCumTon(i)
                    unitEarlyStart = earlyStart + (i-2);
                    miningUnits(columnModelUnits(i).index).earlyUnitStart =
unitEarlyStart;
                    break
                end
            end
        end
    end
    save('Elevation Optimization\MatFiles\miningUnits.mat','miningUnits')
    save('Elevation Optimization\MatFiles\ucutModel.mat','ucutModel')
end
```

f22_concatenateConst

```
function f22_concatenateConst

    load('Elevation Optimization\MatFiles\c1_targetCap.mat')
    load('Elevation Optimization\MatFiles\c2_reserves.mat')
    load('Elevation Optimization\MatFiles\c1_rhs_targetCap.mat')
    load('Elevation Optimization\MatFiles\c2_rhs_reserves.mat')
    load('Elevation Optimization\MatFiles\c3_verticalPrecedence.mat')
    load('Elevation Optimization\MatFiles\c3_rhs_verticalPrecedence.mat')
    load('Elevation Optimization\MatFiles\c4_maxDR.mat')
    load('Elevation Optimization\MatFiles\c4_rhs_maxDR.mat')
    load('Elevation Optimization\MatFiles\c5_horPrecedence.mat')
    load('Elevation Optimization\MatFiles\c5_rhs_horPrecedence.mat')
    load('Elevation Optimization\MatFiles\c6_uCutDelRate.mat')
    load('Elevation Optimization\MatFiles\c6_rhs_uCutDelRate.mat')
    load('Elevation Optimization\MatFiles\c7_drawControlUB.mat')
    load('Elevation Optimization\MatFiles\c7_rhs_drawControlUB.mat')
    load('Elevation Optimization\MatFiles\c7_drawControlLB.mat')
    load('Elevation Optimization\MatFiles\c7_rhs_drawControlLB.mat')
    load('Elevation Optimization\MatFiles\params.mat')
    load('Elevation Optimization\MatFiles\c8_gradeQUB.mat','c8_gradeQUB')
    load('Elevation
Optimization\MatFiles\c8_rhs_gradeQUB.mat','c8_rhs_gradeQUB')
    load('Elevation Optimization\MatFiles\c8_gradeQLB.mat','c8_gradeQLB')
    load('Elevation
Optimization\MatFiles\c8_rhs_gradeQLB.mat','c8_rhs_gradeQLB')

Aineq=[c2_reserves;c3_verticalPrecedence;c4_maxDR;c5_horPrecedence;c6_uCutD
elRate;c7_drawControlUB;c7_drawControlLB];
    [r,c]=size(Aineq);
    Aineq=[Aineq,zeros(r,4*params.S*params.T)];
    Aineq=[Aineq;c8_gradeQUB;c8_gradeQLB];

bineq=[c2_rhs_reserves;c3_rhs_verticalPrecedence;c4_rhs_maxDR;c5_rhs_horPre
cedence;c6_rhs_uCutDelRate;c7_rhs_drawControlUB;c7_rhs_drawControlLB;c8_rhs
_gradeQUB;c8_rhs_gradeQLB];
    Aeq=[c1_targetCap];
    beq=[c1_rhs_targetCap];
    save('Elevation Optimization\MatFiles\Aineq.mat','Aineq')
    save('Elevation Optimization\MatFiles\bineq.mat','bineq')
    save('Elevation Optimization\MatFiles\Aeq.mat','Aeq')
    save('Elevation Optimization\MatFiles\beq.mat','beq')

end
```

f23_STWH

```
function solution = f23_STWH(n)

    load('Elevation Optimization\MatFiles\f.mat')
    load('Elevation Optimization\MatFiles\Aineq.mat')
    load('Elevation Optimization\MatFiles\bineq.mat')
    load('Elevation Optimization\MatFiles\miningUnits.mat')
    load('Elevation Optimization\MatFiles\params.mat')
    load('Elevation Optimization\MatFiles\Aeq.mat')
    load('Elevation Optimization\MatFiles\beq.mat')

    U = length(miningUnits);
    C = max([miningUnits.columnIndex]);
    T = params.T;
    S=params.S;
    lb=zeros(length(f),1);
    ub=[];
    sostype=[];
    sosind=[];
    soswt=[];
    Cplex.Param.mip.tolerance.mipgap=0.05;
    solTimes = zeros(1,T);
    periodNPV = zeros(1,T);
    for t = 1:T
        ctype1 = repmat('B',U*T,1)';
        ctype2 = repmat('C',(2*T*S)+(2*T*S),1)';
        ctype = [ctype1 ctype2];
        fprintf('\nSetting variables for Period %d at binary\n',t)
        if T - n - t + 1 > 0

            ctype(U*(t-1)+(U*n)+1 : end) = 'C';

        end

        fprintf('\nFinding Solution for Period %d\n',t)
Cplex.Param.mip.tolerance.mipgap=0.05;

        solTime = tic;
        solution =
cplexmilp(f,Aineq,bineq,Aeq,beq,sostype,sosind,soswt,lb,ub,ctype,options);
        solTimes(t) = toc(solTime);
        fprintf('\nSolution for Period %d found after %f
seconds\n',t,solTimes(t))

        npv = round(sum(f.*solution')/1000000,3);
        periodNPV(t) = npv;
        fprintf('\nNPV for solution at Period %d is %f M$\n',t,npv)

        fprintf('\nSolution for Period %d found\n',t)

        solString = ['Elevation Optimization\MatFiles\Solutions STWH by
Period\solutionP',num2str(t),'.mat'];
```

```
        save(solString,'solution')
        if t < T
            fprintf('\nFixing solution for period %d into next
optimization\n',t)

            for u = U*(t-1)+1 : (U*t)
                r = size(Aeq,1);
                r2 = size(beq,1);
                Aeq(r+1,u) = 1;
                beq(r2+1,1) = solution(u);
            end


            AeqString=['Elevation Optimization\MatFiles\Solutions STWH by
Period\AeqP',num2str(t),'.mat'];
            save(AeqString,'Aeq')
            beqString=['Elevation Optimization\MatFiles\Solutions STWH by
Period\beqP',num2str(t),'.mat'];
            save(beqString,'beq')
        end
    end

    save('Elevation Optimization\MatFiles\solution.mat','solution')
    save('Elevation Optimization\MatFiles\solTimes.mat','solTimes')
    save('Elevation Optimization\MatFiles\periodNPV.mat','periodNPV')
end
```

f24_Main_Scheduling

```
level=l;
fprintf('Starting Optimization of Layout at Level %d\n',l)
totTime=tic;
f7_Main_Agg
f1a_readPar('Elevation
Optimization\Parameters\ParametersSaha_TN.xlsx','Elevation
Optimization\Parameters\MiningTargets_Saha.xlsx')
fprintf('\nTransferring solution into horizontal/vertical discounting
optimization\n')
f8_buildPU
f9_updPU
fprintf('\nBuilding Mixing Scenarios\n')
f10_mixScenarios
fprintf('\nCalculation Objective Function Coefficients\n')
f11_objFunctionCoeffSch
fprintf('\nBuilding Constraints\n')
fprintf('\nConstraint 1 - Mining Capacity')
f12_constMiningTarget
fprintf('\nConstraint 2 - Reserves')
f13_constReserves
fprintf('\nConstraint 3 - Vertical Precedence')
f14_constVertPrec
fprintf('\nConstraint 4 - Draw Rate')
f15_constMaxDrawRate
fprintf('\nConstraint 5 - Horizontal Precedence')
f16_buildHorPrec
f17_constHorPrec
fprintf('\nConstraint 6 - Undercut Development Rate')
f18_constUndercutRate
fprintf('\nConstraint 7 - Caving Slope')
f19_constDrawControl
fprintf('\nConstraint 8 - Grade Quality')
f20_constGradeBounds
f21_concatenateConst
fprintf('Solving the Optimization Problem\n')
solution = f22_STWH;
save('Elevation Optimization\MatFiles\solution.mat','solution')
save('Elevation Optimization\MatFiles\eTime.mat','eTime')
totalTime = toc(totTime);
save('Elevation Optimization\MatFiles\totalTime.mat','totalTime')
```