Heuristic Stochastic Stope Layout Optimization

by

Brandon Wilson

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

Stope layout optimization is a critical process for the evaluation and optimization of underground mining operations. Conventionally, stopes layouts are either planned manually or optimized using deterministic methods. Developing stochastic approaches to stope optimization is valuable due to the impact of uncertain inputs on optimization results, the non-linear nature of objective functions common to mining, and the large number of variables that impact the results. Deterministic methods are not able to account for non-linearities and produce sub-optimal solutions when the inputs are averaged before the transfer function response is calculated.

In this thesis, an existing two-stage heuristic algorithm that greedily selects strips from a deterministic block model is modified to consider a set of realizations to account for uncertainty in the subsurface and non-linearities in the transfer function. As an example implementation, the algorithm is applied to a set of synthetic copper data with a non-linear recovery function and compared to the results from a deterministic stope optimization, showing an improvement of 4.90% in value.

A full case study is produced using the algorithm for a real-world vein-like deposit, showing the ability for risk evaluation and the improved calculation of non-linear objectives with a 17% increase in value when calculation over realizations is considered. After optimizing layouts for a variety of risk discounts, stochastic dominance is utilized to reduce the field of possible solutions. The layouts are scheduled, and an overall solution is determined from the resulting net present value distributions.

Acknowledgements

I would first like to thank my wife, Rebecca Lambert, for her patience and support as I completed my degree. I would also like to show my appreciation for my supervisor Dr. Jeff Boisvert for his continued advice and direction as I travelled down this path. Without his continued assistance, this would not have been possible.

Finally, I owe gratitude to my family, including my sister Kathryn who loved to tell everyone about the “rock math” I was working with and my parents Lynn and Tom, who were always there when I needed support.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description
λ_i	Weight given to data point i
c	Constant added in inverse distance estimation to avoid unreasonably high weights
$Z^*(u)$	Estimated value at location u
m	Unbiased mean of a given set of data
O_d	Deterministic objective function
O_s	Stochastic objective function
L	Number of realizations
l	Considered realization
v	Value
p	Penalty
B	Set of evaluated entities (stopes or blocks)
α	Risk penalty multiplier
σ	Standard deviation
b	Binary with value 1 if associated entity is mined
r	Process recovery
g	Metal grade
mg	Grade of maximum recovery
C_m	Mining cost
C_p	Processing cost
C_t	Total cost
P	Metal price
γ	Density
cg	Cutoff grade
t_o	Tonnes of ore mined

List of Abbreviations

IDE	Inverse distance estimation
SMU	Selective mining unit
SGS	Sequential Gaussian simulation
SSO	Stope shape optimizer
MILP	Multiple integer linear programming
ODEDA	One-dimensional exact dynamic algorithm
GSLIB	Geostatistical software library
USGSIM	Ultimate sequential Gaussian simulation program

Chapter 1: Introduction

1.1 Context

The ultimate goal of mine planning and optimization is to determine and maximize the value of mineable reserves. Several critical stages in the mine planning process are necessary to produce disclosures that satisfy government requirements and encourage investor support, beginning with the development of an appropriate model of the subsurface based on the available data. The depth and properties of the potentially valuable mineralization lead to a decision between surface and underground mining. If surface is selected, planning proceeds with the optimization of an ultimate pit, the largest incremental excavation that provides value. This pit is then broken into mining stages, or pushbacks, according to a defined rate of mining per period, allowing for an evaluation of time-discounted project value. Underground mining follows a similar process, with an intermediate step of choosing a more specific underground mining method. When optimization is undertaken for underground operations, it takes a variety of forms. In the context of caving mines, the spacing of the draw points is optimized to maximize flow while minimizing access costs. Stope operations are both optimized with envelope methods, finding the maximum area in which mining will occur, and layout methods, which place the stopes considering geotechnical constraints. The stopes are then scheduled by panel or level to inform net present value calculations.

Underground mining plans are generally either manually designed or optimized in stages. Mining extents or stope placements are optimized to maximize contained metal or economic value, and the mining schedule is subsequently optimized for net present value. Additional considerations such as development to access mining areas in underground mining are necessary to inform parameters such as capital expenditure. Results from one stage of the optimization are fed as an input to the next, such as ultimate pit limits informing pushback generation or stope placement into extraction scheduling. These stages are often iterated to approach a solution that is closer to optimal overall.

1.2 Motivation

The staged and iterative nature of mine design and optimization means incremental improvements of any optimization stage results in improvements in the overall outcome. At the same time, improvements made by changing procedures in one aspect of the design process will be dampened if other aspects are not adjusted to account for the changes in the input. Methods to improve the consideration of any aspect of mine optimization are, therefore, valuable to develop. Here, the consideration of risk and non-linearities in stope optimization are identified as areas in which the optimization process can be improved.

Metal content is a significant source of uncertainty in the mine planning process due to the limited sampling of the subsurface. Geostatistical modelling characterizes and quantifies subsurface uncertainty. Refining uncertainty is an ongoing focus in geostatistical research. The ability to develop a model that is representative of subsurface uncertainty facilitates the consideration of risk in optimization. This ability is seen in recent advancements in risk-inclusive optimization, particularly for surface mining (Acorn & Deutsch, 2018; Bootsma et al., 2018; Dimitrakopoulos, 2018; Godoy, 2018; Sotoudeh et al., 2019; Villalba Matamoros & Kumral, 2018; Wilson et al., 2019).

In industry, risk is not often directly considered in mine planning optimizations. Projects with a higher level of risk require a correspondingly higher anticipated return in order to be worth pursuing; this is obvious where two investments are very close in expected value with significant risk differences. Projects can be considered equivalent to investments in this regard, with project decisions represented along an efficient frontier, an early investment comparison scheme that has been adapted for use in mine optimization (Acorn & Deutsch, 2018; Markowitz, 1952). A decision-making paradigm is necessary to choose between options on the frontier, taking a position on risk. The less tolerant a decision-maker is to risk, the greater the value differential required to select a higher risk plan.

Additional value in stochastic optimization is derived from the nature of non-linear transfer functions. When a transfer function does not average linearly, optimization of layouts on deterministic models will not reach a true optimum. Optimal results are only be found through the implementation of optimization over all realizations.

1.3 Thesis Statement

The development and implementation of a stochastic stope optimization algorithm allow for the correct consideration of non-linear functions and the direct inclusion of risk in the optimization process, improving decision-making when designing stope layouts.

1.4 Objective and Scope

The objective of this research is to develop a stochastic approach for stope layout optimization. As a starting point, existing stope optimization techniques are evaluated for conversion to include simulated grade models when assessing the objective. The selected method reduces the dimensionality of the optimization to manageable computational levels, computing an envelope of overlapping stopes within which a layout can be designed.

In this thesis, a 3-dimensional heuristic algorithm is adapted from the existing deterministic method to optimize stope placement over all realizations, allowing for the consideration of uncertainty in optimization as well as non-linearities in the optimized response. The algorithm is formulated to work in stages, first optimizing in one dimension and combining the results into two and three dimensions. While the algorithm is designed to be flexible, there are limits on the mining types it can be applied to due to the nature of underground mining constraints; the factors which determine constraints in caving operations are highly dissimilar to those in stope or drift mining. The presented algorithm is designed to optimize open or sublevel stoping. Additional methods such as drift and fill may be optimizable with the algorithm as well, but this capability is not assessed. The algorithm is expected to perform best where there is selectivity in the mining method at a smaller scale than the ore boundaries. Any method which can be defined by a minimum and maximum size for continuous stopes, pillar requirements between stopes, and expected cost and value functions are optimizable using this algorithm. However, it has not been thoroughly tested for all valid configurations.

As a heuristic algorithm, the optimizer does not guarantee optimal solutions. A tabu search strategy is implemented to close the gap between the heuristic and fully optimal solutions. A further stage is implemented to investigate the value of stochastic scheduling and combining scheduling and stope placement simultaneously. These are developed as a post-process to the initial results. A simulated annealing optimization is implemented to approach an approximately optimal solution iteratively.

1.5 Thesis Overview

The remainder of this thesis consists of 4 chapters. Chapter 2 reviews relevant concepts and research to date, focusing on existing implementations of stochastic stope optimization while reviewing deterministic stope optimization, geostatistical modelling, uncertainty evaluation, decision-making practice, and general methodology in underground mining. Chapter 3 introduces the novel heuristic algorithm to optimize stope layout stochastically as implemented in the Python programming language. The algorithm is validated with no risk discount applied by comparing results to existing stope optimization software. The algorithm is implemented on a case study in Chapter 4, optimizing a stope layout for a previously planned sublevel stoping operation. Concluding remarks are given in Chapter 5, showing that the algorithm is near-optimal, appropriately accounts for risk, and is flexibly applicable to a variety of mining scenarios.

Chapter 2: Research Review

Existing research directly related to this work, as well as a consideration of other materials necessary for its understanding, are reviewed here to provide a contextual base for further developments. This review includes information related to geostatistical modelling, an overview of underground mining methods, general concepts of mine planning and optimization, existing work on stope layout optimization, and a review of basic decision-making concepts.

2.1 Geostatistical Methods

One of the significant inputs to any mining optimization or evaluation is the subsurface grade model, as it can be converted to metal contents and economic values. The grades are based on a limited set of sample data, making the grade model a common and significant source of uncertainty. There are two main types of subsurface models. The first is a deterministic, estimate based paradigm that produces a single "best" representation of the modelled parameter. The second is simulation, where several realizations are presented as equally probable interpretations of the domain. The goal of estimation is to produce a single model that is the best possible estimate of metal grades. There are various approaches to estimation, all of which determine weights to apply to known data for each estimated location in the model. Early estimation consisted of ideas such as interpolation and inverse distance estimation (IDE) methods. Inverse distance methods determine weights for estimation using Eq. 1 (Rossi & Deutsch, 2014).

$$\lambda_i = \frac{1}{c + d_i^{\omega}} \qquad 1$$

Where λ_i is the weight of sample i, d_i is the distance between sample i and the estimate location, c is a constant to limit overweighting of very close samples, and ω is an exponent, often set to 2 or 3, which effectively sets the relative weight given to more proximate samples compared to more distant ones. Setting the value of ω to 0 results in an equally weighted moving average within the search range and setting it to an arbitrarily high value only weights the closest sample, which is equivalent to a nearest-neighbour estimation. Typical implementations of IDE where the value of ω is assumed do not directly account for the continuity of the subsurface and, like all estimation methods, do not reproduce data variance and results in a smooth interpretation of the domain. However, the exponent ω can be calibrated considering the selective mining unit (SMU) variance and Krige's relation (Noble, 2011). IDE can also account for directional continuity by weighting distance measures relative to anisotropy by replacing the Euclidean distance d with the covariance weighted Mahalanobis distance (Rossi & Deutsch, 2014).

Kriging, alternatively, accounts directly for data continuity in its implementation. Kriging is derived to minimize the mean squared error of estimates in an unbiased fashion (Journel & Huijbregts, 1978). It requires knowledge of the covariance among samples as well as that between sample and estimate locations. The modelling of variograms allows the calculation of covariance relations based on distance, facilitating the determination of the optimal kriging weights (Journel & Huijbregts, 1978). Kriging, being an estimation technique, does not reproduce the global distribution of the data, but as an unbiased estimator, the mean is reproduced (Journel & Huijbregts, 1978). While there are a variety of kriging estimators, the most basic is simple kriging. The estimator for simple kriging is defined in Eq. 2 (Rossi & Deutsch, 2014).

$$Z^*(u) = \sum_{i=1}^{n} \lambda_i \, [z(u_i) - m] + m \qquad 2$$

Here, Z* is the estimated value, u is the estimate location, u_i is the location of the i^{th} data point, and $z(u_i)$ is the value of the data at that location. λ_i is the weight calculated for the data at location u_i, and m is the unbiased mean of the data.

The second paradigm of geostatistics is simulation, the generation of several probabilistic models. Where estimation attempts to generate the best single subsurface interpretation, simulation samples the multivariate distribution of possible subsurface grade models reproducing spatial and univariate variability (Rossi & Deutsch, 2014). This includes an accurate representation of extreme values. While no single simulated realization is more correct than the others, the ensemble is a valuable representation of uncertainty (Deutsch, 2018).

A commonly used geostatistical simulation method is sequential Gaussian simulation (SGS). SGS is a Monte Carlo Simulation of possible values at each gridded location facilitated by parameterizing conditional distributions with the kriged estimate as the mean and kriging error as the standard deviation (Rossi & Deutsch, 2014). The methodology of SGS leverages a multiGaussian assumption, the assumption that all higher order probability distribution functions are Gaussian when the univariate cases are transformed to a Gaussian form. This is advantageous as Gaussian functions are fully described with two parameters: mean and variance, making simulation possible at a location where these parameters are known. Because kriging at each location assigns weight to both original data and previously simulated nodes, the spatial continuity of the data is reproduced in each realization.

Estimation and simulation both produce valuable information for the mining process. The paradigm which should be used depends on the anticipated use of the model. The impacts of non-linearities are lost if an estimated model is used, so where non-linearities are known to exist in a transfer function of optimized objective, the use of simulated models is preferable (Deutsch, 2018). Theoretically, averaging a sufficient number of realizations converges to a kriged model, so considering a simulated model where there are no non-linearities instead of an estimated model should not cause a deviation from the correct result. However, where non-linearities do exist, only considering simulated models is the correct approach (Deutsch, 2018).

2.2 Underground Mining Methods

The geostatistical model of the subsurface is one aspect that is considered when the method of exploitation is decided. An early mining decision is whether a surface or underground method is pursued. This decision is based on factors including deposit shape, extent, and depth, as well as location and the social implications of mining in the region (Nelson, 2011). If underground mining is selected, a variety of specific mining methods might be preferred. If the ore is relatively massive and weak enough that undercutting will cause it to fail, caving methods can be employed (Bullock, 2011). These methods are often used to exploit lower grade deposits because the bulk nature of the operations results in lower operating costs. The capital expense of caving, however, is relatively high as there is extensive excavation required before extraction can begin. This investment is only recovered where there is a large volume of extracted ore, especially when considering net present value.

Room and pillar mining is a supported method that maintains the stability of openings by leaving ore unmined as pillars, with the extracted ore forming the corresponding rooms (Bullock, 2011). This method is mostly used for well-delineated, horizontal to sub-horizontal deposits. Coal is commonly mined with room and pillar mining.

Stoping is a group of extractions methods that are primarily used in sub-vertical to vertical deposits that require some selectivity in extraction (Bullock, 2011). Stopes are blasted openings that leverage their steep dip to allow material extraction from a draw-point at the base (Fig. 2.1). The methods are also used in some massive deposits that are vertically expansive enough to create appropriate stoping dimensions.

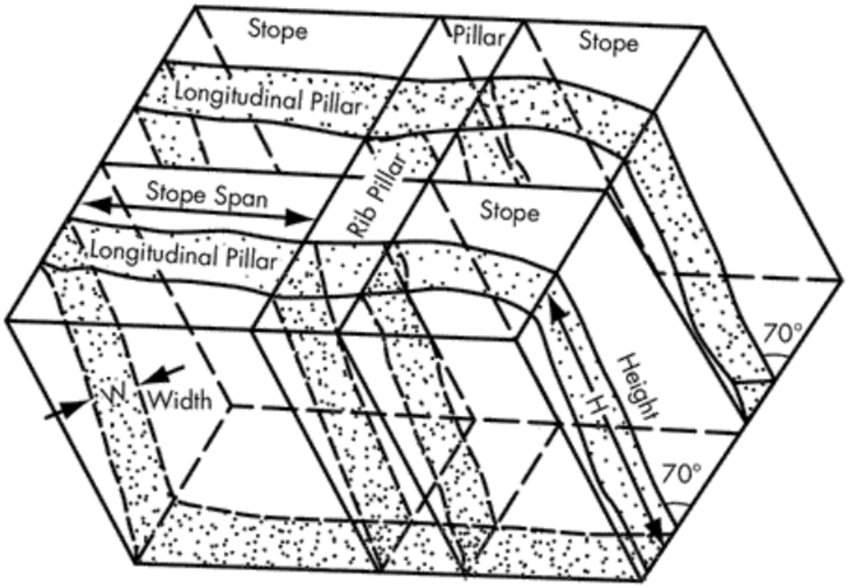


Fig. 2.1: Stope parameters (Pakalnis & Hughes, 2011).

Stopes are bound by minimum and maximum stope sizes derived from geotechnical and equipment requirements, as well as the pillars required to maintain stability (Pakalnis & Hughes, 2011). Types of stoping include sublevel, long-hole, vertical crater retreat, vein mining, and open stoping. These methods vary in terms of typical stope size and orientations, selectivity, costs, and pillar recovery requirements. However, the shapes of the final openings are constrained by similar factors and planning between the methods follows similar processes.

2.3 Mine Planning and Optimization

Whether a surface or underground mining method is selected, the mine planning and optimization processes involve determining what units should be extracted to create a profitable project. Generally, after producing a block model of grades with geostatistical methods, each block is assigned a binary variable representing whether it is mined. A group of mined blocks represents a pit or underground opening and must follow geotechnical and equipment-based limits. Mine design is often completed manually through reduction to 2-dimensional slices, particularly in underground cases (Pakalnis & Hughes, 2011). Surface optimization is more thoroughly studied than underground overall.

2.4 Surface Mining Optimization

The planning and optimization process in surface mining has a long research history with early examples such as the development of the Lerchs-Grossmann algorithm and subsequent improvements to its implementation (Khalokakaie et al., 2000). Monkhouse and Yeates (2018) describe a path forward from what is described as naïve optimization, assuming single values for orebody tonnes and grade, processing methods and costs, maximum sales volumes, commodity prices, and discount rates. They suggest adding flexibility into optimization to account for the daily decisions made by mine planners. The progress and future of global optimization for surface operations are described by Whittle (2018). The existing ProberB optimizer globally optimizes mining, processing, and blending, but is restricted from completing complex operations (Whittle, 2018). ProberC, which has since been released, is capable of handling more complex scenarios such as differential processing costs by stockpile source (Whittle, 2018).

Journel (2018) describes the necessity to account for error with simulated data realizations rather than any best estimate model. The case for preferring structural accuracy over local accuracy for subsurface models is presented as the only eventual path to avoiding biases in the combined modelling and optimization process. Examples of surface operation optimization considering risk are further presented in (Dimitrakopoulos, 2018; Godoy, 2018; Khosrowshahi et al., 2018; Menabde et al., 2018; Ramazan & Dimitrakopoulos, 2018). Acorn and Deutsch (2018) present an additional heuristic method to optimize ultimate pit limits under uncertainty using a novel heuristic method that actively accounts for geological uncertainty.

The primary motivation for exploring the state of surface optimization is the notable gap between surface and underground methods. The early Lerchs-Grossmann algorithm is recognized as an exact solution to the ultimate pit problem, albeit with limitations in its implementation (Khalokakaie et al., 2000). No comprehensive underground equivalent is available.

2.5 Underground Mining Optimization

Alford et al. (2007) provide an overview of underground optimization research encompassing much of the body of work on non-stochastic optimization in underground mining to date, including considerations on optimization of infill drilling, cut-off grade determination, stope layout optimization, development design, decline design, stope scheduling, and some integrated approaches. Generally, underground optimization is completed in these separate stages, resulting in suboptimal solutions (Little et al., 2013). More recent studies consider various combinations of these stages to form integrated optimizations. In one instance, Little et al. (2013) integrate stope placement and scheduling into a single optimization. However, the process only considers regularly spaced stopes with two possible sizes, limiting the exploration of the solution space. The algorithm operates on a deterministic model. Block caving is also a focus of recent study, as Zarate et al. (2020) consider the optimization of drawpoint placement over many realizations produced by SGS.

The varying constraints and controlling factors between the various underground methods require the development of a variety of optimization schemes to facilitate mine planning and maximization of value. Reviews of existing deterministic stope optimization methods have been produced (Erdogan et al., 2017; Nhleko et al., 2018). The methods are generally divided into envelope, layout, and geometry methods. Envelope methods, such as floating stope and maximum value neighbourhood, optimize an outer boundary of stope extraction within which individual stopes are planned by hand (Alford et al., 2007). Layout methods, on the other hand, output stopes that are extracted over the life of mine. Geometry methods consider individual stopes, varying their dimensions to maximize value or contained metal above cutoff locally. While layout methods are the focus of this work, notable work on stope geometry optimization includes Manchuk's use of simulated annealing to vary the position of the defining points of a stope triangulation (2007). Manchuck enforces constraints on each point based on boundary conditions such as adjacent stopes. Case studies utilizing this method improve in value by 15%.

2.5.1 Stope Layout Optimization

Stope layout optimization is still a focus in current research, as indicated by recent advancements in both deterministic and stochastic methods. For deterministic methods, both heuristic and exact algorithms are considered. Sens and Topal (2010) extend floating stope principles, selecting greedily from the available stopes while disallowing overlapping stopes from the solution. This transitions from an envelope result to a stope layout. The greedy nature of the algorithm does not ensure that a global optimum is reached. Sandanayake (2014) introduces an algorithm that also considers combinations of non-overlapping stopes as initially defined through a floating stope implementation. In this case, all non-overlapping sets of stopes are assessed and compared, leaving only the maximum layout as the solution. It is found, however, that this method is prohibitive in examples with a large number of stopes, leading to approximations replacing the precise solution, including a reduction of the number of retained solution sets to a maximum number to avoid time and memory issues.

Bootsma et al. (2018) approach stope optimization following the cut-off grade ideas introduced for surface mining by Lane (2016). In this approach, layouts are optimized in an existing commercial package considering various cut-off grades. The NPV for each layout is calculated to compare results. Since NPV calculation is time-dependent, accesses are manually designed to facilitate extraction scheduling. This method improves the cut-off grade selection process but does not guarantee an optimal result.

There are multiple methods used in the stope shape optimizer (SSO) as implemented in multiple commercial mining software such as Maptec Vulcan (*Vulcan Envisage*, 2018). The methods include a slice and seed, which is often used in the mining industry. Slice and seed works by evaluating 2-dimensional slices of the ore body and selecting regions that exceed cutoff grades on average. The seeds are combined between layers to form stopes, and iterations are run to alter the seeds and improve outcomes, often maximizing contained tonnage over the cutoff grade. The algorithm is limited to optimization with 1 or 2 degrees of freedom, although pillars in other dimensions can be added through the substoping utility. Dimensions that are not optimized by stope are considered by altering the gridded framework on which seeds are generated, and stopes are placed. An alternate prism method is also available in the SSO that uses linear programming to determine the best set of stopes that can be mined in a given area (*Vulcan Envisage*, 2018). The problem is divided into smaller subsets because of the complexity of the optimization. The resulting solutions are combined at the end of the process.

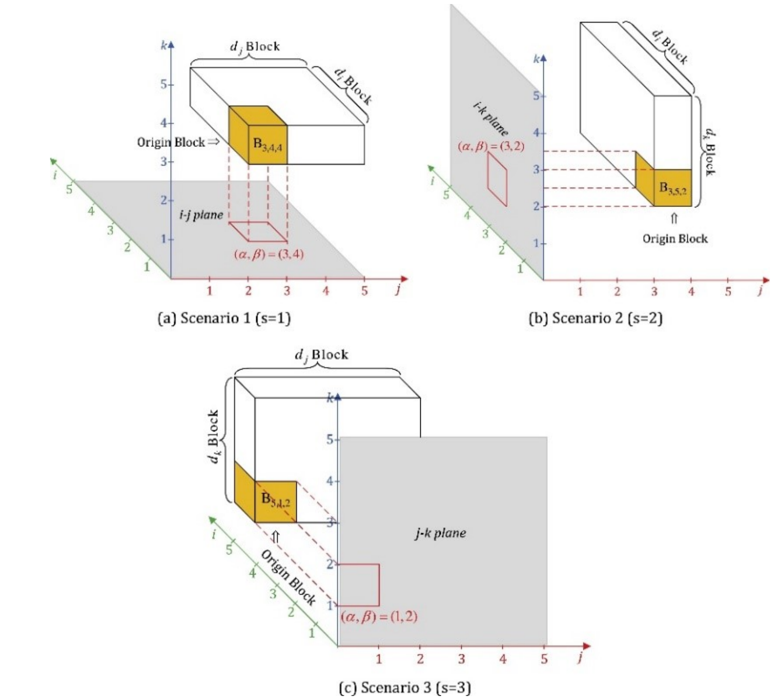
Nikbin et al. (2018) produce an integer programming model to optimize stope boundaries by implementing valid cutting planes to improve convergence and optimization speed. The cutting planes allow the algorithm to solve the stope boundary problem in a case study with a model of 473,600 blocks, but it is noted that more complex stope boundary optimization problems might still be unsolvable. Two configurations of a heuristic iterative enumeration algorithm are also produced to address this issue and solve larger problems. This algorithm invokes either a greedy or dynamic algorithm with an input number of stopes in a specified range within the number found through the implementation of an initial greedy algorithm. The first greedy algorithm is similar to that presented in Sens and Topal (2010), adding each maximum value stope to the solution as long as they remain positive. A second, improved greedy algorithm is also introduced by Nikbin et al. (2019). This algorithm randomly populates an initial solution based on an assumed total number of stopes and greedily exchanges stopes that are in the solution with ones that are excluded based on their value difference. When no beneficial exchanges are found, the algorithm is halted. Finally, an approximate dynamic algorithm is produced. This algorithm is also initialized with a random solution of a set number of stopes. In this instance, sets of stopes to exchange between included and excluded sets are developed, described as states. At each stage, the best value of all current and previous states is retained.

Nikbin et al. (2018) produce an additional algorithm that combines the dynamic and greedy algorithms in stages to produce 3D results. This algorithm follows these steps:

- Divide the model into overlapping slices perpendicular to each axis based on the minimum stope size in that dimension (Fig. 2.2)
- Combine slices into strips parallel to each axis (Fig. 2.3)
- Optimize each strip in one dimension using a one-dimensional exact dynamic programming algorithm
- Greedily select the best strip as determined by best ratio (Eq. 3)
- Add blocks from selected strip to solution set
- Set value of blocks included in the solution to 0
- Repeat steps 1-6 until no strips with positive values are calculated

$$BR = max \frac{\sum_{\gamma=1}^{\Gamma} c_{\gamma}^{*} c_{\gamma}^{+}}{| \sum_{\gamma=1}^{\Gamma} c_{\gamma}^{*} c_{\gamma}^{-} \ |} \qquad 3$$

Here BR is the best ratio, c^{*} is a binary variable with a value of 1 if block γ is mined and 0 if it is not, while c_{γ}^{+} and c_{γ}^{-} are binary parameters with a value of 1 if stope γ is positive or negative, respectively.



(a) Scenario 1 (s=1)

(b) Scenario 2 (s=2)

(c) Scenario 3 (s=3)

Fig. 2.2: Slice population with corresponding origin blocks (Nikbin et al., 2018). Scenarios in (a), (b), and (c) show i-j, i-k, and j-k slices respectively

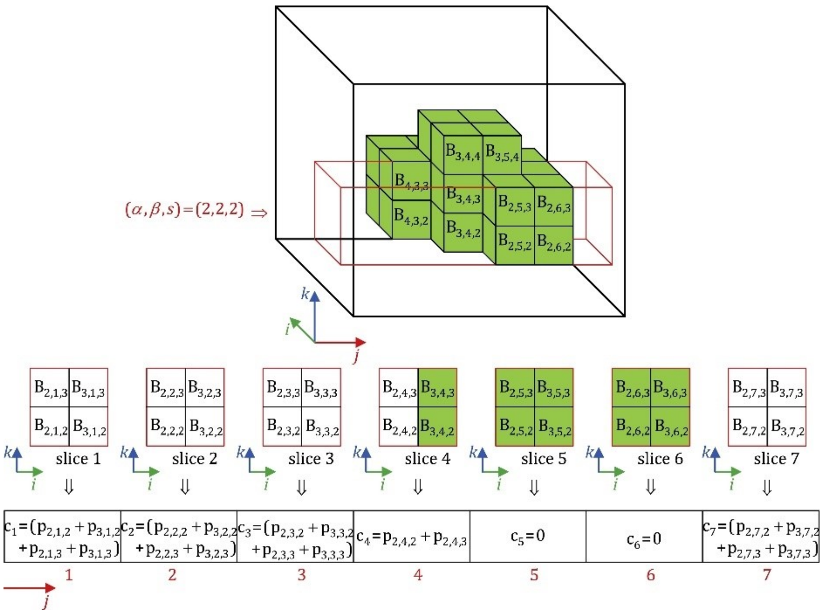

$(\alpha,\beta,s)=(2,2,2)$ ⇒
$B_{3,4,4}$
$B_{3,5,4}$
$B_{4,3,3}$
$B_{3,4,3}$
$B_{4,3,2}$
$B_{3,4,2}$
$B_{2,5,3}$
$B_{2,6,3}$
$B_{2,5,2}$
$B_{2,6,2}$
k
i
j
$B_{2,1,3}$ $B_{3,1,3}$ $B_{2,1,2}$ $B_{3,1,2}$
slice 1
$B_{2,2,3}$ $B_{3,2,3}$ $B_{2,2,2}$ $B_{3,2,2}$
slice 2
$B_{2,3,3}$ $B_{3,3,3}$ $B_{2,3,2}$ $B_{3,3,2}$
slice 3
$B_{2,4,3}$ $B_{3,4,3}$ $B_{2,4,2}$ $B_{3,4,2}$
slice 4
$B_{2,5,3}$ $B_{3,5,3}$ $B_{2,5,2}$ $B_{3,5,2}$
slice 5
$B_{2,6,3}$ $B_{3,6,3}$ $B_{2,6,2}$ $B_{3,6,2}$
slice 6
$B_{2,7,3}$ $B_{3,7,3}$ $B_{2,7,2}$ $B_{3,7,2}$
slice 7
⇓
$c_1=(p_{2,1,2}+p_{3,1,2}+p_{2,1,3}+p_{3,1,3})$
$c_2=(p_{2,2,2}+p_{3,2,2}+p_{2,2,3}+p_{3,2,3})$
$c_3=(p_{2,3,2}+p_{3,3,2}+p_{2,3,3}+p_{3,3,3})$
$c_4=p_{2,4,2}+p_{2,4,3}$
$c_5=0$
$c_6=0$
$c_7=(p_{2,7,2}+p_{3,7,2}+p_{2,7,3}+p_{3,7,3})$
1
2
3
4
5
6
7

Fig. 2.3: Strip population (red) including the behaviour of previously included blocks in value calculation (Nikbin et al., 2018)

Ideas of stochastic optimization are also explored by using the risk-based option in a commercial stope optimizer (Bootsma et al., 2018). This function allows the user to consider a suite of realizations which represent uncertainty in the subsurface. A user set limit is then applied to determine the minimum allowable certainty to include a stope in the resulting plan. For example, when 100 realizations are considered, if the stope is over cut-off in 30 realizations, it has a 30% confidence. In the case study presented by Bootsma et al. (2018), it is expected that a stope that is above cut-off on a kriged model has low confidence when the set of realizations is considered, with the majority of stopes ranging from 0% to 30%. In this application, it is shown that the previously optimized layout is only profitable in one realization, and only when no discounting is considered. This is because of the underrepresentation of low and high grades and the spatial variability in the estimated model. The risk-based optimizer is run with minimum confidence levels from 0 to 80%, and the resulting layouts are assessed to evaluate the potential for a lower risk mine plan (Fig. 2.4). The stopes in the plan are so uncertain that a cut-off of 40% confidence, a state in which some stopes are still more likely to be below cut-off than above, is selected for further analysis because further limiting stope confidence resulted in unminable layouts. The optimization is set to maximize the objective (Eq. 4).

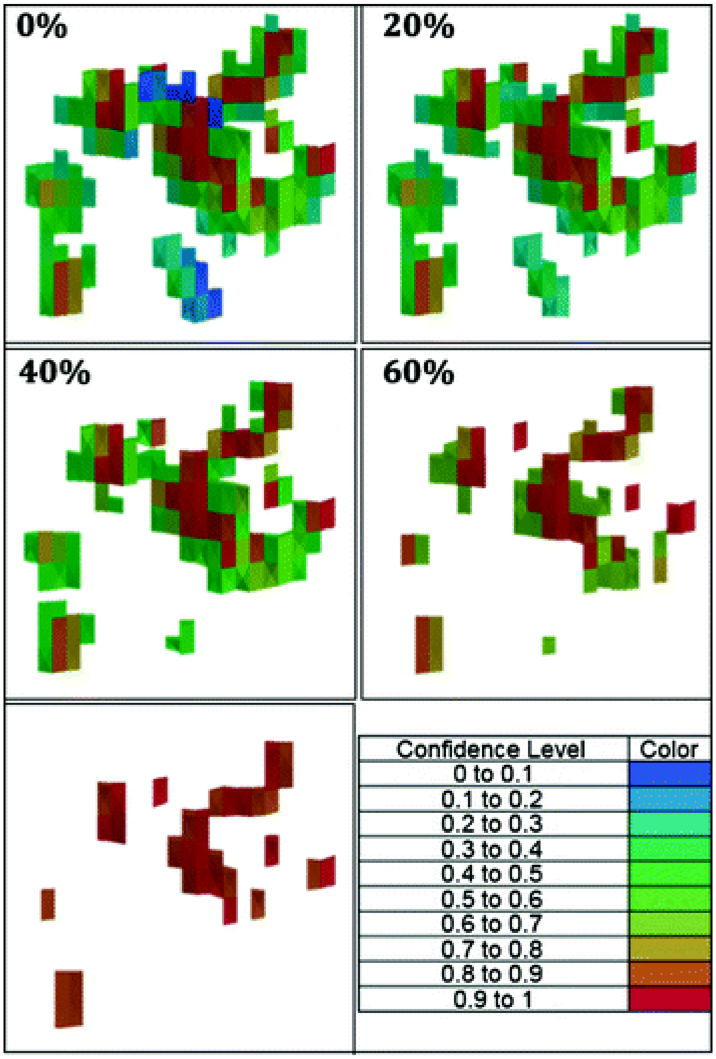


Fig. 2.4: Stope layout by minimum confidence (Bootsma et al., 2018)

Dimitrakopoulos and Grieco (2009) suggest an alternate approach to stochastic stope optimization. This application centers on the use of Multiple Integer Linear Programming (MILP) and includes a similar risk limit, as described in Bootsma et al. (2018). The flexibility in stope positioning is limited for this approach to maintain a manageable problem size (Dimitrakopoulos & Grieco, 2009). To facilitate MILP, the block model is divided into sections which can be assigned a variable value representing either inclusion or exclusion from the mining plan. First, the block model is divided into panels that are optimized independently. These panels are further divided into 2-D rings, which are the units considered in the optimization. The rings have a fixed height and lateral position. Combinations of rings form planned stopes, abiding by set stope size and pillar constraints.

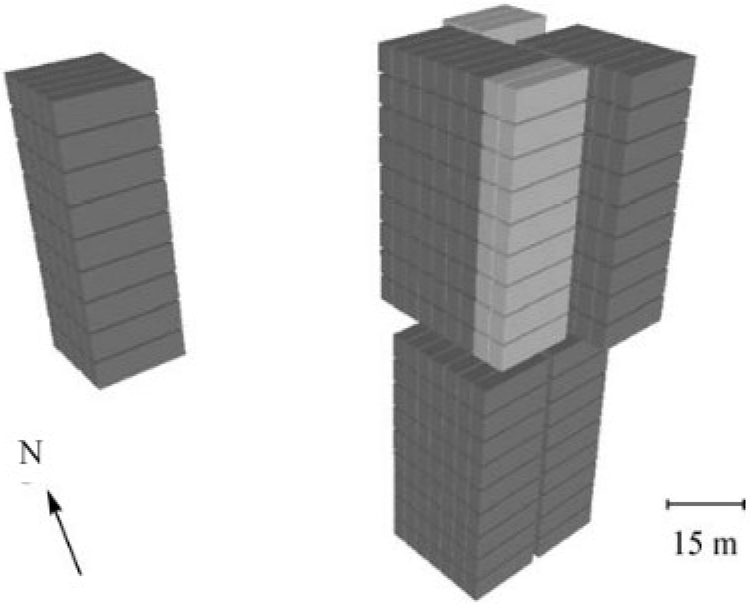


Fig. 2.5: Schematic of rings and stopes. Rings are vertical stope sections consisting of 9 consecutive blocks. Consecutive rings form stopes (Dimitrakopoulos & Grieco, 2009).

$$\sum_{j=1}^{m}\sum_{i=1}^{n} g_{ij}p_{ij}B_{ij} \qquad 4$$

Where m is the number of panels, n the number of rings, g is the grade of a ring above the cutoff, p is the probability for the ring to be above the cutoff, and B is the integer variable indicating whether the ring is included in the plan. The results are subjected to a panel-by-panel risk limit (Eq. 5).

$$\sum_{i=1}^{n}\left(p_{ij} - PL\right)B_{ij} \geq 0 \qquad 5$$

Here, PL is a defined risk limit, similar to the minimum confidence level from Bootsma et al. (2018). There is a difference in application, as here the limit applies panel by panel, allowing individual stopes with lower confidence to be included in the design, even if it is below the risk limit on an individual basis (Dimitrakopoulos & Grieco, 2009).

Genetic algorithms are heuristics that alter solutions according to rules similar to evolution in nature. Two applications of genetic algorithms are presented to optimize stope layouts stochastically. Verhoff (2017) considers the value of stopes over all realizations as an objective function – the value of the stope configuration is assessed over each realization and averaged, and this average value is maximized. A genetic algorithm is implemented to improve random parent chromosomes through selection, cross-over, mutation, constraint enforcement, and elitism. The algorithm is limited to two dimensions (Verhoeff, 2017).

The genetic algorithm presented by Matamoros and Kumral (2018) begins by deterministically optimizing each input realization and combining the results geometrically to an average stope layout (Fig. 2.6). This algorithm maximizes value while penalizing dilution through the application of a recovery function. The solution is also limited as a certain percentage of included stopes must also appear in the average solution.

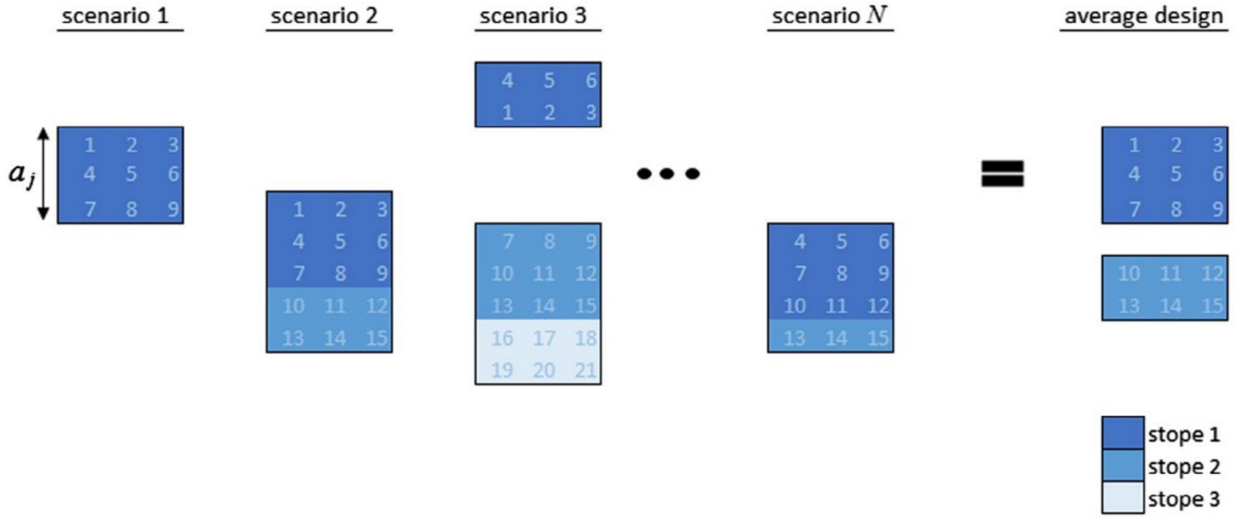


Fig. 2.6: Geometric averaging of N scenario plans to average design (Villalba Matamoros & Kumral, 2018)

Sotoudeh et al. (2019) adapt the deterministic algorithm from (Sandanayake, 2014) to consider stochastic input. The adapted algorithm produces a layout for each of the input realizations following the algorithm produced by Sandanayake (2014) and does not consider a stochastic objective function.

2.5.2 Production Scheduling Optimization

Production scheduling is important to consider in conjunction with layout optimization as layout results act as the input for production scheduling. The ideal case is concurrent stope layout and schedule optimization, but no general deterministic or stochastic case for this has been presented.

If combined optimization is not an option, an appropriate stochastic stope scheduling optimization algorithm should be implemented. Otherwise, the benefits of the stochastic layout optimization will be lost. Stope production scheduling is reviewed here, showing a lack of stochastic options.

Maybee et al. (2010) develop a heuristic framework for stope schedule optimization, the schedule optimization tool (SOT). A genetic algorithm is implemented as the basis of this framework, iterating on an initial schedule to find a better solution. The SOT uses a process called guidance that rates each stope by a series of parameters to help drive the decision-making of the algorithm. After each run, sliding is implemented. Sliding is a process that delays development until it is required, delaying and increasing NPV. The guidance portion of the algorithm results in a reduced time to completion of the algorithm, while reducing the probability of finding the optimal solution.

Nehring and Topal (2011) produce an MIP production scheduling optimization program. The program enforces a production order of prepare, extract, void, fill, optimizing a 9x9 series of stopes in 2:45:08. Little et al. (2013) streamline existing MIP formulations by making the simple assumption that stope related processes will follow each other directly without delay. This formulation reduces the number of decision variables as only one variable is necessary as a decision. All other constraints are derived from the results. The algorithm is applied to a 2-dimensional example consisting of 50 stopes. A result is reached in under 3 hours, a large reduction from the 64 hours required for a classical approach.

Manchuk (2007) presents a general framework for stope schedule optimization as an alternative to integer methods. Both a simulated annealing and probabilistic decision-making method are presented. Random restarts are used to escape local minima and reach near-optimal solutions.

Recently, Rosado et al. (2019) focus on quantifying geological risk by optimizing stopes on each realization with an unspecified optimization tool. Density is calculated based on a density-grade regression model, compounding the impact of grade uncertainty on the resulting metal content calculations. The model results are combined to assess the total probability that grades are above both the estimated value for the stope and the average grade of the reserve. These probabilities are combined into a risk indicator with values from 1 (low) to 3 (high). Lower risk stopes are prioritized early in the extraction schedule, showing a decrease in the standard deviation of grade, particularly in early periods.

Sari and Kumral (2019) develop an MILP schedule optimization that uses chance constraints and a risk-intolerance factor to generate a risk-inclusive stope schedule. The algorithm considers stability and backfill constraints, but not stope access precedence.

2.6 Nonlinearities and Stope Optimization

One benefit of using stochastic grade models in mine optimization is that calculation over all realizations correctly accounts for the nature of averaging when the transfer function is non-linear. Linear functions are defined as those which can be represented as a straight line and, conversely, non-linear cannot be. Examples of non-linear functions are quadratic and exponential functions. In mining, it is expected that the processing recovery relates non-linearly to input ore grade and it is known that non-linear transfer functions must be averaged after calculation to obtain the correct result where the input includes uncertainty (Deutsch, 2018). An example is shown here to depict the importance of correctly accounting for nonlinearities in value calculation.

The first consideration for this example is when recovery is applied. The level of support that is correct to consider when applying process recovery varies according to the scenario being optimized. Factors such as the size of a batch as it enters the processing plant and the implementation of ore blending change how grade values should be averaged. If it is a batched process that, for example, uses milling to generate correctly sized material for the rest of the processing circuit, each batch should be considered separately. If it is a continuous process choosing what represents a batch is more complicated and further assumptions are made. This includes considering the smallest available scale, such as block scale if recovery is expected to vary quickly, or a larger scale, such as the stope scale, if the process is less sensitive to head grade changes. An additional consideration is what ore is combined or blended before processing occurs. It is common to blend ore to maintain a consistent head grade or dilute deleterious elements, changing what grade value is appropriate to use in recovery calculations. Considering blending at the time of stope layout optimization is difficult due to the necessity of timing information.

Some assumptions are made here to simplify the consideration of both timing and batching on value calculations. Due to the lack of timing information in layout optimization, blending is not considered. Instead, single stopes are divided into batches individually. Within each stope, various batch sizes can be used. In this example, a batch is either a single mined block or a single mined stope.

Three recovery functions are presented that inflect at their maximum values, which occurs at grades of 2.5 and 1.5. Recovery functions with various symmetries are evaluated. (Figs. 2.7 to 2.9).

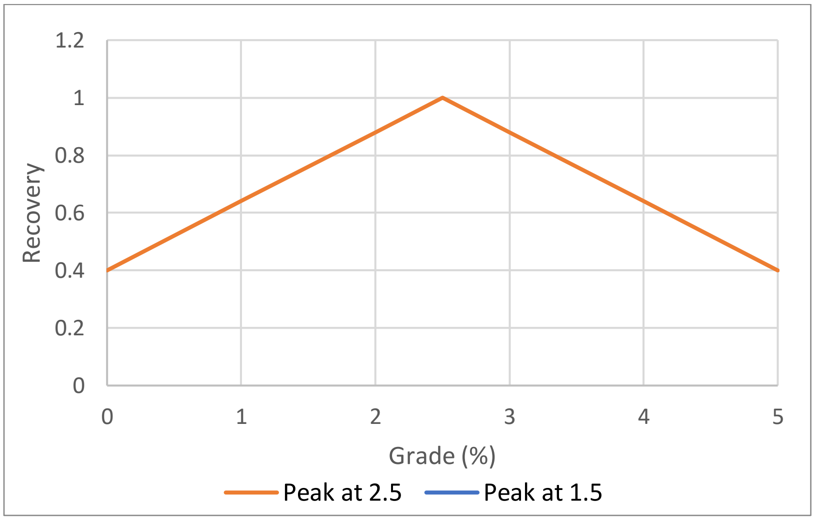


Fig. 2.7: Symmetric recovery functions

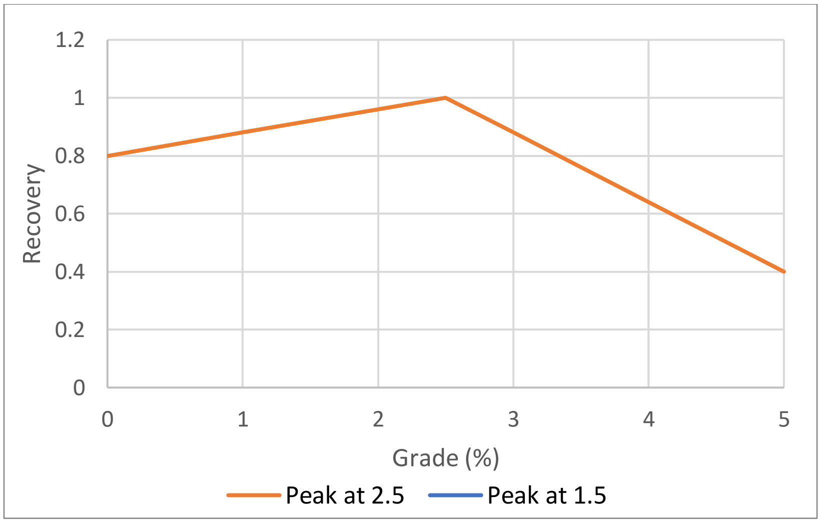


Fig. 2.8: Low grade favoured recovery functions

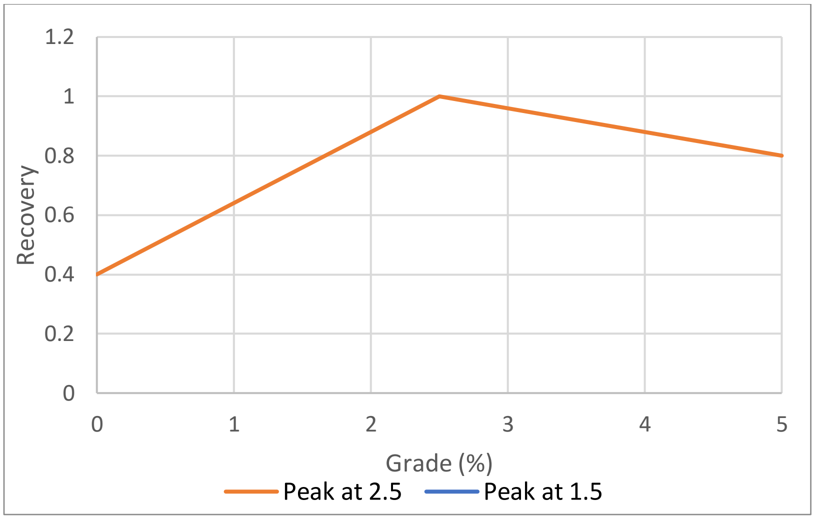


Fig. 2.9: High grade favoured recovery functions

Five realizations of a single 2-dimensional stope are produced to evaluate the importance of nonlinear calculation. The stope dimensions are 2 blocks by 8 blocks. Grade values between 0 and 5 are assigned randomly to each block by drawing from a high tailed distribution (Eq. 6, Fig. 2.10). The resulting stopes and average case are shown in Fig. 2.11.

$$F = \frac{\log(g+1)}{\log(6)}, g \in (0,5) \qquad 6$$

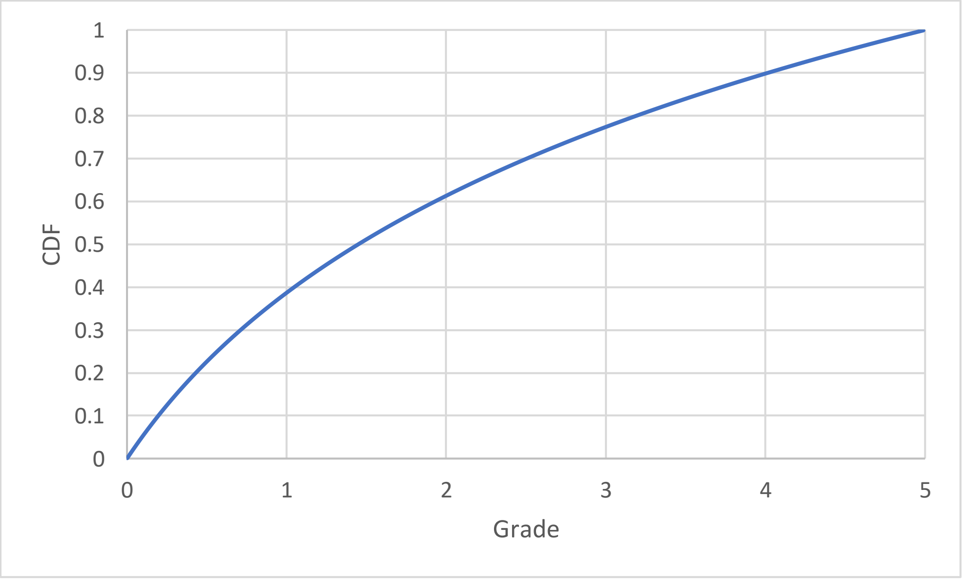


Fig. 2.10: CDF of grade distribution

1.15	1.74
1.51	1.72
2.43	2.28
4.75	1.01
2.97	2.21
4.59	2.75
0.76	0.36
3.10	1.55

(a)

1.84	0.07
0.21	1.46
0.41	2.67
0.44	4.25
3.48	0.20
2.01	1.42
0.07	1.47
0.74	1.12

(b)

0.56	0.31
0.68	2.68
1.94	0.21
4.78	3.56
2.75	3.01
1.26	1.84
2.70	0.51
1.68	0.13

(c)

0.45	0.84
0.50	3.16
1.62	0.27
0.90	1.44
0.67	0.40
1.96	0.21
4.35	2.12
2.14	2.71

(d)

4.64	2.10
1.52	1.25
1.55	1.47
4.80	4.88
3.64	0.62
0.26	0.65
1.52	1.30
0.86	0.48

(e)

1.73	1.01
0.88	2.05
1.59	1.38
3.13	3.03
2.70	1.29
2.02	1.38
1.88	1.15
1.70	1.20

(f)

Fig. 2.11: Stope realizations (a-e) and average case (f)

The average grade in the stopes is 2.18, 1.37, 1.79, 1.48, and 1.97 respectively, with an overall average of 1.76. For this example, units are inconsequential; however, considering all grade values to be g/t is reasonable.

The recovery is calculated in stochastic and deterministic frameworks considering both block and stope scale support for processing batches. This process is repeated for each recovery function, as presented above (Figs 2.7 to 2.9). The results for each case are shown in Fig. 2.12 and Fig. 2.13.

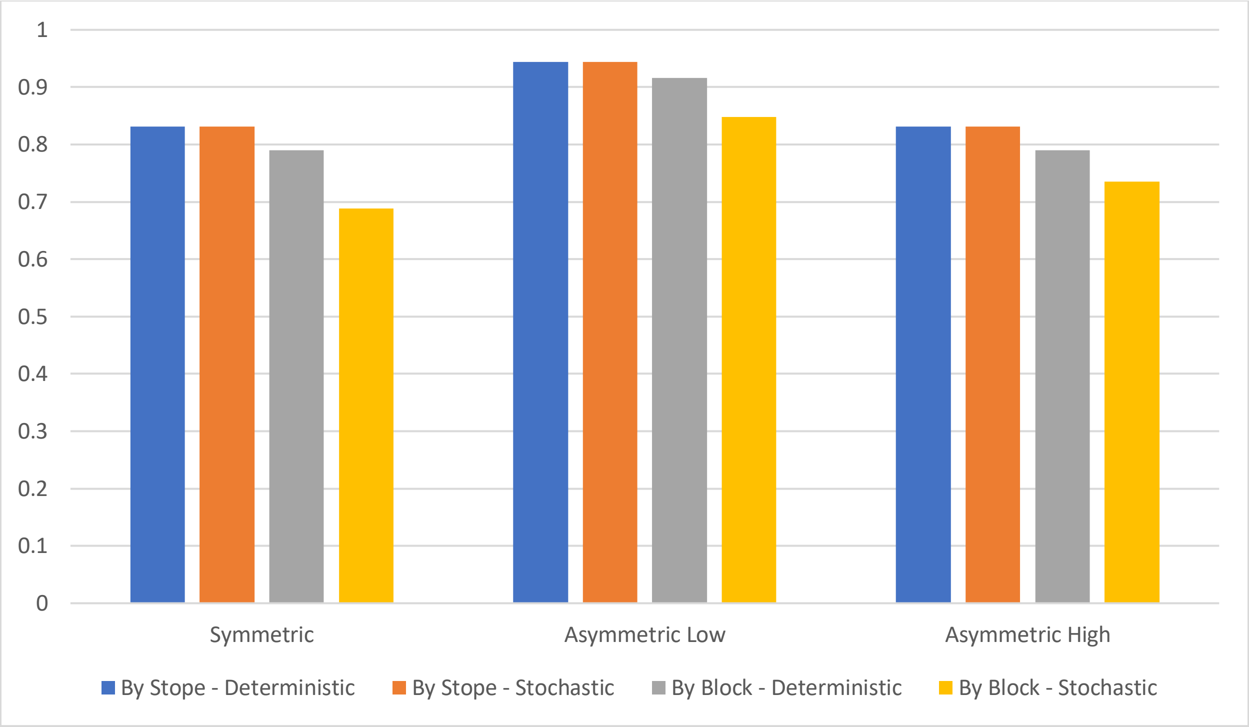


Fig. 2.12: Recovery by paradigm and support with various recovery function shapes and peak recovery at 2.5

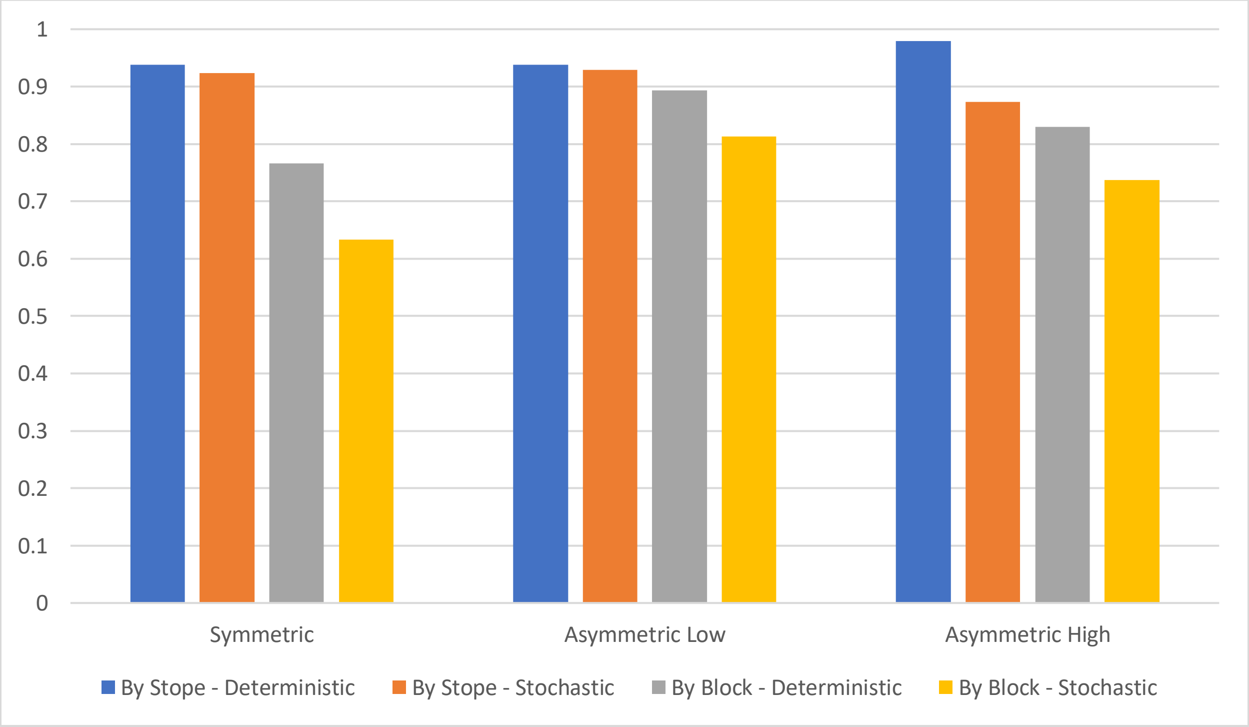


Fig. 2.13: Recovery by paradigm and support with various recovery function shapes and peak recovery at 1.5

Notably, when the peak recovery is set at a grade of 2.5, there is no difference between deterministic and stochastic calculation by stope (Fig. 2.13). While this appears to contradict the assertion that stochastic calculation is the correct method, the reason for this discrepancy can be seen in Fig. 2.14.

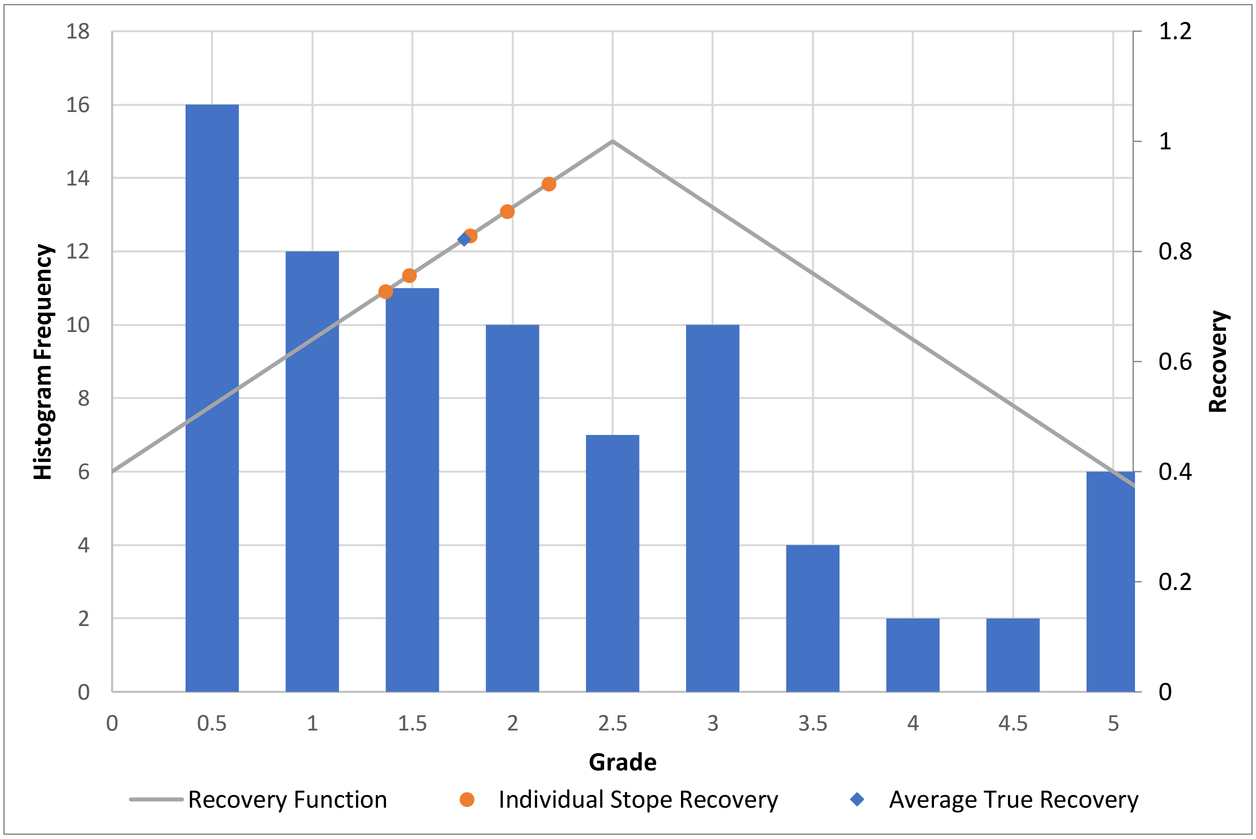


Fig. 2.14: Grade distribution and recovery by stope with symmetric recovery inflection at 2.5 including block grade distribution (bars)

In this case, the average values of the stopes all fall on the same side of the inflection point. This interval of the function is linear, leading to the equivalency of the average case. The block support case, alternatively, results in different deterministic and stochastic results due to the value distribution across the recovery function.

This evaluation is supported by the results for the 1.5 peak recovery case: the stochastic and deterministic results vary for both the stope and block scale calculations. Fig. 2.15 shows the distribution of the individual and average stope cases overlaid on the recovery function.

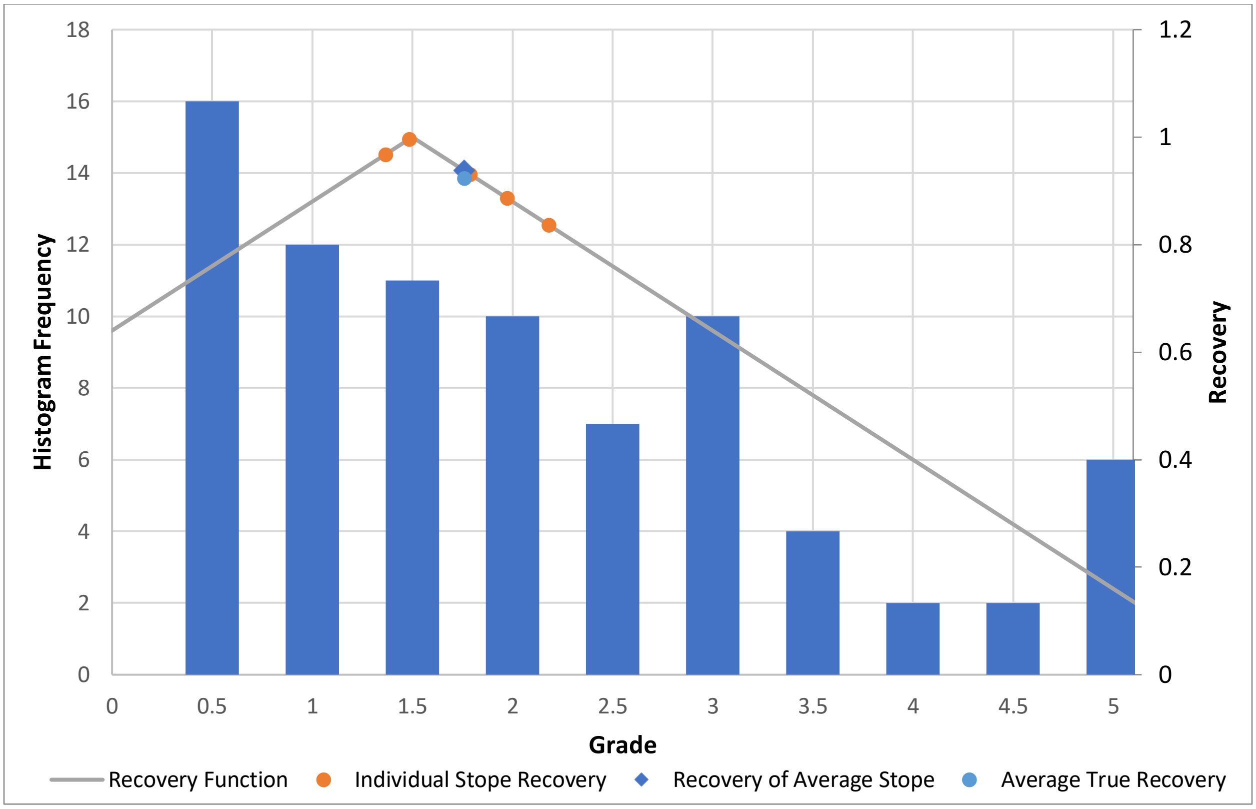


Fig. 2.15: Grade distribution and recovery by stope with recovery inflection at 1.5 and symmetric recovery

In this case, the recovery difference is not stark, with only a 1.5% overestimation when considering the deterministic calculation. Tabs. 2.1 and 2.2 show the percent difference between deterministic and stochastic calculations for all the considered cases.

Tab. 2.1: Value change with maximum recovery at 2.5

Calculation	Summary	Percent Difference
By Stope	Symmetric	0.0%
	Asymmetric Low	0.0%
	Asymmetric High	0.0%
By Block	Symmetric	14.8%
	Asymmetric Low	8.0%
	Asymmetric High	7.4%

Tab. 2.2: Value change with maximum recovery at 1.5

Calculation	Summary	Percent Difference
By Stope	Symmetric	1.5%
	Asymmetric Low	1.0%
	Asymmetric High	12.1%
By Block	Symmetric	21.2%
	Asymmetric Low	10.0%
	Asymmetric High	12.7%

The difference when calculating recovery by stope ranges between 0% and 11.7%, with the deterministic case always exceeding the stochastic. Because of the low number of stopes and random nature of grade assignment, there are no outlying high or low stope grades as might exist in a real case. The difference between deterministically and stochastically block scale recoveries is much more notable, ranging from 7.4% to 21.2% (Tab. 2.1). Note that in all cases, recovery is higher in the deterministic case than the stochastic because the recovery function is concave down. This is confirmed in the results with the inflection at 1.5 (Tab. 2.2), with stope scale recovery differing by between 1% and 12.1%, and block scale recovery by between 10% and 21.2%. These differences are significant and have a direct impact on value calculations.

A final extreme case is posed as a theoretical example, where two realizations of a single block are modelled. High and low realization values are selected such that their mean is representative of the reference distribution. Consider the symmetric recovery function with a peak at a grade of 2.5 and realization grade values of 0 and 5. Both values correspond to the minimum recovery value of 0.38 from the symmetric distribution and average to 0.38. However, the average of the grade values, 2.5, results in a recovery of 1 when assessed with the same function. This is a 163% overestimation.

From these examples, some generalizations are drawn:

- Stochastic calculation is often significantly different from deterministic calculation
- If the interval of the recovery function utilized in the stochastic calculations is approximately linear, the deterministic calculation is equivalent
- Deterministic recovery is always overestimated if the recovery function is concave down

- Larger processing batches have a lower discrepancy than smaller batches due to the averaging which takes place in calculating batch grades, reducing the presence of extreme values

This evaluation shows that non-linear transfer functions are not correctly assessed when deterministic estimates or averaged values are generated before the calculation is completed. Due to the principles of MCS, it is understood that calculating over all realizations is the correct approach when considering non-linear transfer functions (Deutsch, 2018). Unless a study can be undertaken that shows approximate linearity in a calculation, stochastic methods should always be pursued to ensure the best model is produced. Additional benefits of optimizing on a stochastic grade model, such as risk analysis, are secondary to this issue.

2.7 Decision-making and risk

Optimization of the mine plan is a decision between many potential extraction options. Because of the large number of possible non-overlapping stope combinations that make up the solution space, the decision-making process must limit the solution space explored. The number of possible solutions is initially reduced by constraints placed on minable blocks. Underground mining constraints vary widely by the specific mining type. In general, there are geotechnical limits that inform the maximum size of openings and equipment-based limits that inform the minima (Bullock, 2011).

To decide between valid plans, a means for comparison must be determined. A method that can be used to populate an initial subset of possible solutions is the idea of an efficient frontier in portfolio theory (Markowitz, 1952). This theory recognizes that a decision can only be correct if it maximizes expected value at a certain level of risk, creating what is known as an efficient frontier of viable options. While the idea of the efficient frontier is nearly 70 years old, it is still considered in portfolio optimization (Elton et al., 2014). Choosing between solutions on the efficient frontier requires taking a position on risk tolerance, a qualitative process (Deutsch, 2018). In general, risk tolerance can be tuned to either seek, avoid, or take a neutral stance on risk. One approach to defining risk tolerance includes producing objective functions that penalize risk. It common to evaluate a range of positions on risk and work towards an overall solution. Ideally, a formal decision-making framework is adapted to formulate a single value that can be maximized or minimized appropriately, such as expected utility or regret theory (Bleichrodt & Wakker, 2015).

Expected utility combines aspects of a problem into a single maximizable value, utility. It can include penalties due to the expectation of unwanted characteristics in a possible decision such as risk in the case of stope optimization. Regret theory, alternatively, looks to minimize the realization of regret if the chosen course of action was not the best choice in the real case (Loomes & Sugden, 1982). Both paradigms can be applied to mining scenarios where planning alternatives are considered (Wilson et al., 2019). Treating potential stope layouts as planning alternatives leads to an optimization problem where risk can be accounted for directly by optimizing not value but a parameter that is calculated to include the risk of a layout that is being considered.

2.8 Summary

Stope optimization is an integral part of the mine planning process where stoping methods are employed. Geostatistical methods are well developed in their ability to access joint risk through simulation modelling paradigms such as SGS. While stope optimization methods exist that access this quantification of risk and include its impacts on the stope optimization process, no comprehensive method has been produced. Existing stochastic optimization methods that are reviewed in this chapter are limited either in their consideration of risk, the methods which they consider, their ability to produce mineable layouts, or the flexibility of stope and constraint consideration. As mentioned in Chapter 1, the goal of this research is to develop a flexible stochastic stope optimization algorithm, improving the treatment of non-linearities and consideration of risk with flexibility in the possible layouts. Viewing the stope optimization problem under a decision-making lens leads to the implementation of decision-making paradigms such as utility or regret theory allow the selection of an optimal solution that accounts for the risk which is inherently present. The remainder of this thesis leverages this perspective to generate a novel stochastic stope optimization heuristic based on existing deterministic heuristics.

Chapter 3: The Algorithm

It is possible to adapt deterministic integer methods to stochastic inputs by considering a linear combination of all input values (Eqs. 7, 8); however, this adds additional complexity to a problem that is already often unsolvable in deterministic cases. Heuristics can similarly be adapted to optimize an objective function which is assessed over all realizations and are less dependent on problem setup to be able to find a reasonable or optimal solution.

$$O_d = v - p \tag{7}$$

$$O_s = \frac{1}{L}\sum(v_l - p_l) \tag{8}$$

Here O_d and O_s are deterministic and stochastic objective functions, respectively, formulated as the sum of a value and penalty functions v and p. In the stochastic formulation, v_l and p_l are equivalent functions assessed over each realization. When scaled by the number of realizations, L, O_s is an average probabilistic response. The possibility to substitute a stochastic objective into an existing heuristic algorithm simplifies the algorithm development process. A variety of existing algorithms, as reviewed in Chapter 2, are considered for adaptation: the greedy solution introduced by Topal and Sens (2010), the set-based method from Sandanayake (2014), and the combined greedy and dynamic algorithm presented by Nikbin et al. (2018). These algorithms are reviewed in Section 2.5.

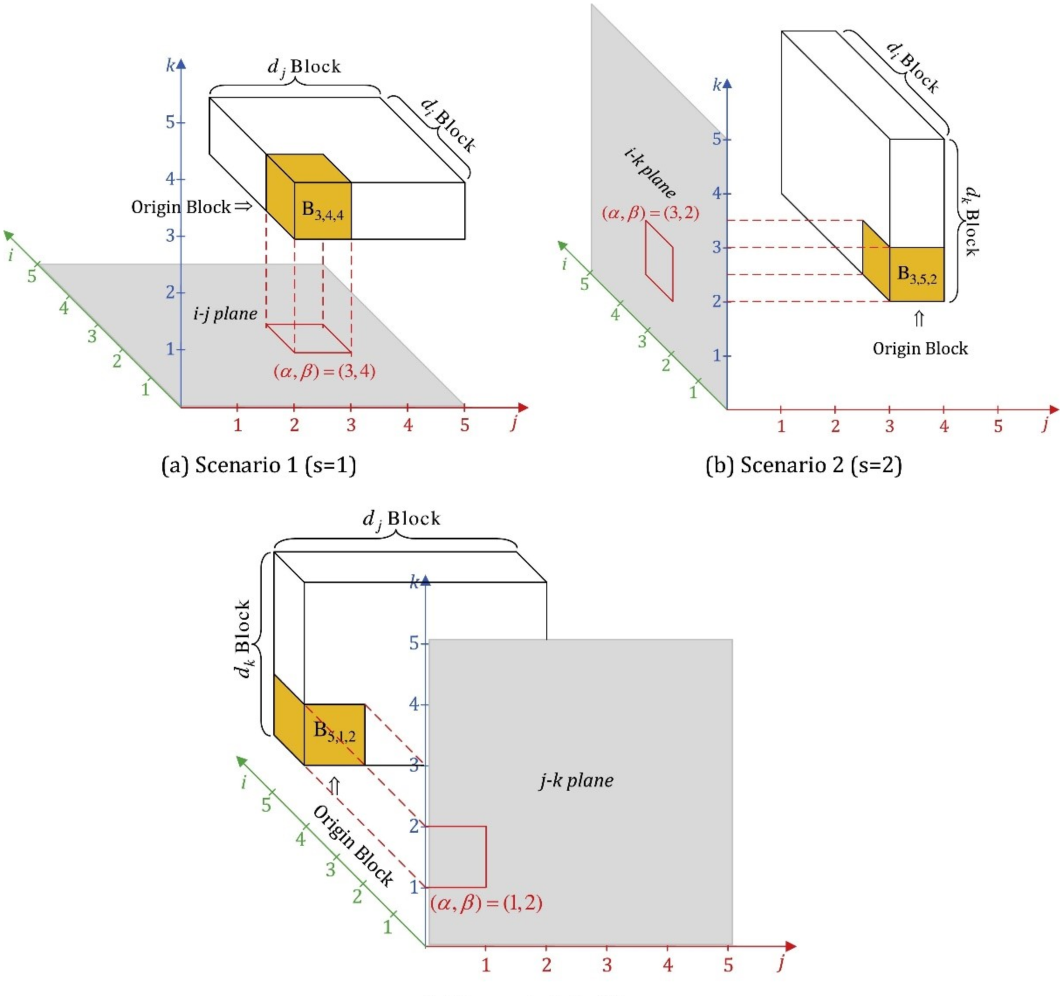
The algorithms are compared by their ability to solve the stope optimization problem efficiently, consider reasonable constraints, and their ability to be adapted to a stochastic case. It is assumed that a stochastic implementation will increase the complexity and time requirements of any existing algorithm. The set-based stope combination optimization is the most rigorous approach but is limited to small problem cases even in deterministic implementations (Sandanayake, 2014). The greedy and combined algorithms both utilize greedy heuristics that select the best incremental solutions but can miss globally optimal results (Nikbin et al., 2018; Topal & Sens, 2010). The main difference between the two is the dynamic algorithm that is a part of the Nikbin et al. method (2018). The dynamic algorithm obtains an exact result by reducing the problem to 1-dimensional sub-problems. It is also common, as described in Section 2.2, that stopes are aligned to facilitate access and extraction, limiting the negative impact of forcing stopes to be aligned when the linear problems are combined. The exact nature of the dynamic portion of the algorithm is expected to find solutions that are not found by a purely greedy algorithm while maintaining algorithm speed. Pillar placement is also accounted for in the one-dimensional problems.

The algorithm selected is the deterministic algorithm presented by Nikbin et al. (2018). It is expected to out-perform the pure greedy algorithm as presented by Topal and Sens (2010) while achieving computational efficiency that will allow for the optimization of larger problems than the Sandanayake algorithm (2014). The following sections show further detail for this algorithm and how it is adapted for stochastic use.

3.1 Original Algorithm

The stochastic algorithm presented here is adapted from the work of Nikbin et al. (2018). The original deterministic algorithm proposed an alternative to commonly used integer programming frameworks. The two-stage heuristic system developed by Nikbin et al. (2018) as an alternative to integer methods follows these steps:

1. Divide the model into overlapping slices perpendicular to each axis based on the minimum stope size in that dimension (Fig. 3.1)
2. Combine slices into strips parallel to each axis (Fig. 3.2)
3. Optimize each strip in one dimension using a one-dimensional exact dynamic programming algorithm
4. Greedily select the best strip as determined by best ratio (Eq. 9)
5. Add blocks from selected strip to solution set
6. Set value of blocks included in the solution to 0
7. Repeat steps 1-6 until no strips with positive values are calculated



(a) Scenario 1 (s=1)

(b) Scenario 2 (s=2)

(c) Scenario 3 (s=3)

Fig. 3.1 Slice population with corresponding origin blocks (Nikbin et al., 2018). Scenarios in (a), (b), and (c) show i-j, i-k, and j-k slices respectively

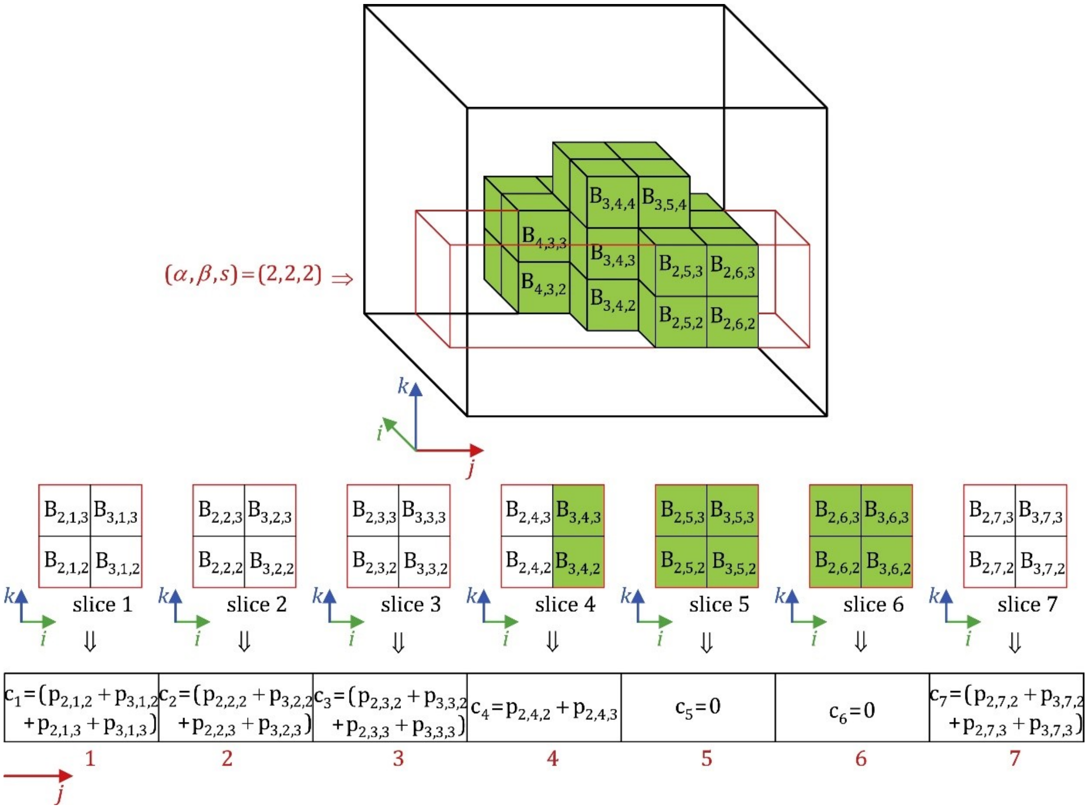

$B_{3,4,4}$
$B_{3,5,4}$
$B_{4,3,3}$
$B_{3,4,3}$
$B_{2,5,3}$
$B_{2,6,3}$
$B_{4,3,2}$
$B_{3,4,2}$
$B_{2,5,2}$
$B_{2,6,2}$
$(\alpha,\beta,s)=(2,2,2) \Rightarrow$
k
i
j
$B_{2,1,3}$ $B_{3,1,3}$ $B_{2,1,2}$ $B_{3,1,2}$
slice 1
$B_{2,2,3}$ $B_{3,2,3}$ $B_{2,2,2}$ $B_{3,2,2}$
slice 2
$B_{2,3,3}$ $B_{3,3,3}$ $B_{2,3,2}$ $B_{3,3,2}$
slice 3
$B_{2,4,3}$ $B_{3,4,3}$ $B_{2,4,2}$ $B_{3,4,2}$
slice 4
$B_{2,5,3}$ $B_{3,5,3}$ $B_{2,5,2}$ $B_{3,5,2}$
slice 5
$B_{2,6,3}$ $B_{3,6,3}$ $B_{2,6,2}$ $B_{3,6,2}$
slice 6
$B_{2,7,3}$ $B_{3,7,3}$ $B_{2,7,2}$ $B_{3,7,2}$
slice 7
$c_1=(p_{2,1,2}+p_{3,1,2}+p_{2,1,3}+p_{3,1,3})$
$c_2=(p_{2,2,2}+p_{3,2,2}+p_{2,2,3}+p_{3,2,3})$
$c_3=(p_{2,3,2}+p_{3,3,2}+p_{2,3,3}+p_{3,3,3})$
$c_4=p_{2,4,2}+p_{2,4,3}$
$c_5=0$
$c_6=0$
$c_7=(p_{2,7,2}+p_{3,7,2}+p_{2,7,3}+p_{3,7,3})$
1
2
3
4
5
6
7
j

Fig. 3.2: Strip population (red) including behaviour of previously included blocks in value calculation (Nikbin et al., 2018)

$$\mathrm{BR} = \max \frac{\sum_{\gamma=1}^{\Gamma} c_\gamma^* c_\gamma^+}{\left|\sum_{\gamma=1}^{\Gamma} c_\gamma^* c_\gamma^-\ \right|} \qquad 9$$

BR refers to the best ratio, γ is the stope index, while Γ is the total number of stopes. c^* is a binary indicator with a value of 1 if block γ is mined and 0 if it is not, while c_γ^+ and c_γ^- are binary parameters with a value of 1 if stope γ is positive or negative, respectively. This results in a ratio of positive and negative stopes in each optimized strip.

The combined algorithm is designed to generate a stope boundary similar in nature to a floating stope or maximum value neighbourhood algorithm, meaning it does not result in the generation of a mineable plan but a reduced set of stopes from which to develop a mine plan manually. This is a function of the decision to set the value of blocks included in interim solutions to 0, allowing them to be included in further stopes with no value or penalty. This can be seen in the value calculations in Fig. 3.2 and is accounted for during the adaptation of the algorithm.

The one-dimensional portion of the algorithm is a one-dimensional exact dynamic algorithm (ODEDA) (Nikbin et al., 2018). The ODEDA leverages the reduced dimensionality of the optimization to solve a lower complexity problem. Only constraints in one dimension are considered when a strip is optimized. The algorithm populates stope values beginning at the minimum index of the strip and combines the minimum number of slices needed to generate a valid stope. Each stope is evaluated to find the maximum value combination of all previously considered non-overlapping stopes. The process is illustrated in Figs. 3.3 - 3.5. Use of the ODEDA reduces the optimization space which must be explored to generate a solution compared to an exhaustive search while guaranteeing an optimal solution in the one dimension considered. The ODEDA proceeds as follows:

1. Select the first stope index in a given strip
2. Populate the value of each stope that begins at current index
3. Calculate the total value for the current stope in combination with any non-overlapping previously populated stopes (including pillar considerations)
4. Retain the maximum value and state for each stope size considered at this index
5. Advance to next index
6. Repeat steps 2-5 until no minimum size stopes can be populated
7. Overall maximum value and corresponding state are retained as the solution

The optimization is simple for the initial stopes as no potential overlapping is considered (Fig. 3.3). The best incremental solution, in this case, is always just the stope itself. Once an index with the potential for non-overlapping stope combinations is reached, all such possible combinations are assessed, and the incremental maximum value layout is retained for each index (Fig. 3.4). The linear progression of the optimization means that if an evaluated stope does not overlap with a previously evaluated stope, it also does not overlap with the solution which is retained at that point of the optimization. After evaluating the final stopes, the incremental solution with the maximum value is retained as the overall solution (Fig. 3.5).

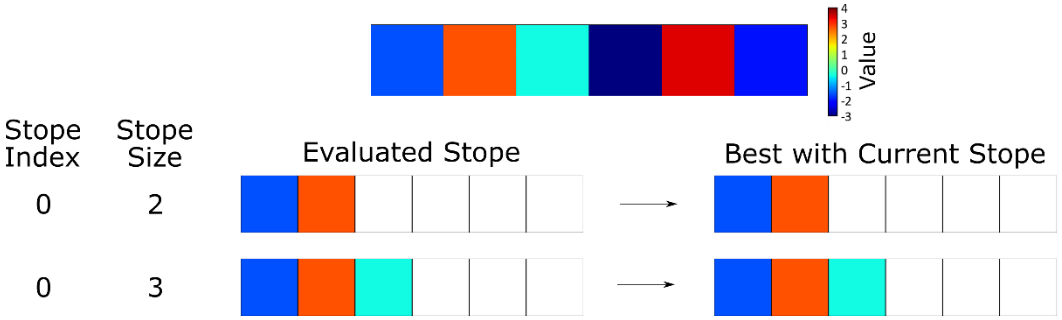


Fig. 3.3: Data and population of first stopes in 1-dimensional dynamic algorithm

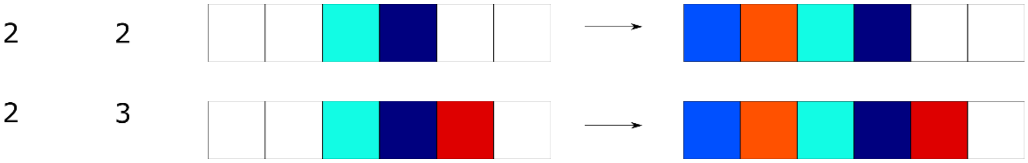


Fig. 3.4: Assessment of stopes at index of 2 with best non-overlapping combination of previously assessed stopes

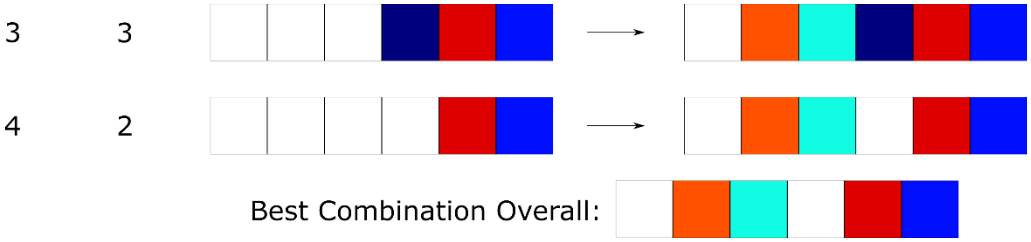


Fig. 3.5: Final stope assessments, associated best incremental solutions, and overall solution

3.2 Adaptation of Algorithm

The algorithm introduced by (Nikbin et al., 2018) is adapted to produce a stochastic stope layout algorithm by 1) altering the objective function of the algorithm to consider stochastic information in the form of grade model realizations including a consideration of risk, and 2) changing the results from a stope boundary optimization to a stope layout optimization. Additional changes are made to reflect typical mining practices, such as applying unified stoping levels. The specifics of these changes are discussed in the following sections.

3.2.1 Stochastic Objective Function

In the deterministic algorithm, two objective functions are considered. The dynamic portion maximizes value, and the greedy portion selects by the best ratio (Eq. 10). The first change is to have the objective function in both stages of the algorithm refer to a value-based objective with the ability to consider risk through the implementation of a proportional risk discount. Eq. 10 shows the objective function.

$$\mathrm{O_s}(B, \alpha) = \sum_{i=0}^{I} (v_{B_i} - \alpha \times \sigma_{B_i}) \times \mathrm{b}_{B_i} \qquad 10$$

O_s is the stochastic objective value for a set of entities B evaluated with risk penalty multiplier α. B is assumed to be a set of stopes but could also refer to individual blocks. B_i is the stope from set B with index i and v_{B_i} is the averaged stochastic value of stope i over all realizations. The standard deviation of the value over all realizations is given by σ, and b_{B_i} is a binary variable representing whether stope i of set B is mined. I represents the total number of stopes in the model. The calculation for v_{B_i} is shown in Eq. 12.

$$v_{B_i} = \left(\frac{1}{L} \sum_{\mathrm{l}=0}^{L} \bar{\mathrm{g}}_{\mathrm{B_i},l} \times r_{\bar{g}} \times m_{B_i} \right) - c_m - c_p \qquad 11$$

Here, L is the number of realizations in the model, while l refers to an individual realization. $\bar{\mathrm{g}}_{B_i,l}$ is the average grade of all blocks in stope B_i on realization l, $r_{\bar{g}}$ is the recovery associated with the calculated average grade and m_{B_i} is the mass of all blocks in stope i. The costs of mining and processing are given by c_m and c_p respectively.

Stochastic value is used in the objective function rather than the best ratio (Nikbin et al., 2018), controlling dilution through a recovery function and grade averaging (Eq. 12). Dilution can have a significant impact on the realized value from an underground deposit, but it should not be the only parameter that controls optimization. Recovery functions are selected to reflect the impact of dilution based on deposit or analogue data and can include penalties for cases such as contamination with a deleterious by-product.

The implementation of recovery can result in a non-linear value function. Where this occurs, using deterministic or averaged input values does not result in the correct output response (Deutsch, 2018). This is illustrated in Eq. 12.

$$\frac{(1+2)^2}{2} \neq \frac{1^2 + 2^2}{2} \qquad 12$$

This shows the necessity of introducing a stochastic evaluation of value to correctly account for the value of a stope layout, even where risk limitations are not considered.

Selecting proper ranges and values for α is not a simple decision. The efficient frontier must be sufficiently informed to decide between the possible plans without omitting the best option but optimizing for additional risk discounts demands more resources. This demand is not, however, prohibitive for the practitioner assuming a high level of computing power and sufficient time. The number of considered values can also be increased after initial options are assessed, better filling in the critical region of the efficient frontier. This likely generates solutions that do not fall below the efficient frontier and are not dominated stochastically. In an exhaustive implementation, each incremental α value is associated with the removal of one stope from the layout until the layouts begin to be dominated by previous options. It is considered sufficient to evaluate a set of chosen α values that produces a set of value distributions that vary to the degree that is meaningful to the scenario. While what constitutes a meaningful difference between scenarios might vary, a target of about 1% variation is suggested to be meaningful.

3.2.2 Dimensional Considerations

Selecting optimized one-dimensional strips parallel to all the x, y, and z axes generates results that ignore conventional mining constraints, producing layouts that are difficult to mine. This is primarily because stoping operations are often mined with unified drilling and extraction levels (Fig. 3.6)

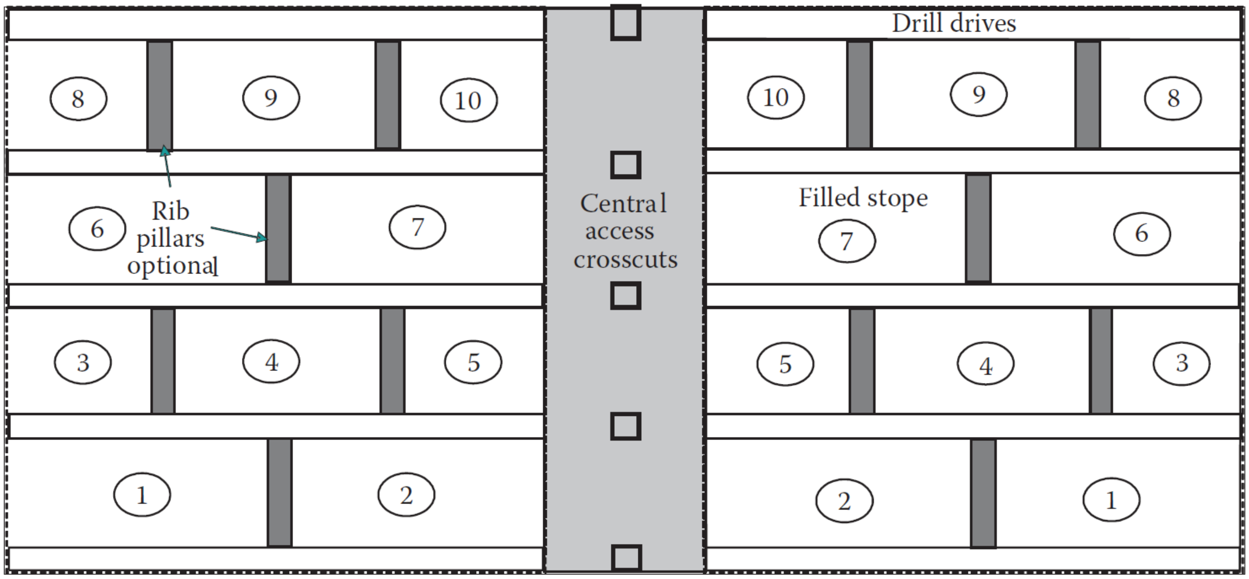


Fig. 3.6: Stoping layout showing level based organization and rib pillars (Villaescusa, 2014)

A typical mining constraint is to mine from unified drilling and extraction levels. The greedy portion of the algorithm is retooled to work in two dimensions to account for this restriction, generating optimized mining levels that are then combined to form a full 3-dimensional layout. The greedy algorithm is altered to select from strips parallel to the x and y axes. Extension to 3-dimensional results is achieved by first implementing the 2-dimensional algorithm on every possible mining level in the domain. A non-overlapping combination of levels that maximizes OBJ is found using a third dynamic algorithm. In this case, each level is assigned an index and height. The optimization follows the workflow defined for the ODEDA.

3.2.3 Layout Optimization

Layout optimization is achieved by disallowing overlapping stopes in the solution. To disallow overlapping combinations, previously included blocks are set to an arbitrarily large negative number. If pillars are required based on the current layout, the associated blocks are also disallowed from future solutions. The greedy algorithm does not consider the potential impact of including stopes in an incremental solution on adjacent strips. The value of adjacent strips is reduced when pillars are required after one strip is mined, disallowing blocks from the solution (Fig. 3.7). Due to the one-dimensional nature of the dynamic algorithm, any blocks disallowed from adjacent strips are not considered as a penalty before the implementation of the greedy algorithm. These pillars' position will be sub-optimal because they are not directly optimized.

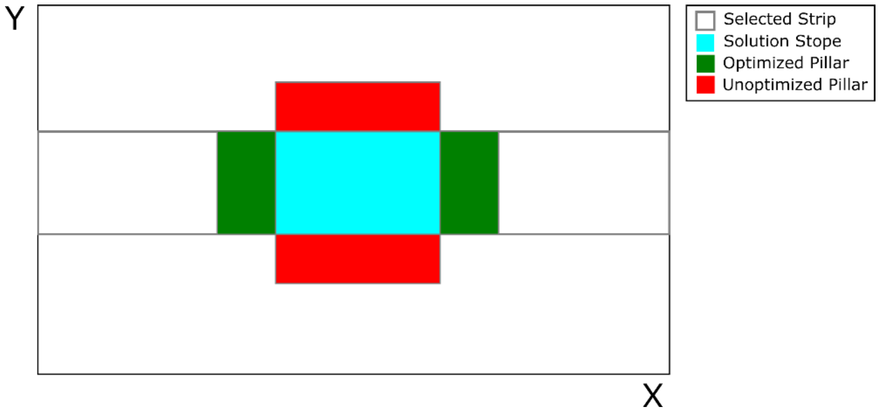


Fig. 3.7: Schematic of pillar optimization limitations for greedy portion of combined algorithm. Only pillars which lie in the selected strip are optimized by the dynamic algorithm

3.3 Heuristic Stochastic Stope Optimization

A detailed overview of the produced stochastic algorithm, as adapted from Nikbin et al. (2018), is provided here. This section explores the basic implementation of the algorithm, while Section 3.4 reviews alternate and optional modules.

3.3.1 Parameters Input

A number of input parameters are required in order to enforce the constraints on the algorithm and to inform the value and objective calculations (Eqs. 10 and 11). The area to be optimized is set by defining a three-dimensional grid. The parameters used in the algorithm are shown in Tab. 3.1. They are described in further detail in Appendix A.

Tab. 3.1: Input parameters for stochastic stope optimization

Parameter	Symbol(s)
Stope size minima (blocks)	$xmin, ymin, zmin$
Stope size maxima (blocks)	$xmax, ymax, zmax$
Model size	$Xdim, Ydim, Zdim$
Model origin	$Xmin, Ymin, Zmin$
Block sizes	Xsz, Ysz, Zsz
Pillars	$xpil, ypil, zpil$
Risk scaling factor	α
Mining cost per block	C_m
Processing cost per block	C_p
Metal price	P
Density	γ
Recovery	Function

Note that metal recovery is not defined with a parameter but is a function. All dimensional parameters are defined by block numbers except the block model origin and block sizes. The risk scaling factor α can be defined as a single value or range of values to compare the resulting layouts.

3.3.2 Preprocessing

In the preprocessing phase of the algorithm, all possible stopes in the layout are populated. The value of each possible stope is calculated on each realization. Value calculation and recovery are applied at the stope level. Applying recovery by stope is equivalent to assuming that no blending of the ore occurs before processing. The value function is shown in Eq. 11. The process for data and parameter input, as well as data initial data formatting, is shown in Fig. 3.8.

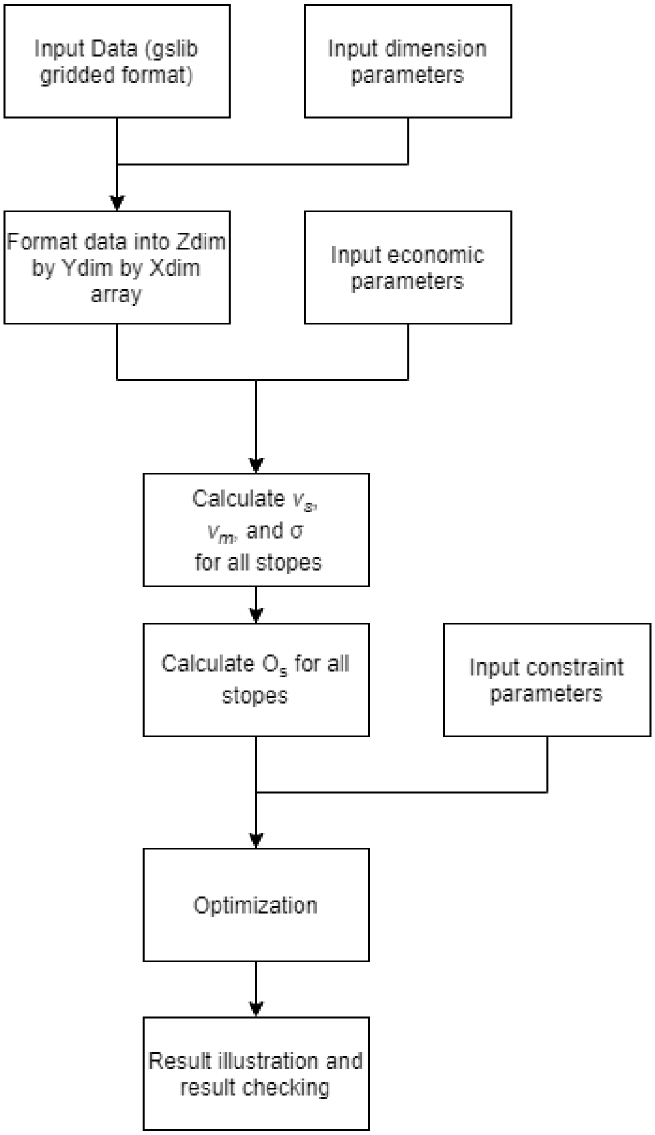

Input Data (gslib gridded format)
Input dimension parameters
Format data into Zdim by Ydim by Xdim array
Input economic parameters
Calculate v_s, v_m, and σ for all stopes
Calculate O_s for all stopes
Input constraint parameters
Optimization
Result illustration and result checking

Fig. 3.8: Flow chart of data parameter input and preprocessing. v_s retains the value of a stope from each realization while v_m is the average of those values

3.3.3 Strip Optimization

To optimize each strip, a dynamic algorithm is implemented to find the optimal one-dimensional solution for a given set of possible stopes following the procedure described in Figs. 3.3 to 3.5 with the additional consideration of pillars. The algorithm maintains a minimum number of blocks between each stope as set by the user. Every strip parallel to the x and y axes are optimized for input into the next algorithm stage. This stage of the algorithm is detailed in Fig. 3.9.

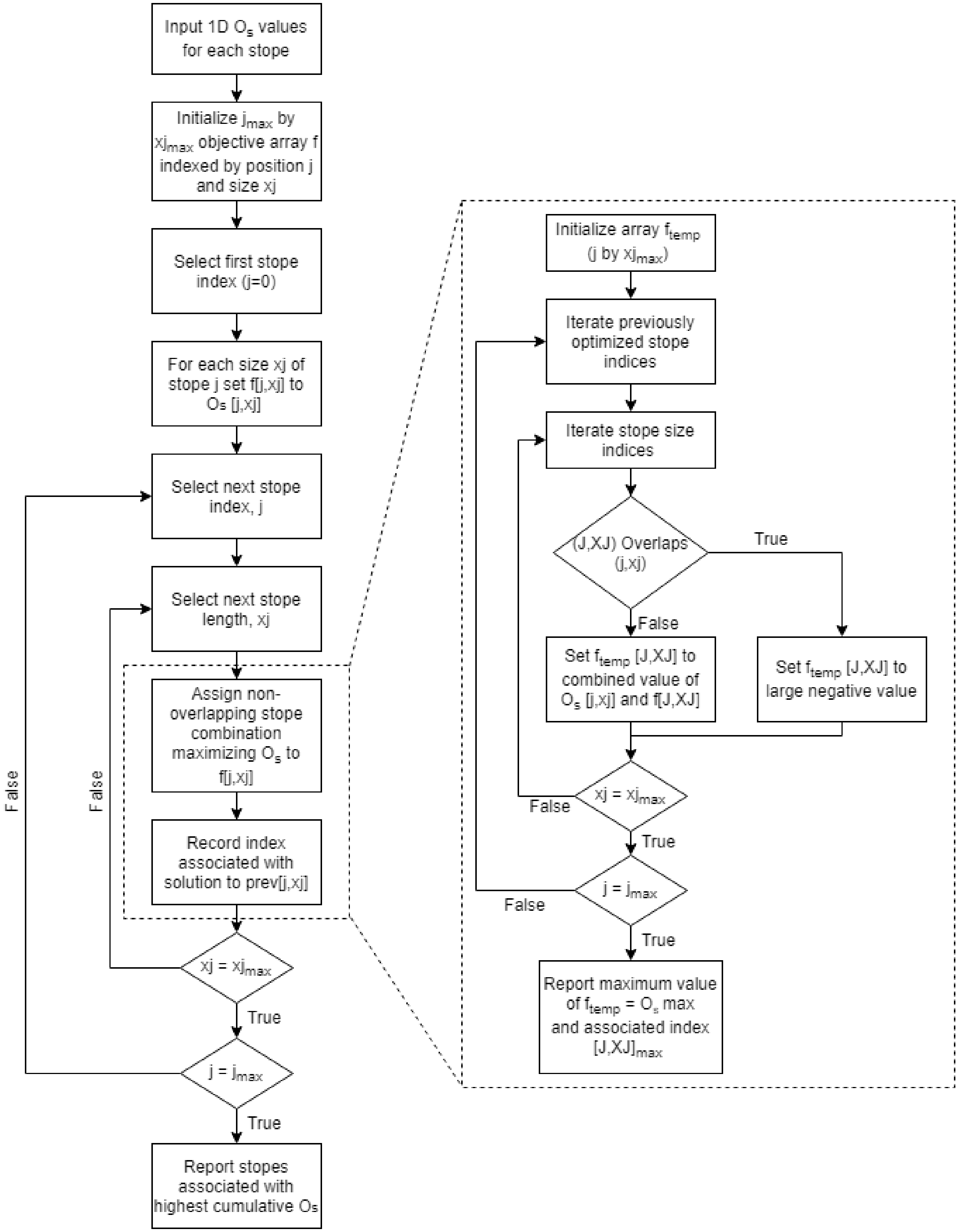

Input 1D O_s values for each stope
Initialize j_{max} by xj_{max} objective array f indexed by position j and size xj
Select first stope index (j=0)
For each size xj of stope j set f[j,xj] to Os [j,xj]
Select next stope index, j
Select next stope length, xj
Assign non-overlapping stope combination maximizing O_s to f[j,xj]
Record index associated with solution to prev[j,xj]
$xj = xj_{max}$
True
$j = j_{max}$
True
Report stopes associated with highest cumulative Os
False
False
Initialize array f_{temp} (j by xj_{max})
Iterate previously optimized stope indices
Iterate stope size indices
(J,XJ) Overlaps (j,xj)
True
False
Set f_{temp} [J,XJ] to combined value of O_s [j,xj] and f[J,XJ]
Set f_{temp} [J,XJ] to large negative value
$xj = xj_{max}$
False
True
$j = j_{max}$
False
True
Report maximum value of f_{temp} = O_s max and associated index $[J,XJ]_{max}$

Fig. 3.9: Flow chart of dynamic strip optimization

3.3.4 Greedy Strip Selection

To scale the one-dimensionally optimal strips into levels, they are combined with a greedy algorithm. At each decision, the strip which provides the best incremental improvement to the objective function value is selected. Between selections, previously optimized strips are evaluated for overlap with existing stopes, accounting for any required pillars. If the next greedily selected strip includes an overlapping stope, it is reoptimized, and the greedy selection is reassessed. The process is shown in Fig. 3.10.

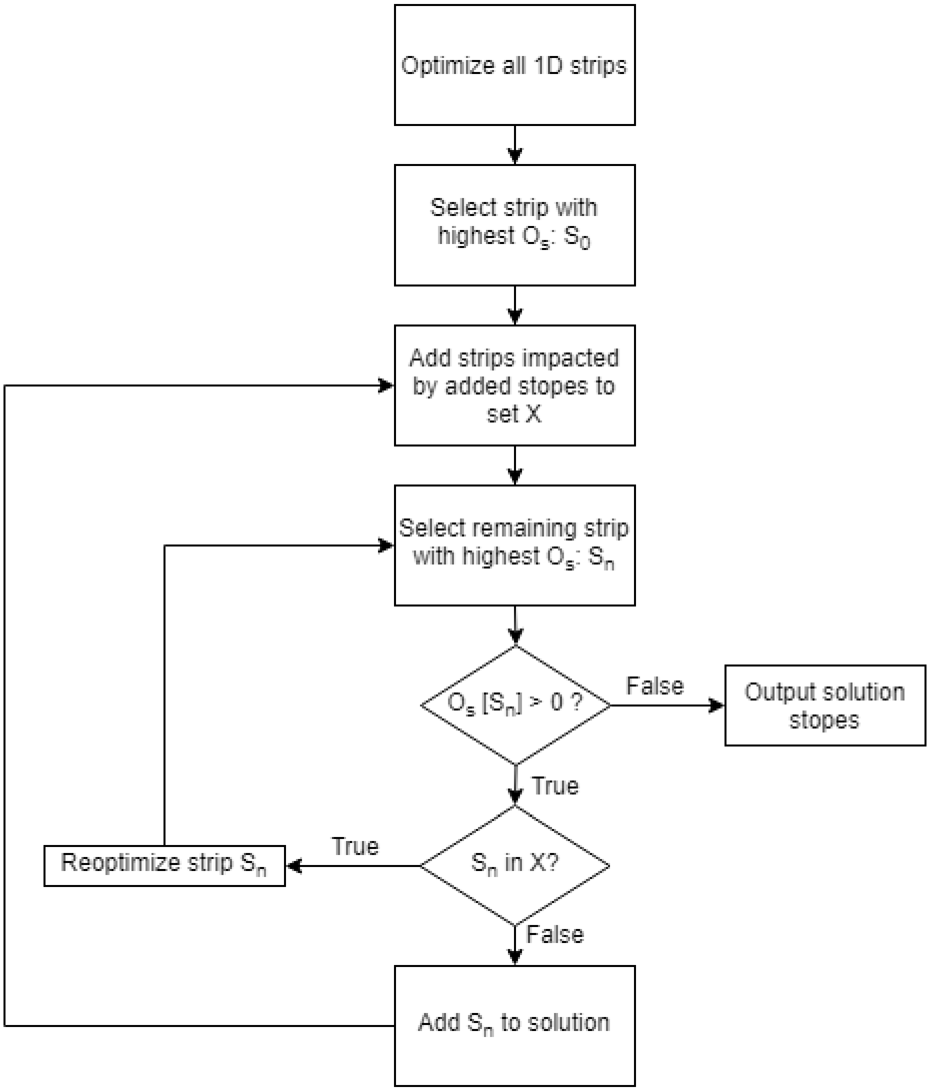


Fig. 3.10: Flow chart of level optimization through greedy strip selection

3.3.5 Dynamic Level Selection

The final step of the algorithm is the combination of 2-dimensional results into a three-dimensional solution. This process reuses the strip optimization function, inputting value data from each level in place of the stopes, with the stope length replaced by level height and z pillars input as required. This produces unified levels with constant sill pillars between them.

3.3.6 Algorithm Summary

The intent of this algorithm is to produce optimal layout plans for stoping operations considering risk over all realizations while respecting physical and conventional constraints. Reducing the problem to a combination of 1-dimensional sub-problems allows for efficient exploration of the solution space at the expense of ignoring some complex layouts. The heuristic nature of the algorithm leads to a gap in optimality which is accepted in exchange for its ability and efficiency in finding solutions. Section 3.4 introduces optional algorithm modules that both look to close this gap and allow for optimization with alternate constraint configurations.

3.4 Optional and Alternate Algorithm Modules

3.4.1 Alternative (Dynamic) Strip Selection

If a regular stoping method with permanent, continuous pillars is desired, the greedy algorithm is replaced with a second dynamic algorithm. In this case, the strip optimization algorithm is used to optimize strips in one dimension, parallel to either the x or y axis with pillars between stopes as usual. The one-dimensional stope optimization algorithm is then called again with the strip results as input to dynamically determine the best combination of strips. This configuration of the algorithm is appropriate where stopes are generally aligned with long, continuous longitudinal pillars between sets of stopes. This can be the case in lower stability conditions.

3.4.2 Tabu Solution Improvement

In order to improve the optimality of the heuristic optimization, a tabu search approach is implemented in the greedy strip selection algorithm. In the tabu implementation, the greedy algorithm is run multiple times with some strips disallowed from the solution. Multiple tabu lists are maintained to ensure a variety of solutions are explored. The lists of tabu selections are developed in two steps. First, a practitioner defined number, or tabu depth (T_d) of seeds are selected based on the top valued stopes in the initial solution. A series of runs are completed with each seed disallowed. The best stope in each run is then added to the tabu list for further runs with that seed. The solution from each run is evaluated and the best overall solution is retained. The process is depicted in Fig. 3.11.

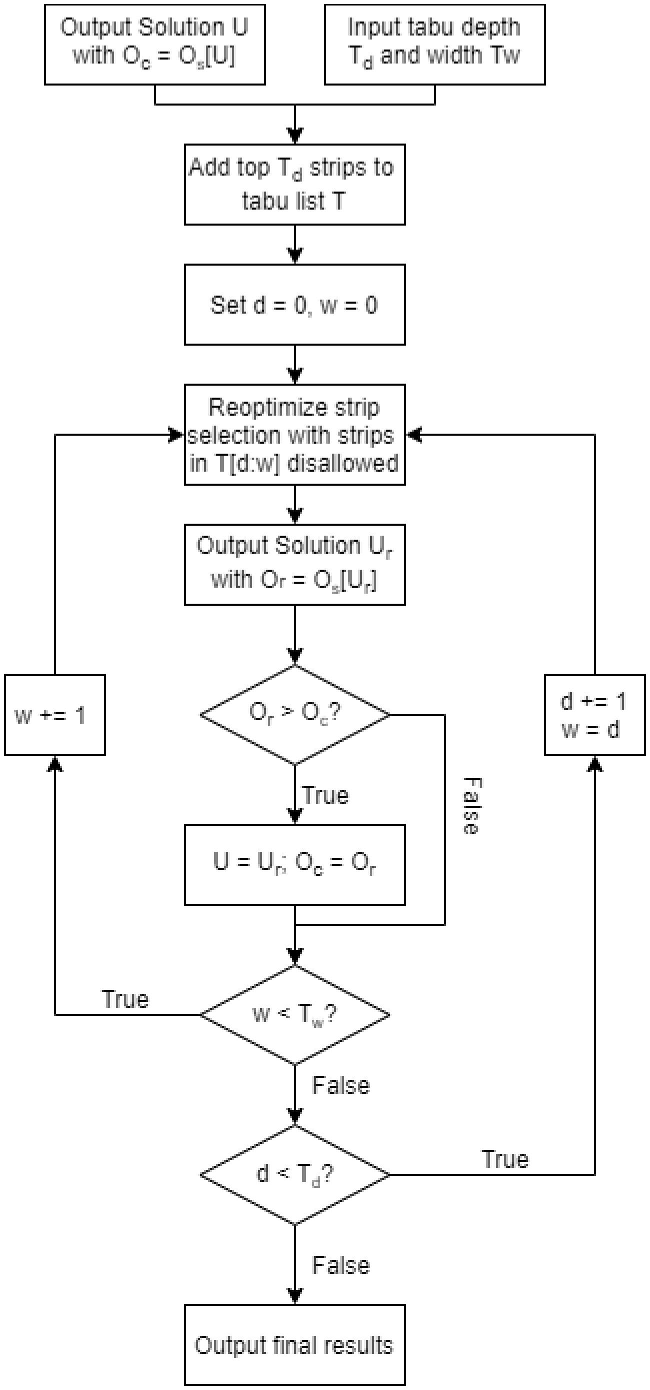

Output Solution U with $O_c = O_s[U]$
Input tabu depth T_d and width Tw
Add top T_d strips to tabu list T
Set d = 0, w = 0
Reoptimize strip selection with strips in T[d:w] disallowed
Output Solution U_r with $O_r = O_s[U_r]$
$O_r > O_c$?
True
False
$U = U_r$; $O_c = O_r$
w += 1
d += 1
w = d
$w < T_w$?
True
False
$d < T_d$?
True
False
Output final results

Fig. 3.11: Tabu algorithm flowchart

This solution is focused on the high-grade regions of the algorithm due to the value sorting of the selected strips. The greedy strip selection might get caught in a local maximum if only a single configuration is considered for extraction. The tabu approach reconfigures the extraction plan in the high-grade areas, possibly escaping minima where it will have the most impact. Fig. 3.12 shows how the choice of the T_d and T_w produce additional solutions by omitting strips from the solution which can outperform the initial case.

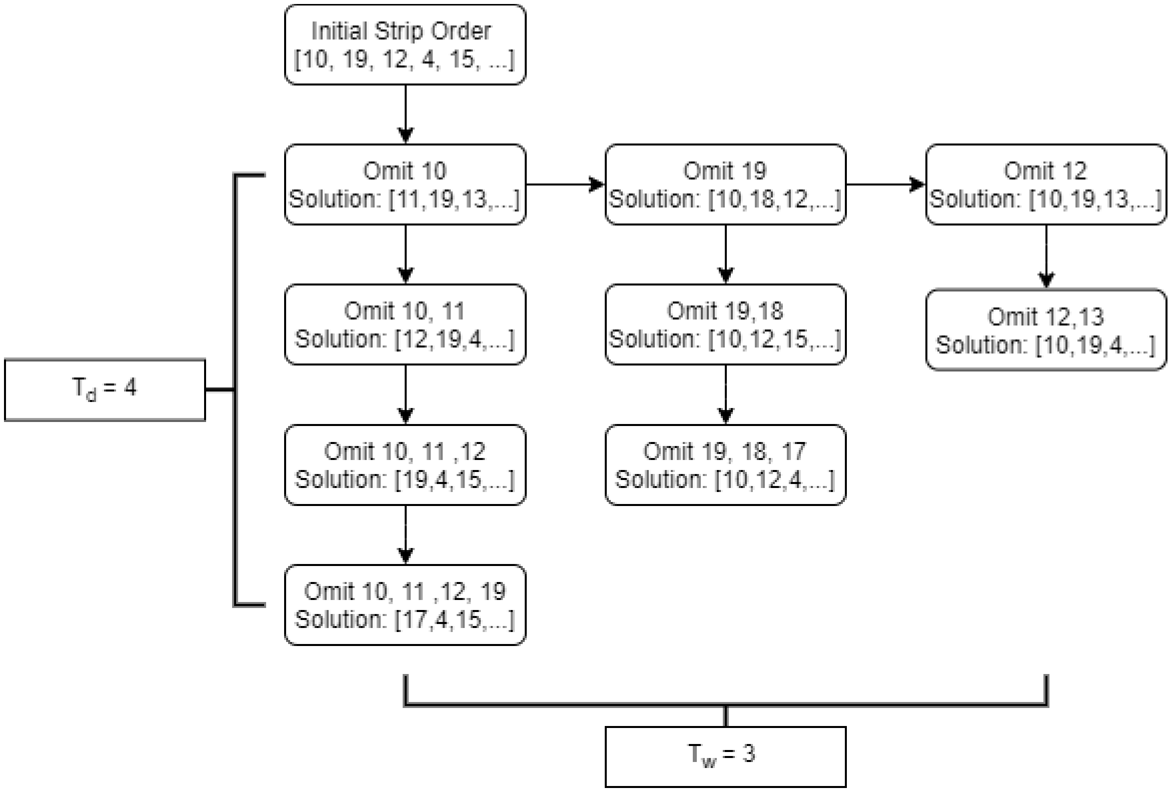


Fig. 3.12: Schematic showing impact of T_d and T_w selection. Each run generates an independent solution

Each set of omitted strips leads to a new solution that is compared to previous results. The best scenario is retained as the solution of the greedy portion of the algorithm.

3.5 Example Implementation of Algorithm

A simple example of the optimization algorithm is shown to illustrate its implementation. The portions of the algorithm which are common to all implementations are shown in Section 3.5.1. The optimization is first shown using the secondary greedy module in Section 3.5.2.1, while the secondary dynamic configuration is shown in Section 3.5.2.2. A comparison to the existing deterministic stope optimizer found in Maptec Vulcan is also produced to show the improvements due to accounting for non-linearities in a stochastic optimization framework and acting as a validation of the optimization results in Section 3.6.

The proposed stochastic algorithm is applied to the same set of simulated copper grade realizations in each configuration. The grade in the model is disseminated, with an average of 1.45% and a standard deviation of 0.23%. The origin of the 189000-block model is at the lower left-hand corner with an origin of (100,0,300) and extends to (550,600,1000). The blocks are 10x10x10 m. The grade distribution is shown in Fig. 3.13. Stochastic stope optimization is implemented for a single level of the model.

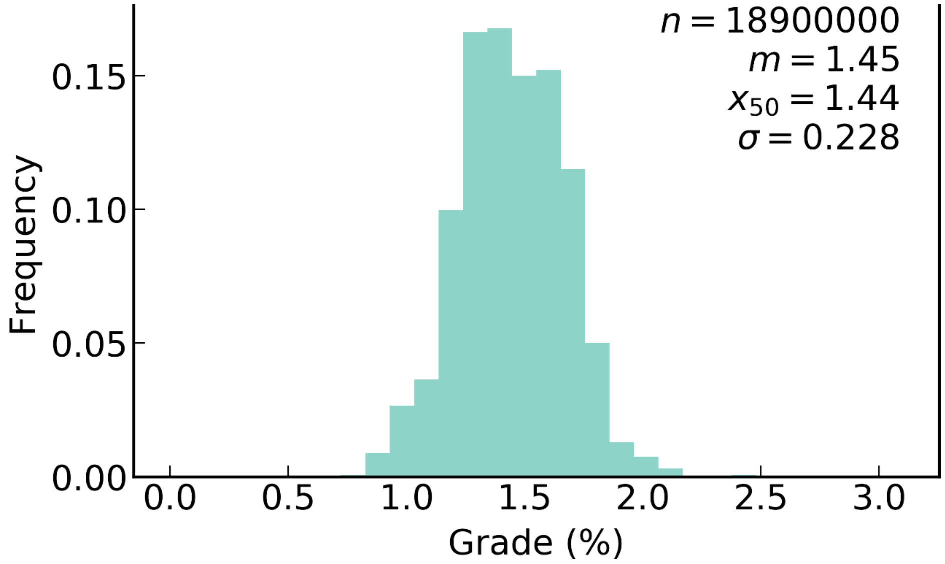


Fig. 3.13: Grade distribution histogram

Stochastic optimization approaches are anticipated to outperform deterministic methods even in cases where risk is not considered because the implemented processing recovery model is non-linear (Deutsch, 2018), applying the recovery by block before averaging to the stope scale is not equivalent to applying recovery on each stope. A slice of the e-type grade model is shown (Fig. 3.14) with two sample realizations (Fig. 3.15). Other pertinent values are assumed to facilitate the optimization (Tab. 3.2).

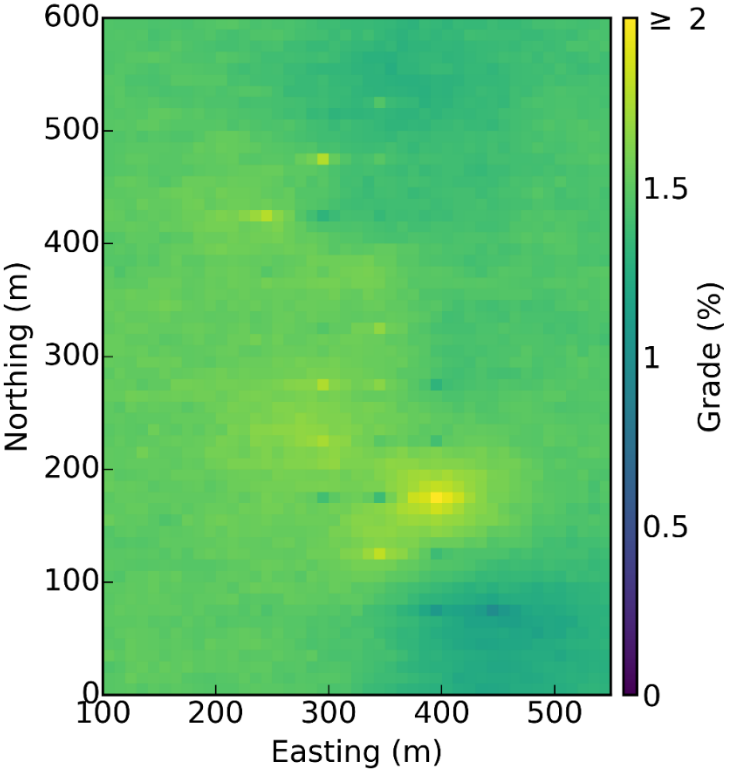


Fig. 3.14: Plan section of E-type model at z = 330 m

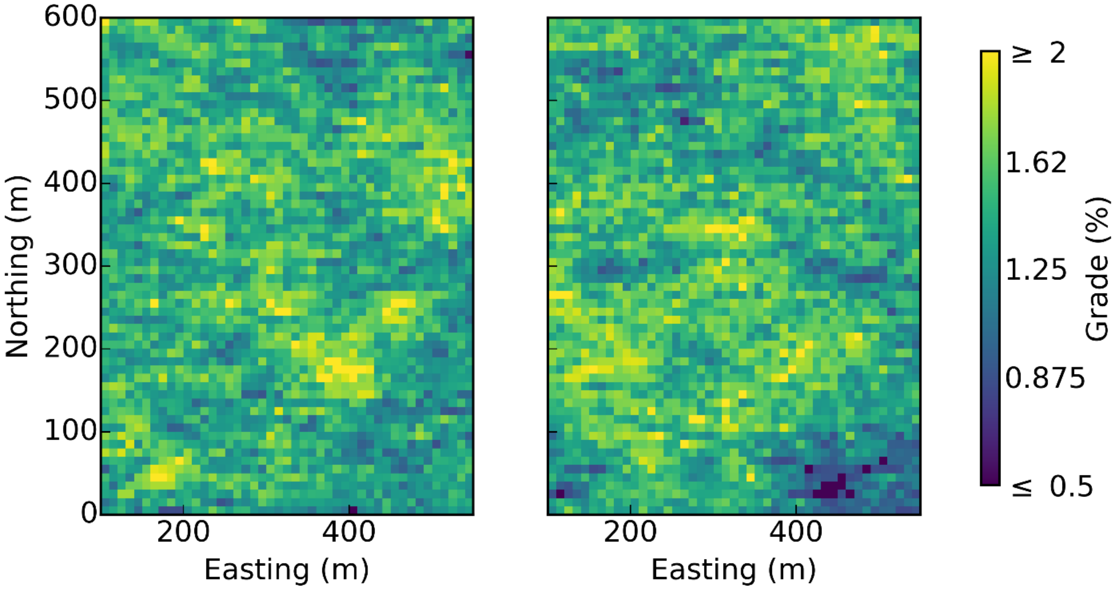


Fig. 3.15: Sample realizations from grade model. Plan section at z = 330 m

3.5.1 Parameters and Preprocessing

Optimization parameters are informed in part by the input data set and other values are defined or assumed to facilitate stope optimization. These values are reported in Tab. 3.2.

Tab. 3.2: Model parameters

Parameter	**Symbol(s)**	**Values(s)**
Stope Size Minima (blocks)	$xmin, ymin, zmin$	$3 \times 3 \times 3$
Stope Size Maxima (blocks)	$xmax, ymax, zmax$	$3 \times 3 \times 3$
Model Size	$Xdim, Ydim, Zdim$	$45 \times 60 \times 70$
Model Origin (m)	$Xmin, Ymin, Zmin$	(100,0,300)
Block Sizes (m)	Xsz, Ysz, Zsz	$10 \times 10 \times 10$
Risk Scaling Factor	a	0 to 0.9 step 0.1
Mining cost ($/block)	$mine$	$20k
Processing cost ($/block)	$proc$	$10k
Metal price	$price$	$2050/tonne
Density	$dens$	2.7
Recovery	Function	Eq. 13: Recovery Function
Pillar dimensions (blocks)	$Xpil, Ypil, Zpil$	1,1,1

Of note, the risk scaling factor is varied in this application of the algorithm to form an efficient frontier of results. The recovery function is defined piecewise with a penalty to recovery below a threshold value (Eq. 13).

$$r = \begin{cases} \left(\frac{g}{mg}\right)^2, & x \leq mg \\ 1 \quad , & x > mg \end{cases} \tag{13}$$

Where r is the processing recovery, g is the grade, and mg is the grade over which there is no penalty to recovery. Here, mg is set to 2.5%.

The data is preprocessed considering the chosen parameters to produce three arrays: stope value on each realization, average stope value over all realizations, and the standard deviation of stope values. These arrays are used to efficiently calculate the objective value for each stope, strip, and overall layout as required. In this implementation, recovery is applied to each possible stope (Eq. 13). Stope recoveries are calculated on each realization to account for non-linearities. When a deterministic model is used, the recovery can only be applied to the average stope value which is inappropriate (Deutsch, 2018). This either results in an over or underestimation of recovery and a correspondingly inaccurate value. Optimization considering the incorrect recovery information cannot produce an accurate layout.

3.5.2 Optimization

The optimization is completed for level 15 of the input data. Greedy and dynamic strip selection are considered in Sections 3.5.2.1 and 3.5.2.2, respectively. The resulting layouts are compared to show how the constraints impact the alignment of stopes and output values.

The 1-dimensional strip optimization is the same for both algorithm configurations, with all strips optimized using the dynamic algorithm outlined in Section 3.3.3.

3.5.2.1 Greedy Implementation

For the greedy implementation of strip selection, the strip with the maximum objective value is selected at each stage, as shown in Section 3.3.4. The progression of the solution is shown in Fig. 3.16.

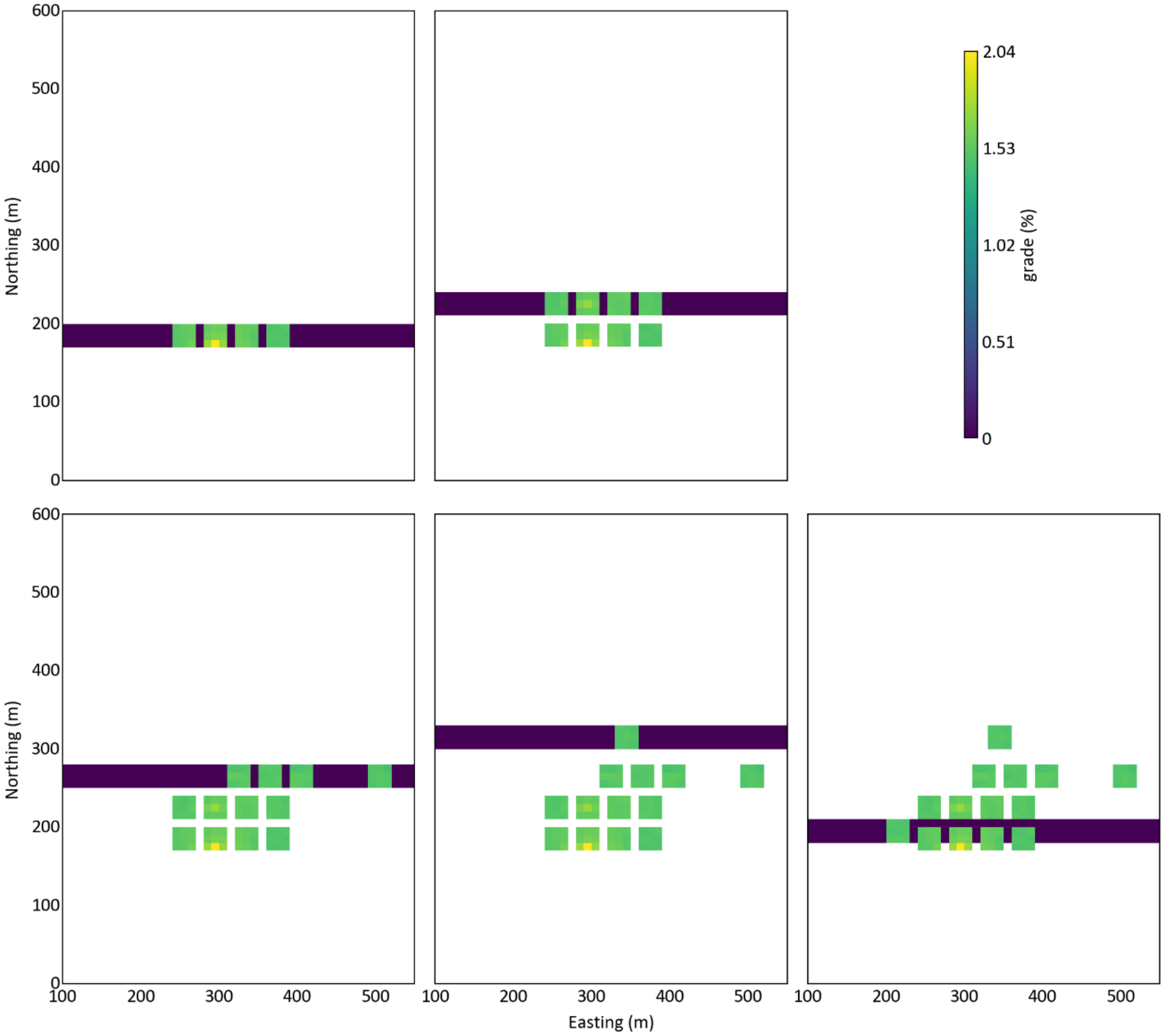


Fig. 3.16: Strip selection process. Selected strip shown in blue

In this case, all selected strips were parallel to the x axis. This process is repeated for each α value, further limiting risk in each layout. The resulting layouts are shown in Figs. 3.17 through 3.26.

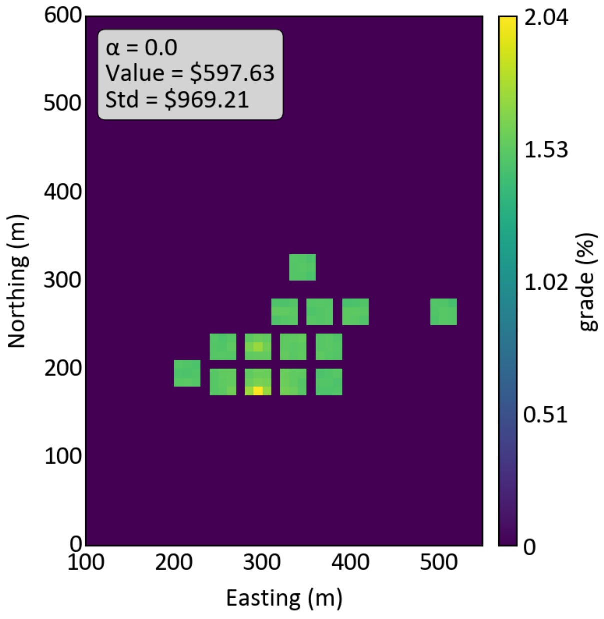


Fig. 3.17: Stopes at $\alpha = 0$

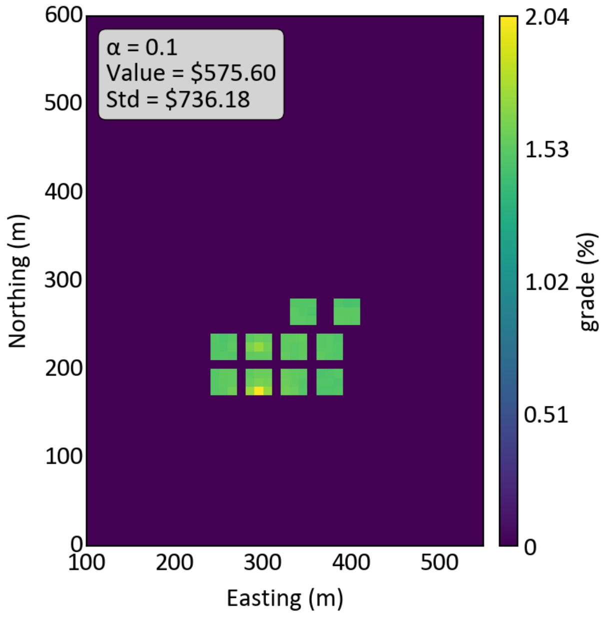


Fig. 3.18:Stopes at $\alpha = 0.1$

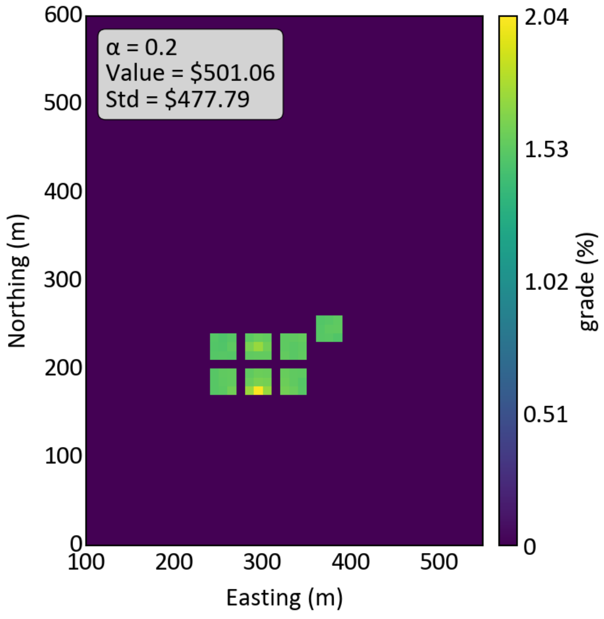


Fig. 3.19: Stopes at $\alpha = 0.2$

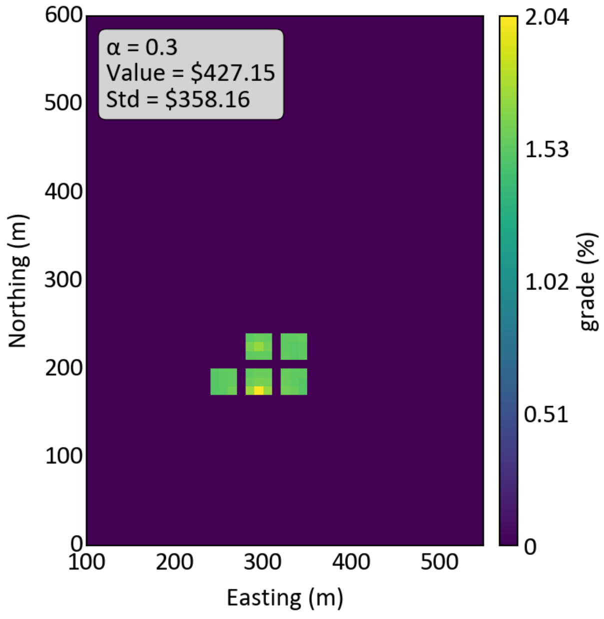


Fig. 3.20: Stopes at $\alpha = 0.3$

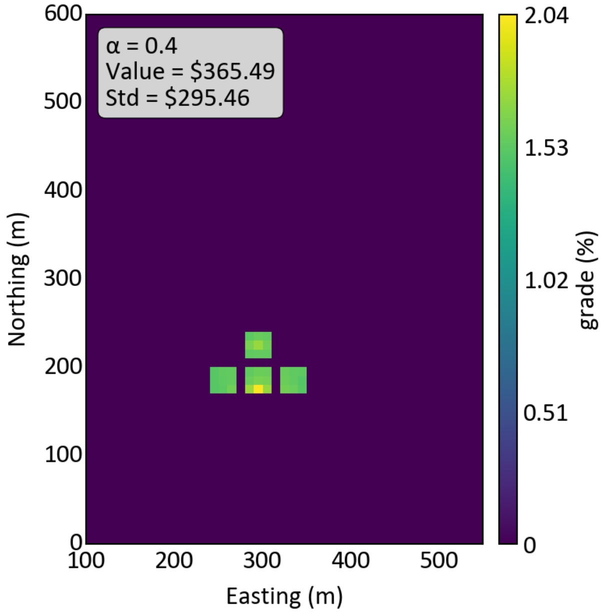


Fig. 3.21: Stopes at $\alpha = 0.4$

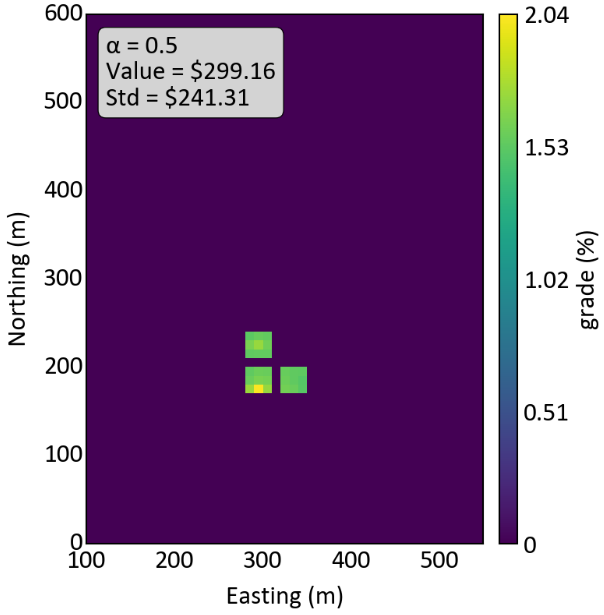


Fig. 3.22: Stopes at $\alpha = 0.5$

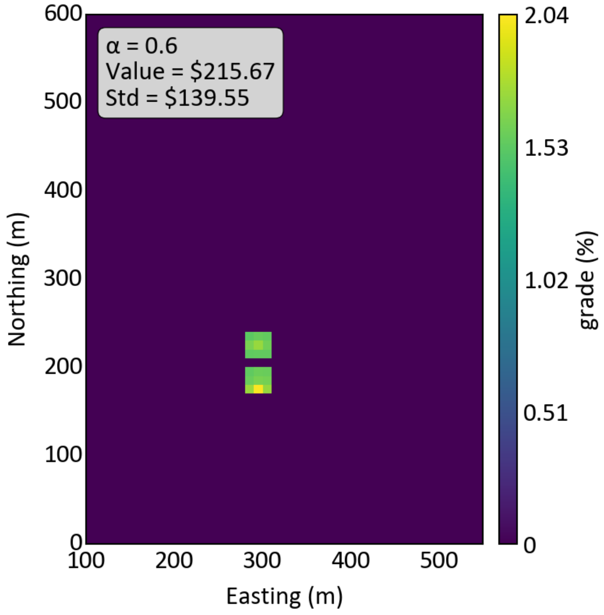


Fig. 3.23: Stopes at $\alpha = 0.6$

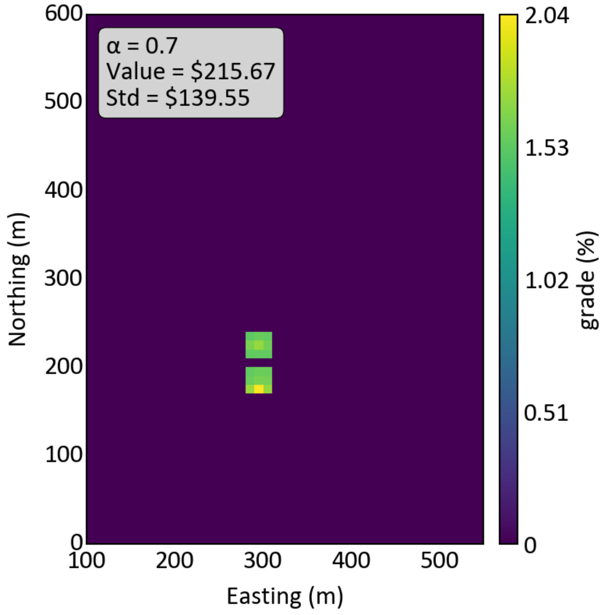


Fig. 3.24: Stopes at $\alpha = 0.7$

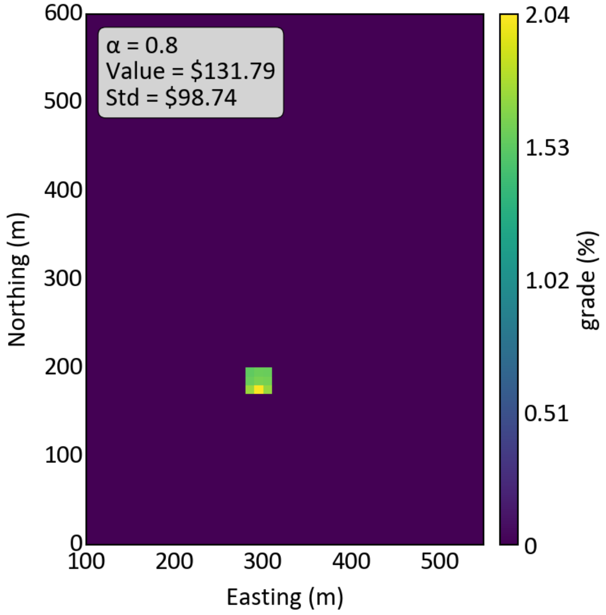


Fig. 3.25: Stopes at $\alpha = 0.8$

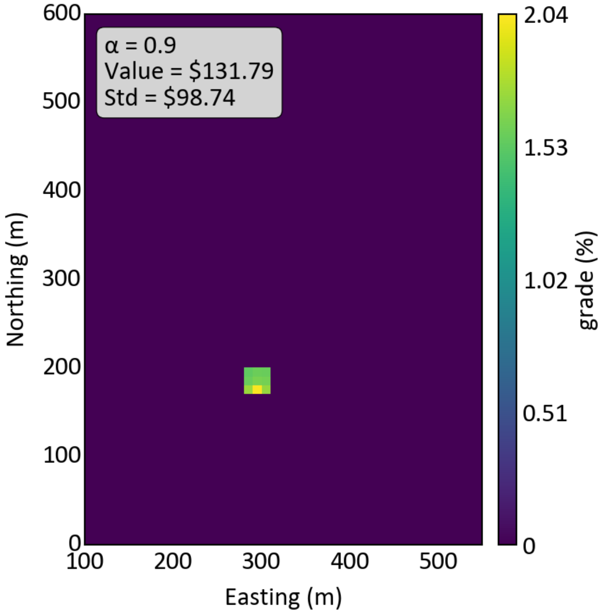


Fig. 3.26: Stopes at $\alpha = 0.9$

Higher risk discounts result in fewer planned stopes, beginning at 14 stopes with no risk discount and reducing to a single stope at $\alpha = 0.8$ and 0.9. Fig. 3.27 summarizes the number and total value of stopes at each risk discount. Optimization results are identical between some risk discounts. Each time the risk discount is increased, any stope with a marginally positive objective value or high uncertainty may become negative and be removed from the solution.

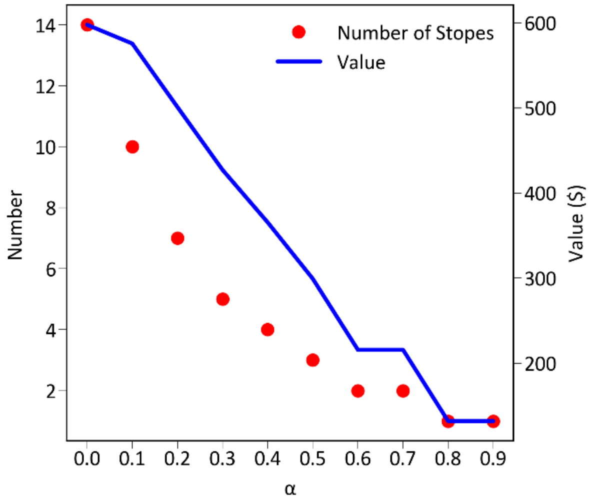


Fig. 3.27: Stope numbers and values

The objective values of the solutions are compared in value-risk space (Fig. 3.28). This forms an efficient frontier of stope layouts, which can be compared and selected based on a considered risk tolerance (Markowitz, 1952).

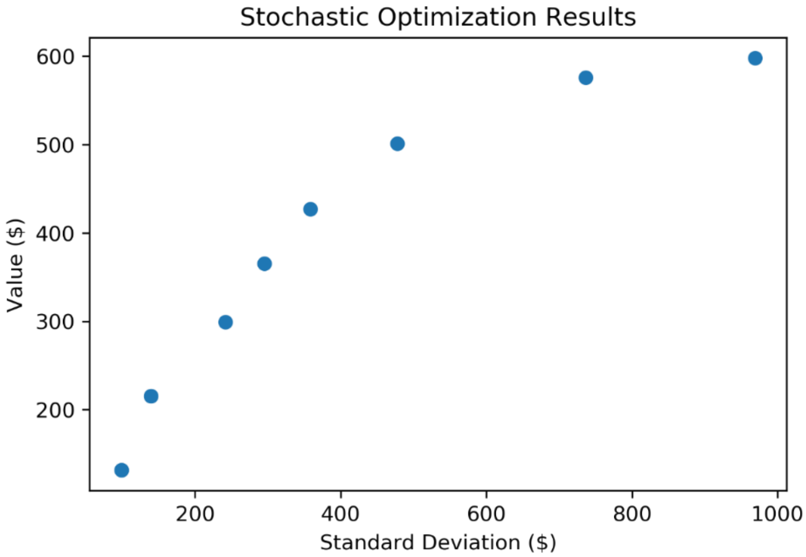


Fig. 3.28: Optimization results for all α values in value-risk space

3.5.2.2 Dynamic Implementation

The same input is considered for the dynamic implementation of the algorithm, but only strips parallel to the x or y axis are considered at one time. Strips parallel to the x axis are considered for this example. The stope layouts resulting from the dynamic strip optimization are presented in Figs. 3.29 through 3.38. This configuration selects the best combination of non-overlapping strips, as described in Section 3.4.1.

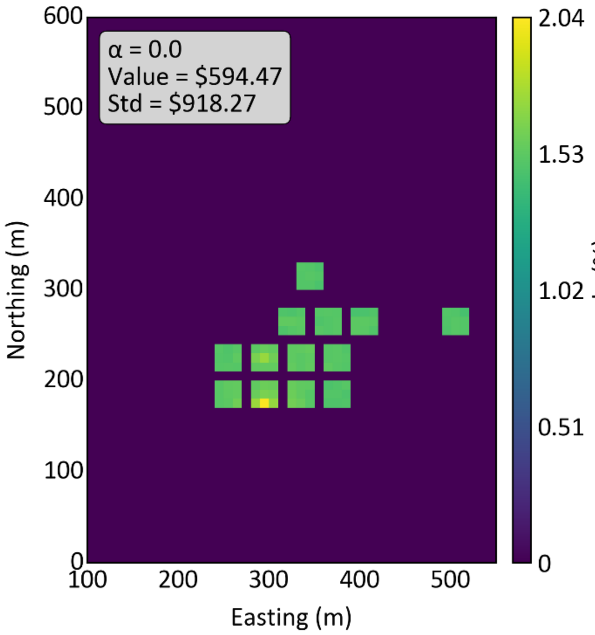


Fig. 3.29: Stopes at $\alpha = 0$

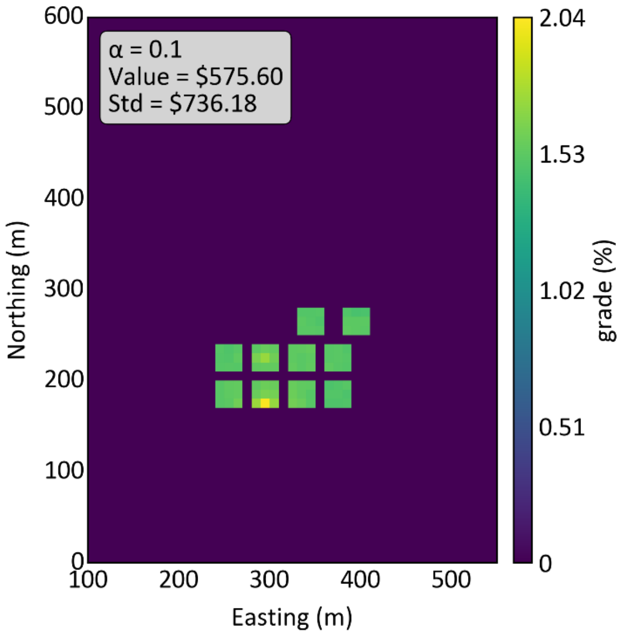


Fig. 3.30:Stopes at $\alpha = 0.1$

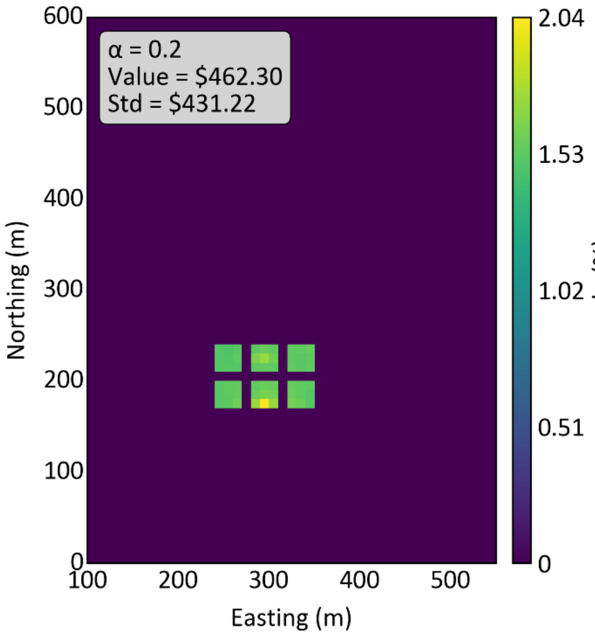


Fig. 3.31: Stopes at $\alpha = 0.2$

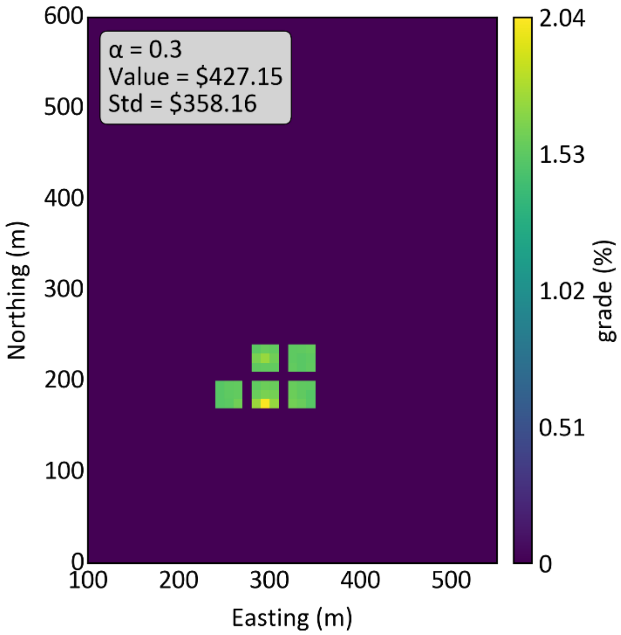


Fig. 3.32: Stopes at $\alpha = 0.3$

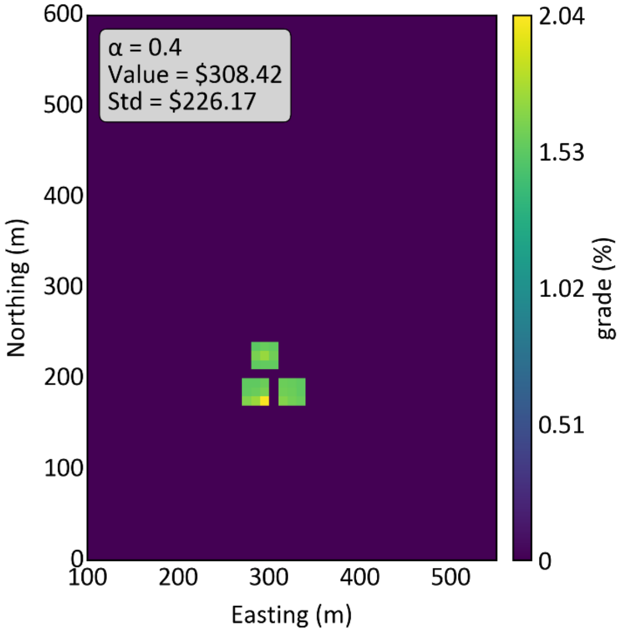


Fig. 3.33: Stopes at $\alpha = 0.4$

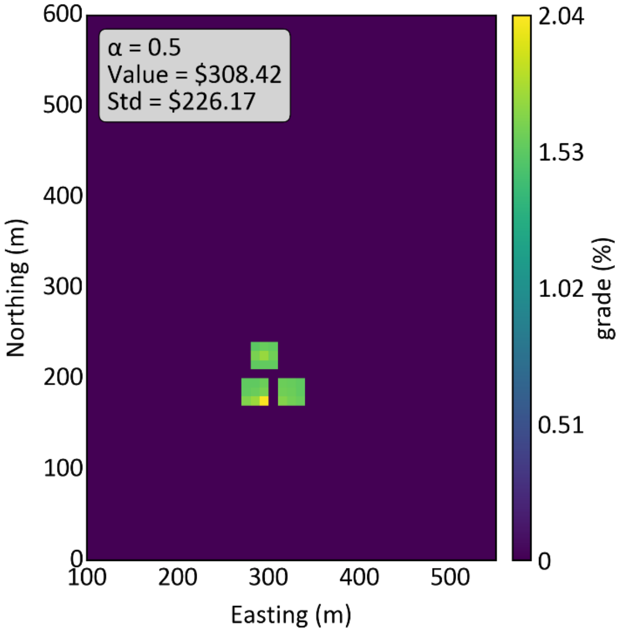


Fig. 3.34: Stopes at $\alpha = 0.5$

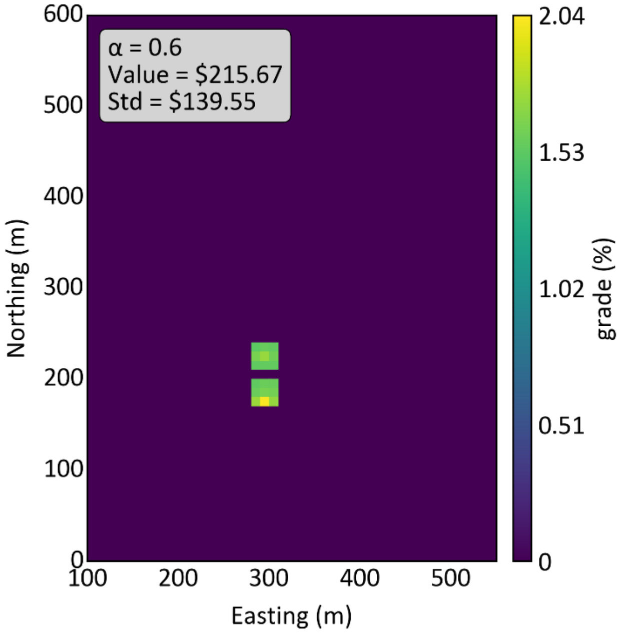


Fig. 3.35: Stopes at $\alpha = 0.6$

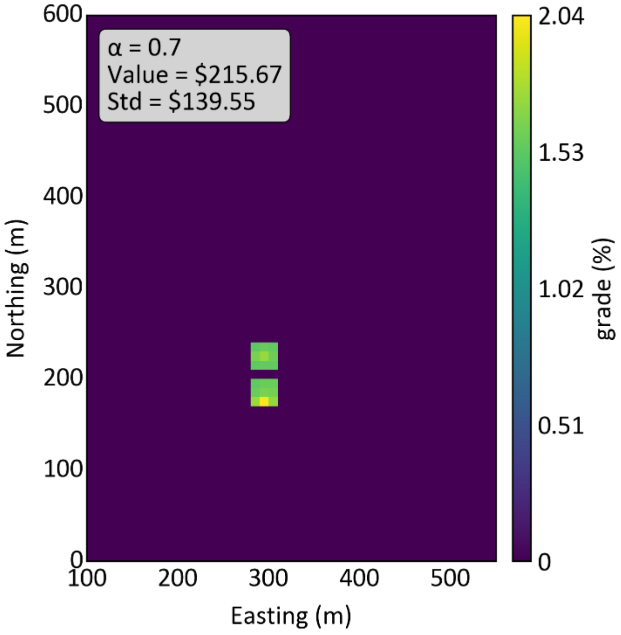


Fig. 3.36: Stopes at $\alpha = 0.7$

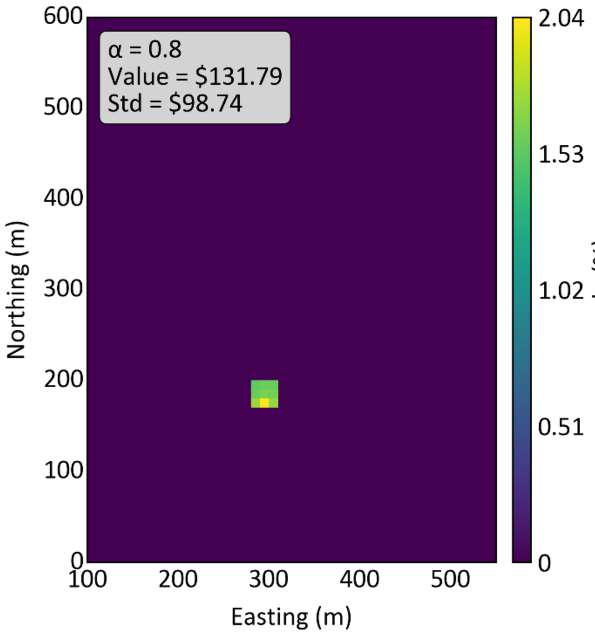


Fig. 3.37: Stopes at $\alpha = 0.8$

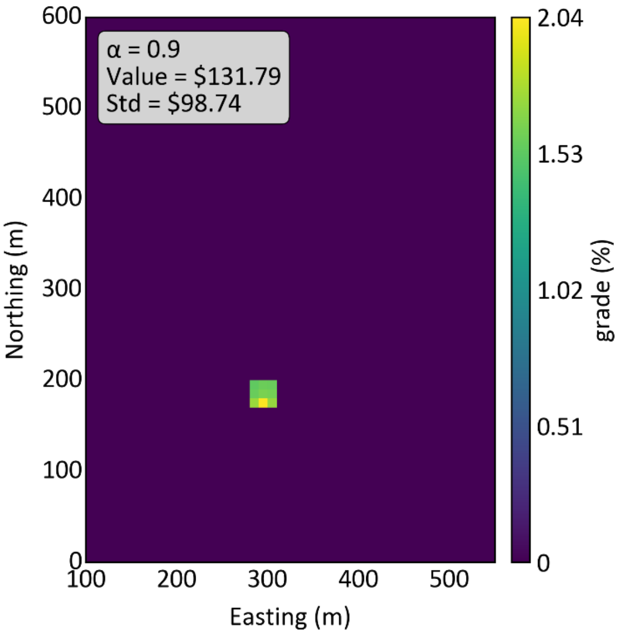


Fig. 3.38: Stopes at $\alpha = 0.9$

The optimization results are similar to those from the greedy strip selection in Section 3.5.2.2. Differences occur with α values of 0.0, 0.2, 0.4, and 0.5. These differences appear to occur for two reasons: 1) a strip is selected in the greedy configuration that overlaps with higher value strips and is disallowed by the dynamic algorithm; 2) a strip selected by the greedy algorithm is in the direction not considered in the dynamic configuration. The first difference will always result in a lower value in the dynamic configuration as valuable stopes are simply eliminated from consideration. The second difference, however, can cause either better or worse results, as seen in the layouts at $\alpha = 0.4$ and 0.5. This is because the greedy heuristic does not guarantee optimality, meaning that what would be considered a suboptimal choice by the greedy algorithm might be better when considered in combination with other choices.

The number and total values of stopes found using the dynamic configuration of the algorithm are plotted in Fig. 3.39. Similarly to the greedy implementation, higher risk discounts result in fewer planned stopes, beginning at 13 stopes with no risk discount and reducing to a single stope at $\alpha = 0.8$ and 0.9. The objective values of the solutions are also shown in value-risk space (Fig. 3.40).

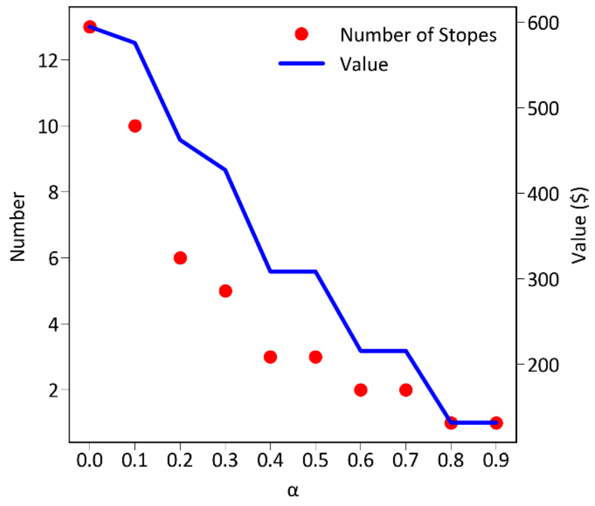


Fig. 3.39: Number of stopes and layout value by risk discount (α)

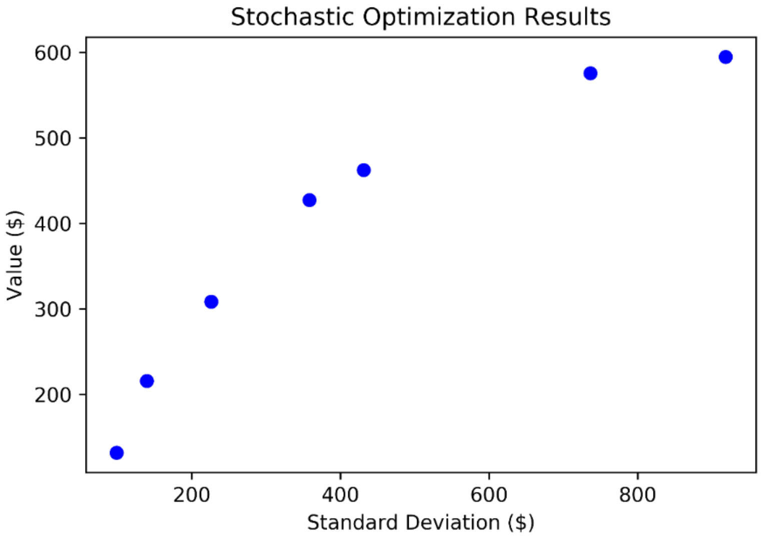


Fig. 3.40: Optimization results for all α values in value-risk space

The regularly distributed α values do not produce solutions across the entire efficient frontier. This is seen as the stope numbers changes between incremental α values (Fig. 3.39) and visible gaps in the frontier itself (Fig. 3.40). It is suggested that the frontier would be better informed if more discount values are considered; however, some discontinuity is expected because removing an individual stope from the solution has a step impact on solution value and risk. Depending on the criteria used to select an overall layout, these gaps may require refinement to ensure an optimal solution is found for a desired level of risk.

3.6 Comparative Evaluation

In order to evaluate and evaluate the algorithm, it is compared to the Stope Shape Optimizer (SSO) results as implemented in Maptek Vulcan (Alford, 2013; *Vulcan Envisage*, 2018). The SSO is deterministic and uses the e-type model after recoveries are applied as input. A cutoff grade equivalent to a \$0 block is calculated in Eq. 14 and Eq. 15. The recovery is applied by block.

$$V = M \times p - C_t = t_o \times (r \times g) \times p - C_t \qquad 14$$

Where V is block value, M is the mass of the contained metal after recovery, P is the metal price, and C_t is the block mining cost. Decomposing M into terms yields $M = t_0 \times r \times g$, with t_o representing mined tonnes of ore, r the expected recovery after processing, and g the fractional grade. Substituting r for Eq. 13 and solving for grade g yields a cutoff grade, cg, where $V = 0$ (Eq. 15):

$$cg = \sqrt[3]{\frac{C_t \times mg^2}{t_o \times p}} \qquad 15$$

Recall that mg is the grade at maximum recovery from Eq. 13.

After substituting the model values in Eq. 15, a cutoff value of 1.50% is found. The SSO slice method stope optimization is then applied with a cutoff of 1.50%. The optimization is run in both YZ and XZ orientations to ensure the maximum value orientation is found. The resulting layouts are shown in Fig. 3.41 and Fig. 3.42.

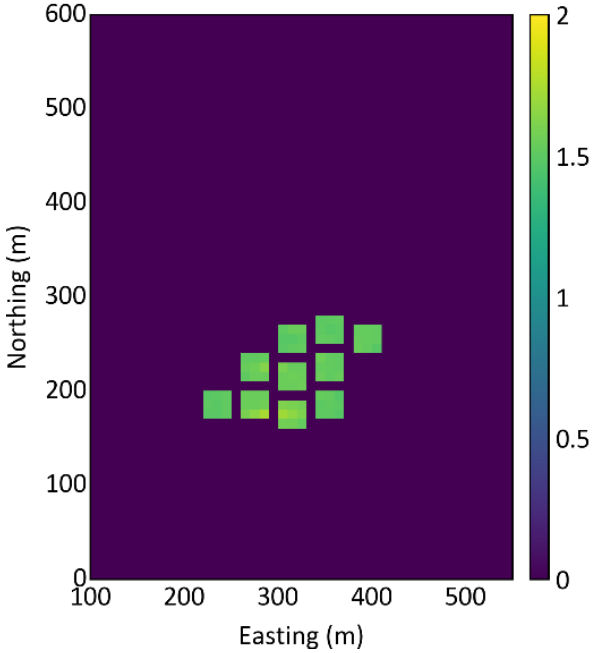


Fig. 3.41: SSO Solution with XZ slices

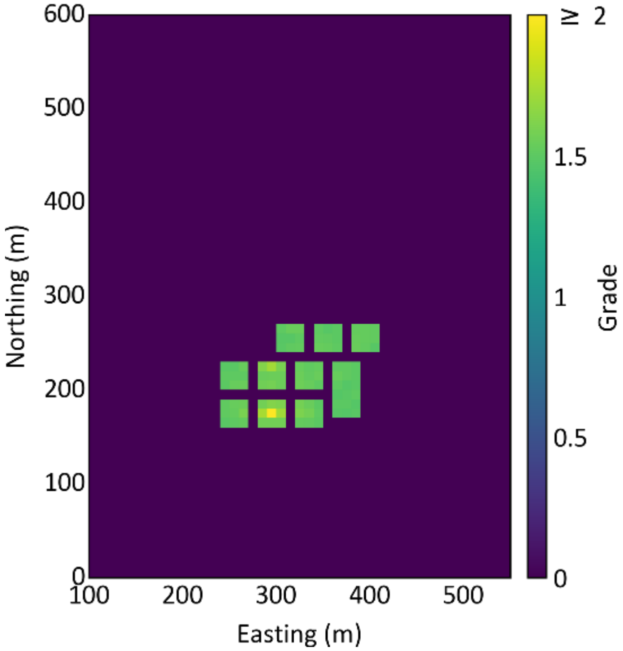


Fig. 3.42: SSO Solution with YZ slices

Note that the Vulcan YZ result does not appear to follow the input pillar requirements. The number of stopes in both cases is less than was found previously with $\alpha = 0$. The value of the YZ SSO implementation is similar to the stochastic implementation with 11 stopes in the produced layout instead of 13 (Section 3.5.2). The results of the Vulcan model SSO are shown in comparison to the proposed stochastic stope optimization results in value-risk space Fig. 3.43.

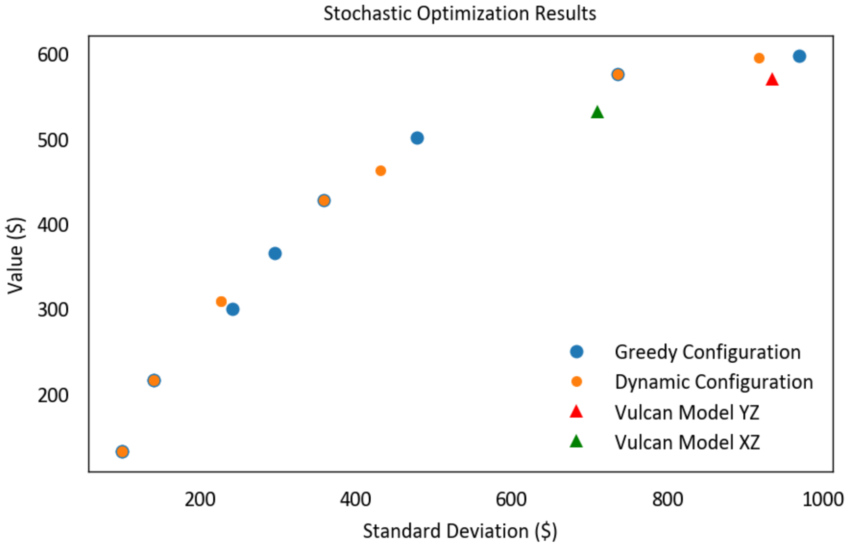


Fig. 3.43: Optimizer solutions in Greedy (Section 3.3.4) and Dynamic (Section 3.3.5) configurations compared to SSO results

The heuristic optimization outperforms the SSO due to the nonlinearities in metal recovery. The placement of the SSO results under the efficient frontier indicates that neither is the optimal solution with or without a consideration of risk. The greedy configuration of the algorithm is 4.90% above the Vulcan YZ results, while the dynamic configuration is 4.16% greater where no risk is considered. This significant improvement confirms the ability of the algorithm to find approximately optimal solutions considering the applied constraints.

3.6.1 Discussion

Adapting a deterministic heuristic stope optimization algorithm to account for stochastic data allows for the proper calculation of non-linear responses and the consideration of risk in an objective function. The combined dynamic-greedy algorithm, as presented by Nikbin et al. (2018), is adapted to consider a stochastic grade model. The algorithm is altered to output a stope layout rather than a design envelope and consider levels individually to better conform with stope mining conventions. Two configurations of the algorithm are developed with different strip selection processes: one greedy and one dynamic. The dynamic implementation is restricted to consider strips parallel to one axis rather than both as in the greedy approach, further limiting the explored solution space.

The dynamic configuration of the algorithm is a constrained version of the general greedy algorithm, meaning it should only be implemented where continuous longitudinal or transverse pillars are required for stability. In general, it should be expected that adding these constraints will lower the value of the solution. However, the heuristic nature of the greedy algorithm means an optimal solution is not guaranteed, and, in some cases, the dynamic implementation does produce a better result than the greedy.

The algorithm is compared to the SSO as available in Maptek Vulcan, showing that the stochastic optimization produces better results by 4.90% in the greedy configuration and 4.16%.

Chapter 4: Case Study

The stochastic stope optimizer is applied to a gold data set from a real-world project that is transitioning from a surface operation to underground mining. The data is modelled with sequential Gaussian simulation to create a suite of realizations (Section 4.1). The grade realizations are then inputted into the stochastic stope optimizer (Section 4.2).

4.1 Geostatistical Modeling

A stochastic grade model is produced according to geostatistical best practices to facilitate the application of the stochastic stope optimizer (Rossi & Deutsch, 2014). Twenty realizations are produced using the GSLIB program USGSIM (Manchuk & Deutsch, 2012). USGSIM generates sequential Gaussian simulation realizations, as described in Chapter 2. The raw data and subset to be modelled are described in Section 4.1.1. The results of the simulation are presented in Section 4.1.4, and they are checked in Section 4.1.5.

4.1.1 Data

The data is provided as a series of drillhole records arranged in the collar, survey, and assay tables. There are a total of 1557 drillhole collars in the data set with associated assay data. Three solids are provided with the data: the existing surface excavation as mined before the transition to underground mining, a low grade solid consisting of a relatively massive vein-like structure trending N 34.7° E and dipping approximately 50° to the SE, and an additional high grade solid. The high grade solid is fully contained within the low grade structure with a similar strike and dip (Figs. 4.1 - 4.3). The high-grade structure is selected for modelling and optimization, with subsetting considered to maintain a reasonable problem size.

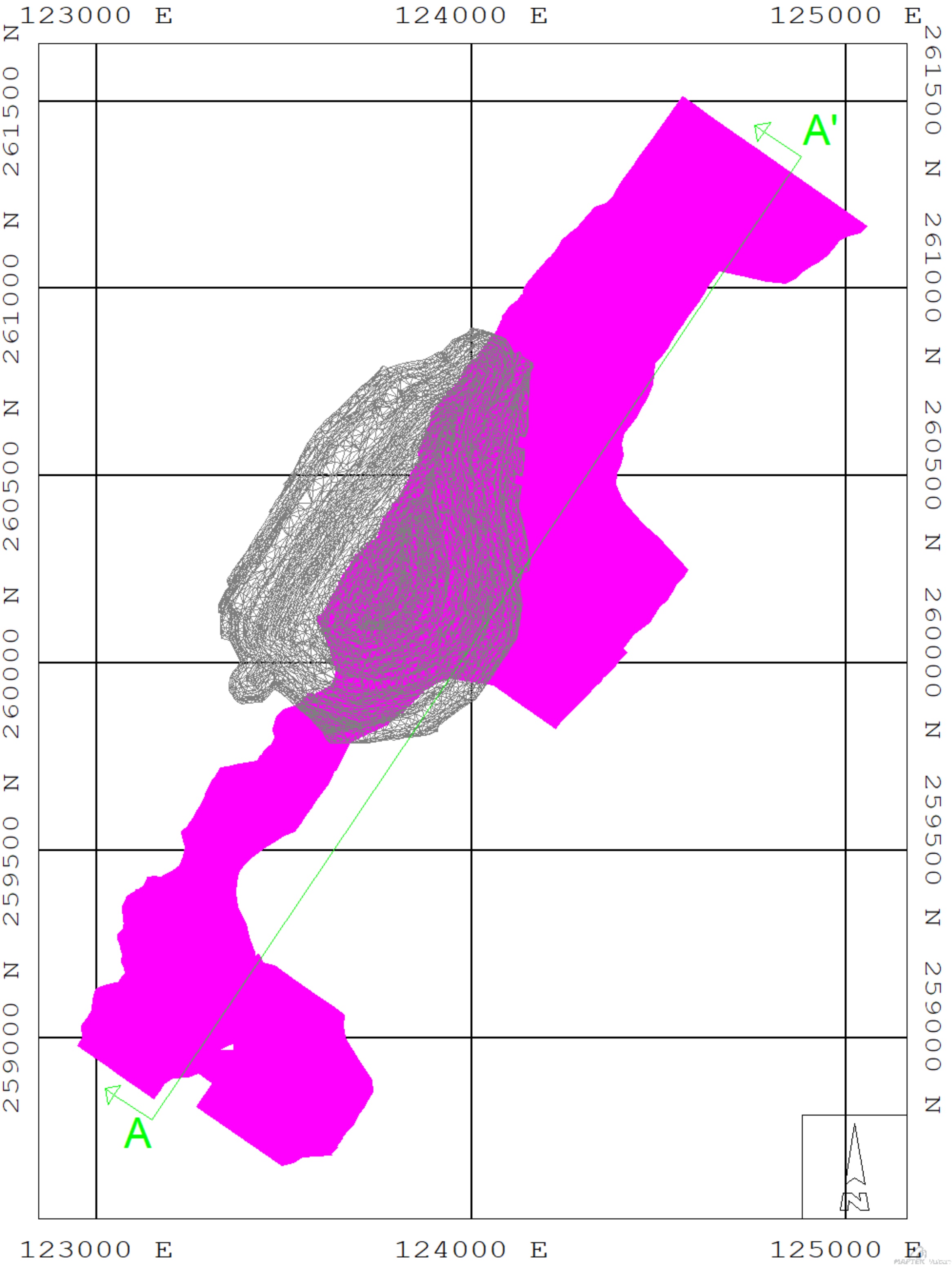


Fig. 4.1: Plan view projection of high-grade vein and existing excavation. Line A-A' shows long section view direction. Grid is in meters

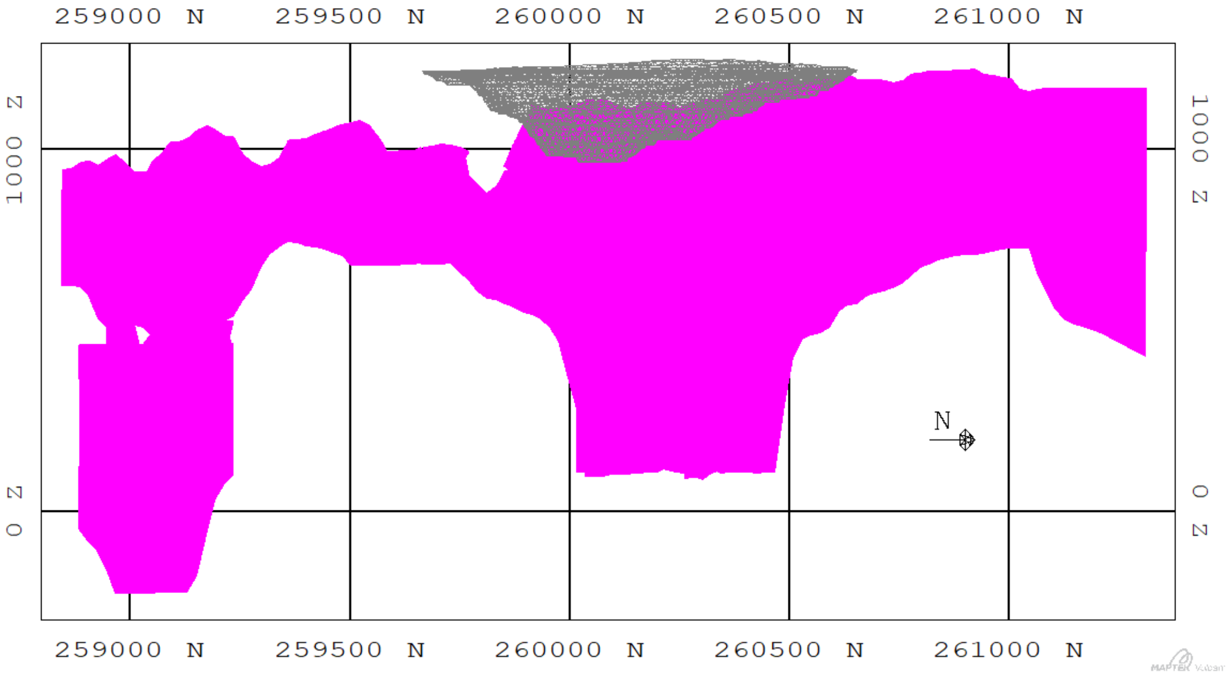


Fig. 4.2: Long section view with full high-grade vein and existing excavation from line A-A' (Fig. 4.1). Grid is in meters

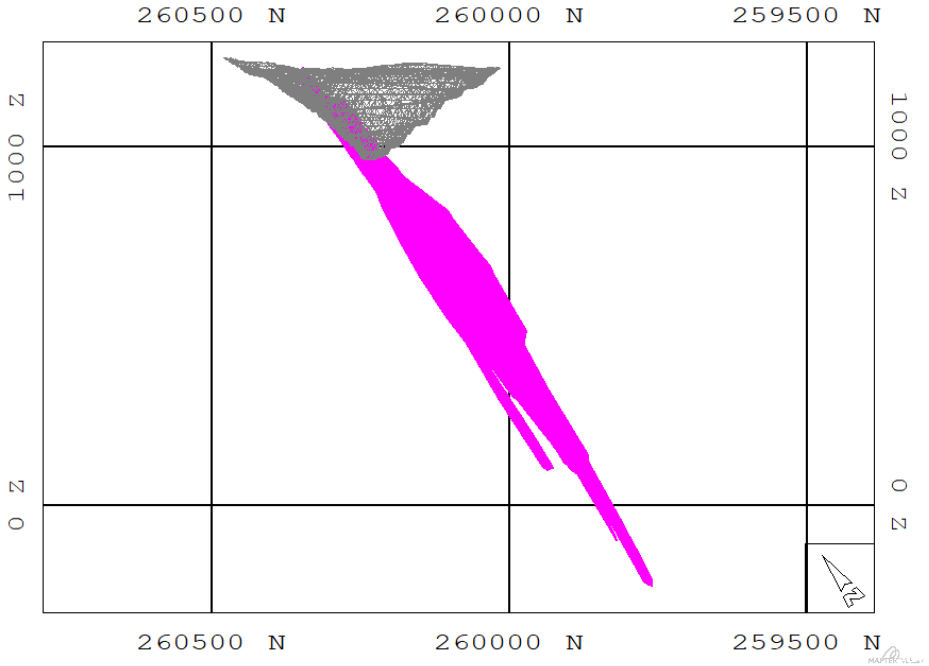


Fig. 4.3: Section view of high-grade structure perpendicular to line A- A' (Fig. 4.1). Grid is in meters

The high-grade structure is 2.5 km in length and dips at about 50°. A subset of the high-grade vein is selected for modelling and optimization (Fig. 4.4). This region is selected to lie in an area of interest below the existing surface excavation and extends to the lower extents of the high-grade data in this location at an elevation of about 420 m. The deposit is very densely drilled on from the surface, with fan drilling delineating the high-grade target (Figs. 4.5, 4.6).

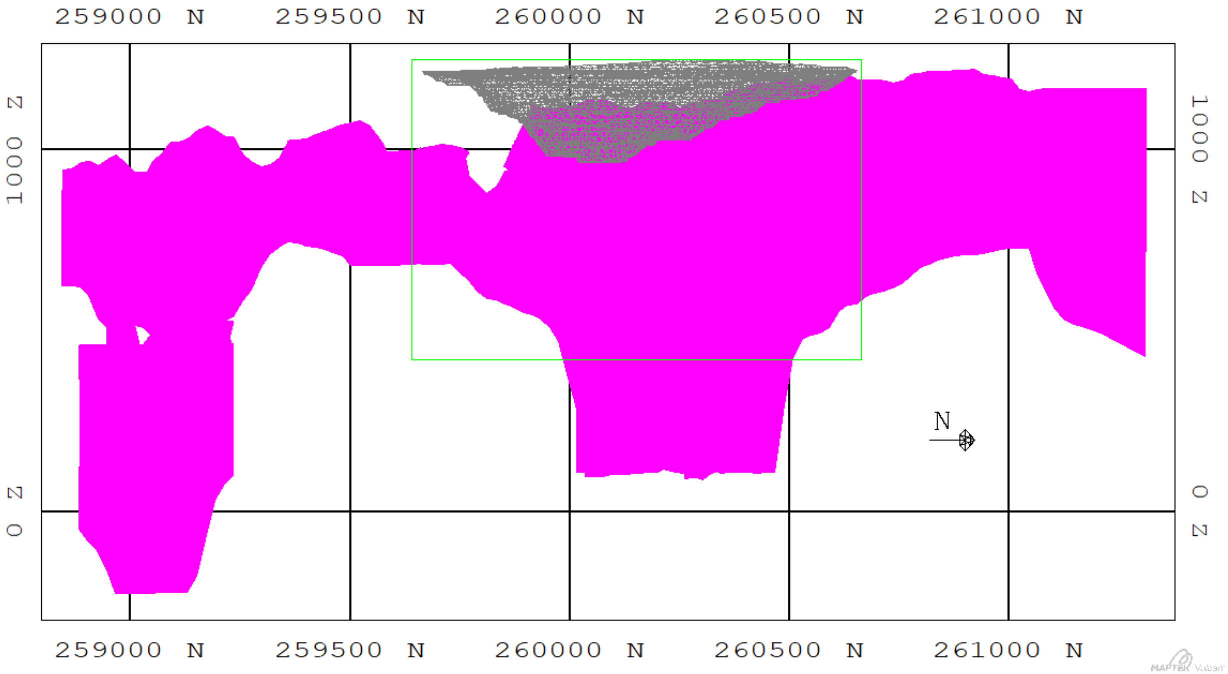


Fig. 4.4: Long section view projection with subset of high grade considered shown in green. Grid is in meters

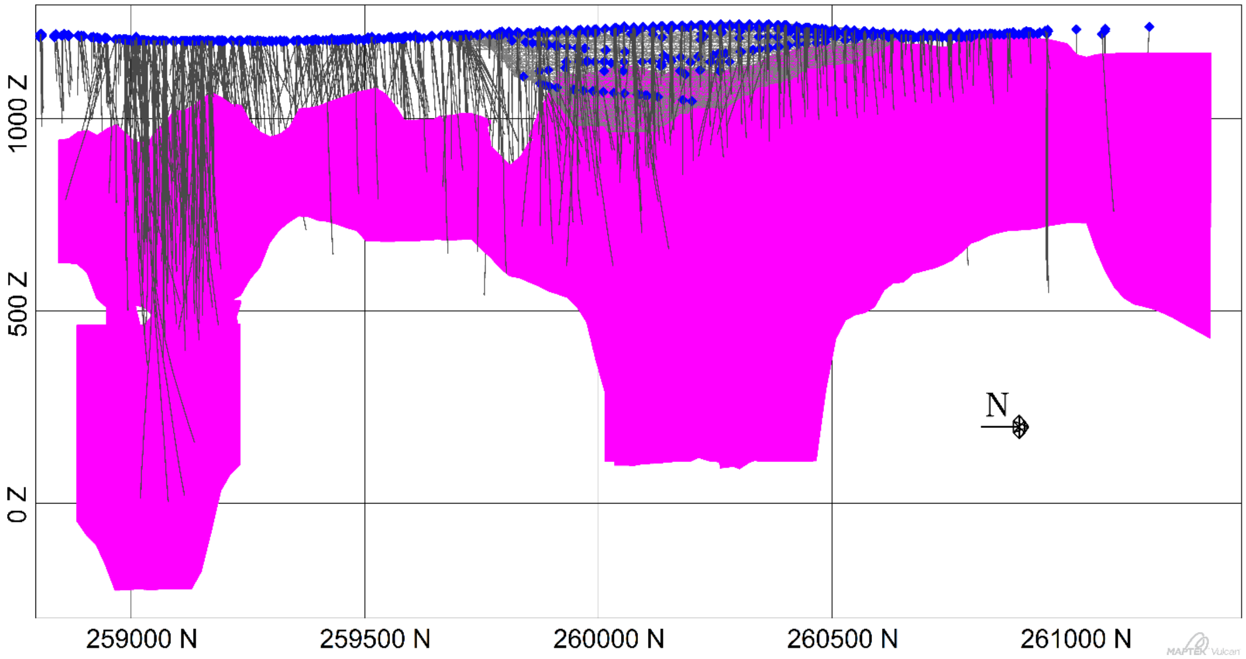


Fig. 4.5: Long section projection with drillhole collars and traces. Grid is in meters

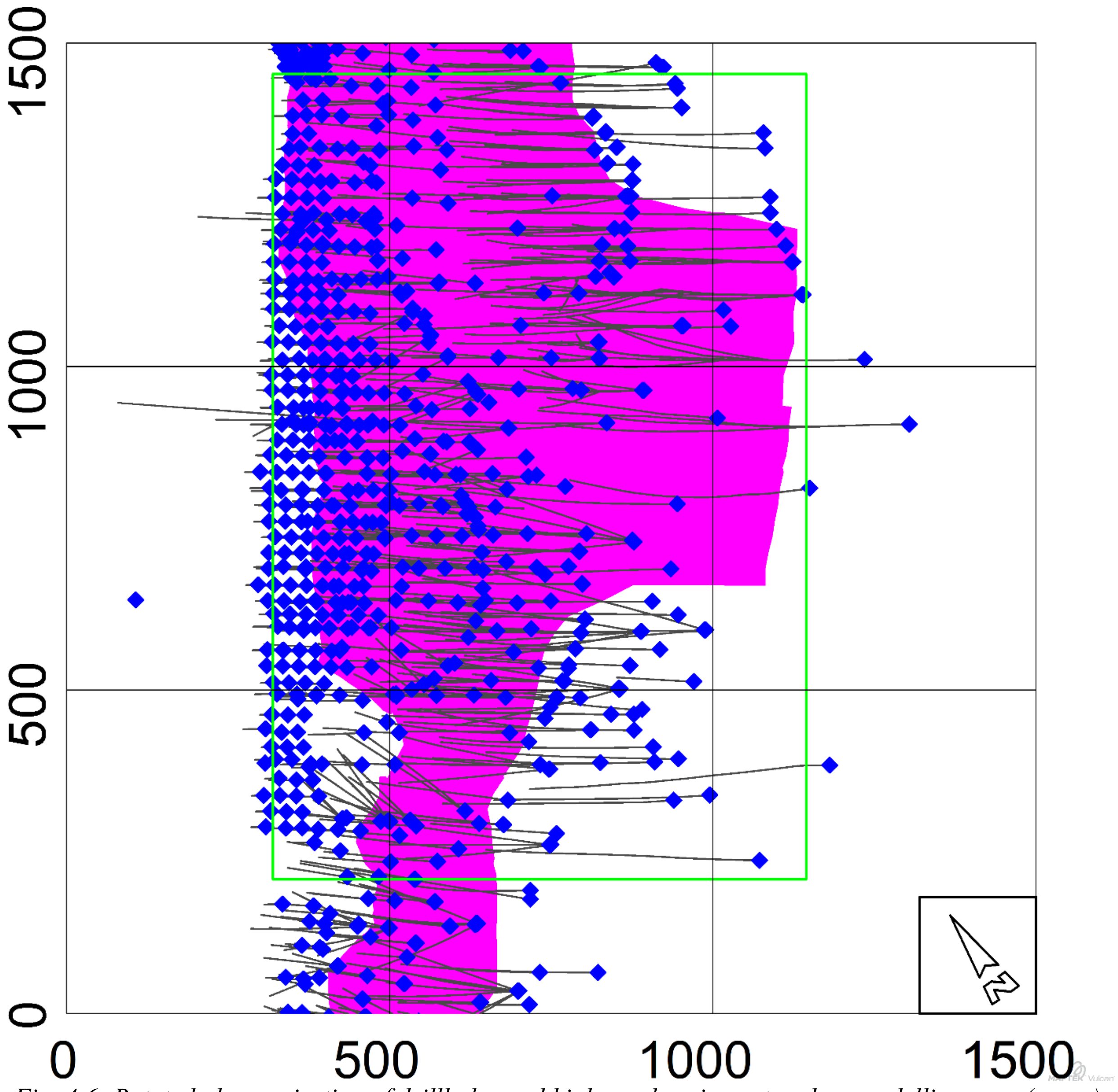


Fig. 4.6: Rotated plan projection of drillholes and high-grade vein centered on modelling area (green). Grid is in meters in rotated frame of reference

The drilling density is lower at depth, with drilling nearly perpendicular to the structure. The data is composited on 5 m lengths for assessment and modelling (Fig. 4.7). Note that the bottom of the region selected for modelling lies below the deepest sample within the high-grade structure. Section 4.1.2 explores the data distribution and declustering. Variography is completed in Section 4.1.3. Finally the simulation is completed in Section 4.1.4.

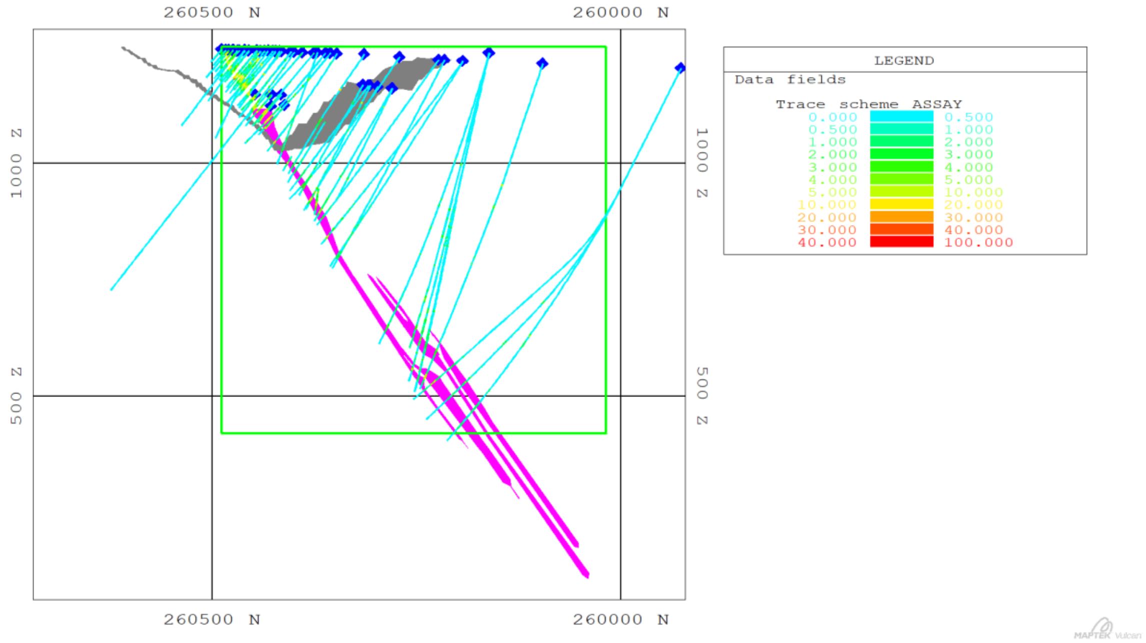


Fig. 4.7: Characteristic section view perpendicular to high grade structure showing drilling intercepts. Grid is in meters

4.1.2 Grade distribution and declustering

Subsurface sampling is biased by nature, primarily due to the clustering of samples (Rossi & Deutsch, 2014). There is a guarantee that samples along a drillhole are clustered and common that drillholes are clustered around higher-grade targets. Declustering is a necessary process to determine the true grade distribution that should be reproduced by simulation techniques (Rossi & Deutsch, 2014). The raw distribution of gold assay data that falls within the modelled area of the high grade structure is right-tailed with a mean of 3.79 g/t and a standard deviation of 3.08 g/t (Fig. 4.8).

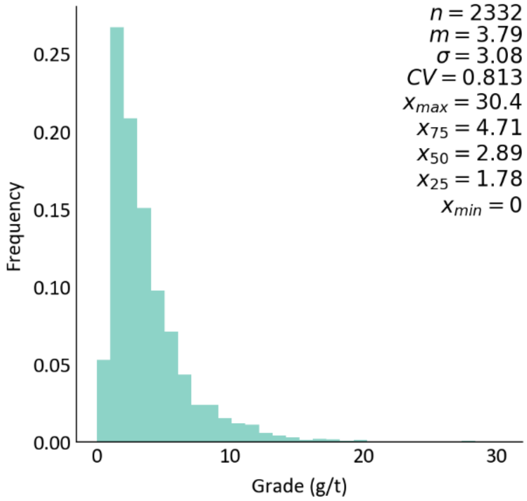


Fig. 4.8: Raw data distribution

The variable thickness of the vein created an interesting dynamic when approaching declustering with standard techniques such as cell and polygonal declustering. Since the drillholes are approximately perpendicular to the solid model and the data spacing along a drillhole, composited to 5 m, is generally less than the spacing between drillholes, the weights are primarily informed by the thickness of the vein and not the density of information. This leads to high weights at thin vein sections, which tend to correspond to lower grade values. Consequently, the declustered average grade is reduced severely to a value of 3.2 g/t at a cell size of 150 m (Fig. 4.9). The influence of high-grade samples is too low using the weights acquired with this method, producing realizations with an average mean of 3.52, 10% higher than the declustered mean.

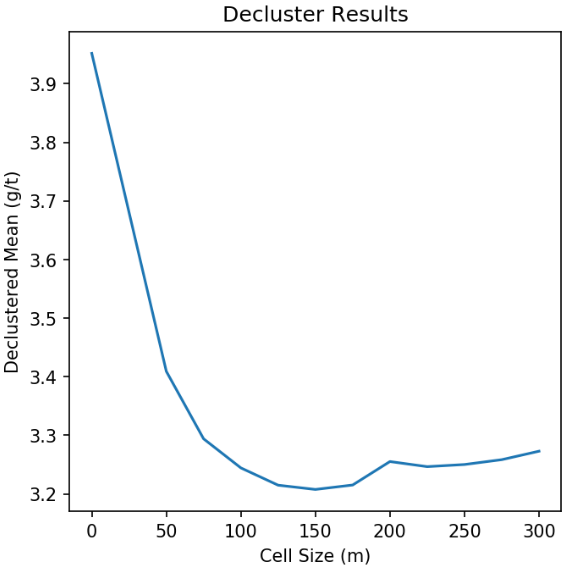


Fig. 4.9: Plot of declustered mean vs. cell size for simple workflow

A novel declustering workflow is implemented to minimize the impact of the data configuration on the grade distribution. The most critical alteration in this workflow is that weights are assigned by drillhole, not by data point. The data is rotated so the hanging wall side of the high-grade structure is horizontal. The point with the maximum z value in the new frame of reference is selected for each drillhole and used in a 2-dimensional cell declustering. A variety of cell sizes are considered, and the size which minimizes the average grade of these samples is selected. The results for each cell size and declustered distribution are shown in Figs. 4.10 and 4.11, respectively. Note that declustering on a subset of the data causes a discrepancy between the means in the declustering curve and the associated declustered mean.

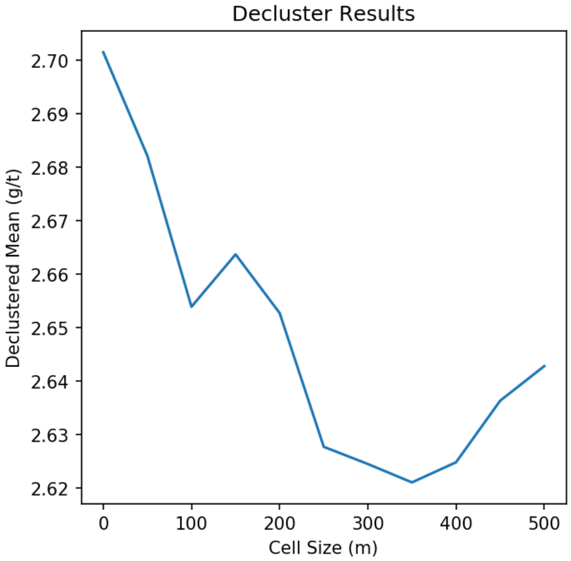


Fig. 4.10: Declustering mean by cell size on rotated data

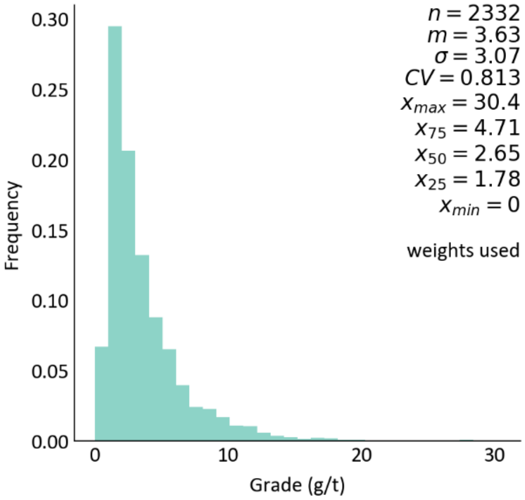


Fig. 4.11: Declustered distribution with a cell size of 350 m in novel framework

The declustered distribution with a mean of 3.63 g/t and standard deviation of 3.07 g/t is reasonably accurate for the following modelling and stope optimization when compared to the simulated distribution with a mean of 3.68 g/t and a standard deviation of 3.07 g/t, a difference in mean of 1.3% as calculated in Section 4.1.5.

4.1.3 Variogram modelling

Variograms are modelled on normal score transformed data that does not consider the declustering weights (Rossi & Deutsch, 2014). The orientation of the distinct vein-like structure of the high-grade region is used to infer the major, minor, and perpendicular directions as the strike, down dip, and perpendicular directions. This equates to an orientation of anisotropy with an azimuth of 34.7°, dip of 0°, and plunge of -50°. The experimental variograms and modelled fits are shown in Figs. 4.12 - 4.14. The parameterized variogram models are given in Tab. 4.1.

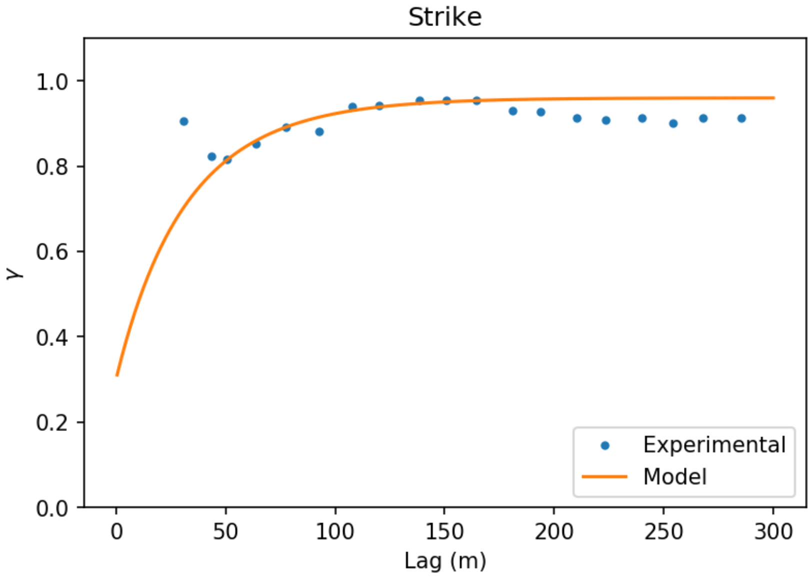


Fig. 4.12: Strike direction experimental variogram and fitted model

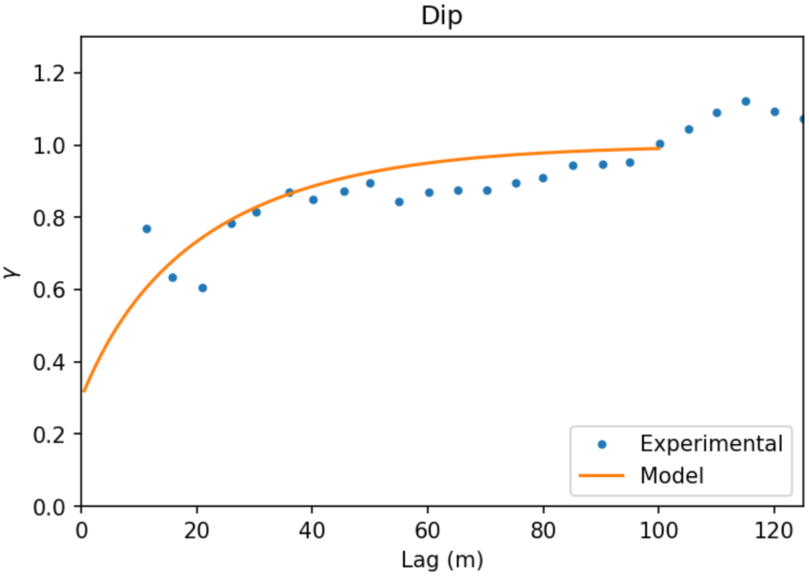


Fig. 4.13: Down-dip experimental variogram and fitted model

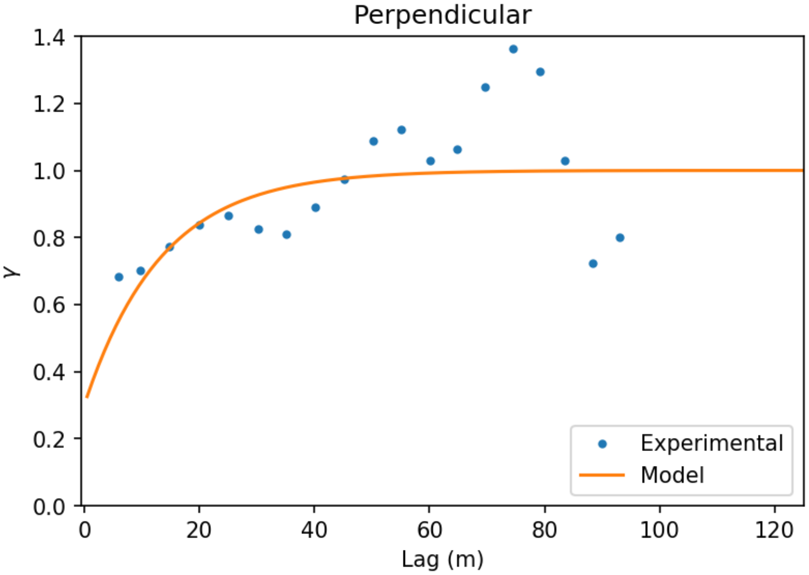


Fig. 4.14: Perpendicular experimental variogram and fitted model

Tab. 4.1: Three-dimensional variogram model parameters

Nugget	First Structure			
	Sill Contribution	Type	Range (m)	Angles
0.3	0.1	*Exp*	50	34.7°
			40	0.0°
			20	-50.0°

Second Structure			
Sill Contribution	Type	Range (m)	Angle
0.56	*Exp*	110	34.7°
		40	0.0°
		75	-50.0°

Third Structure			
Sill Contribution	Type	Range (m)	Angle
0.04	*Exp*	*inf*	34.7°
		40	0.0°
		75	-50.0°

Here, Exp is the exponential variogram. The formula for an exponential variogram structure with a range of a at a lag of h is given in Eq. 16 (Rossi & Deutsch, 2014).

$$Exp_a(h) = 1 - e^{-\frac{3h}{a}} \tag{16}$$

Attention is given to avoiding overfitting the models, especially where the experimental data are noisy. This is common in the perpendicular direction which in this case is approximately equivalent to the downhole direction where the drillholes intersect the high-grade structure. Downhole variograms are often less noisy than their counterparts due to the guaranteed continuity of data; however, the data extents perpendicular to the structure limit the available pairs to inform the corresponding variogram calculation, causing the noise to occur. The fit aims to represent the short-scale variability of the data as well as the experimental range while ignoring some cyclicity, which occurs around a 30 m lag. This parameterization is used in the subsequent SGS implementation in Section 4.1.4.

4.1.4 Sequential Gaussian Simulation

The GSLIB program USGSIM is used to create 20 (SGS) grade realizations. The data is transformed to a normal distribution using a quantile-quantile transformation and accounting for declustering weights. Spatially, the data is transformed with a 34.7° rotation counter-clockwise around point (123340, 259885) to align the strike of the high-grade structure with the N-S axis. Figures 4.15 and 4.16 show the averaged output of the realizations overlaid on the rotated high grade structure on characteristic plan and section views, respectively. The full parameter file for USGSIM and further section and plan views are included in Appendix B.

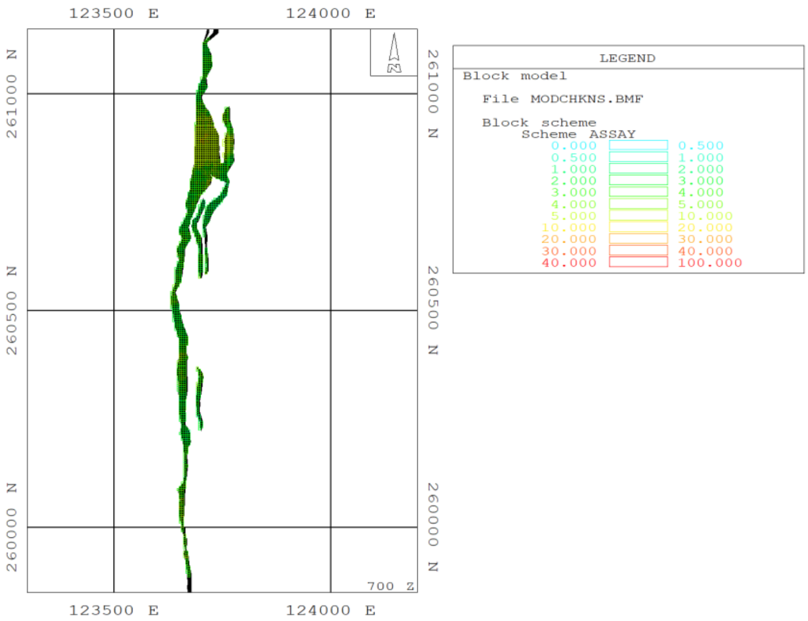


Fig. 4.15: Plan view of averaged simulation results overlaid on high-grade structure at z = 700 m. Grid is in meters

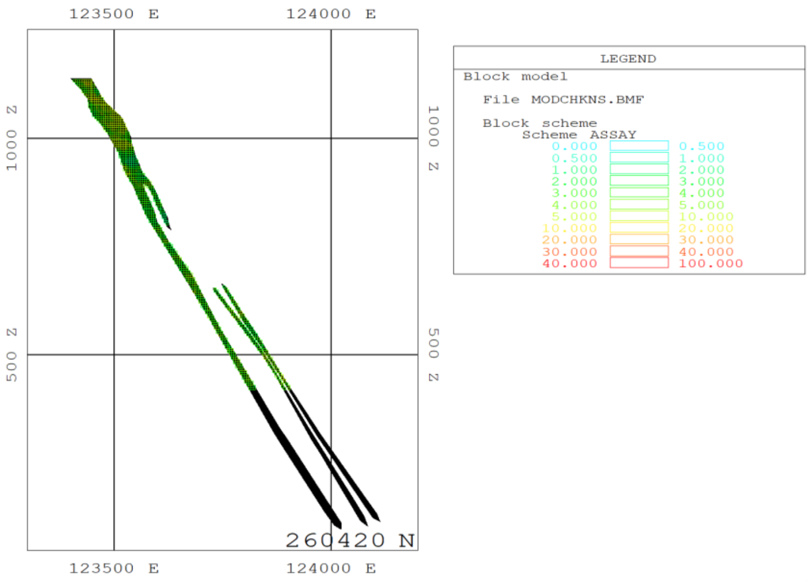


Fig. 4.16: Section view of averaged simulation results overlaid on high-grade structure at y = 260420 m. Grid is in meters

The output realizations produce data distributions with averages between 3.58 g/t and 3.81 g/t, with an average mean of 3.68 g/t. The standard deviations lie between 2.98 g/t and 3.16 g/t, averaging 3.07 g/t. Figure 4.17 shows the distribution of a selected realization, realization zero.

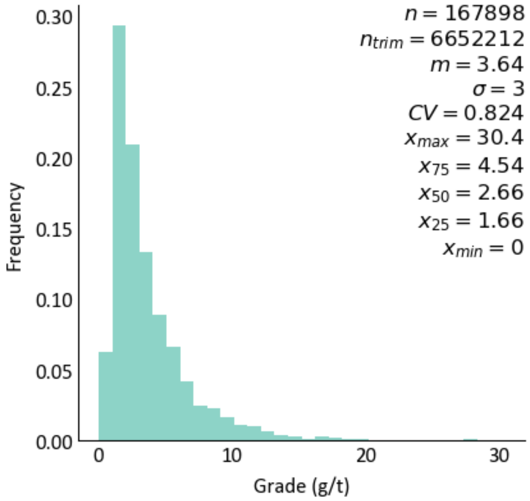


Fig. 4.17: Produced distribution of realization zero

4.1.5 Model Checking

Multiple model checks are performed to validate the stochastic grade model: 1) the reproduction of the grade distribution is assessed, 2) variogram reproduction is checked, and 3) the average of the simulated values are compared to the kriged value at each location. These checks are suggested to validate internal model consistency by Rossi and Deutsch (2014).

4.1.5.1 Grade Distribution Reproduction

If the proper presumptive data distribution is input into the SGS model, its statistical parameters should be approximately reproduced by the data output in each realization and more strongly reproduced on average (Rossi & Deutsch, 2014). Fig. 4.18 compares the distributions of each realization and the input declustered data.

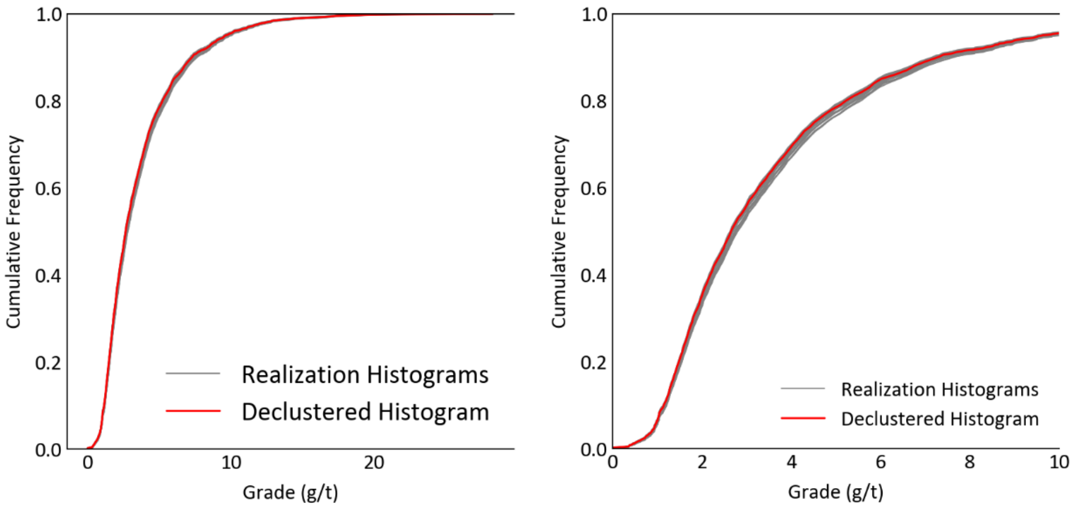


Fig. 4.18: Cumulative histogram for all realizations and declustered data for full grade range (left) and subset (right)

The spread of the realization distributions is very narrow, and the declustered histogram falls within their range. To provide a more detailed comparison, the relative spread of the mean and standard deviation of each realization is shown in Fig. 4.19. For comparison, the declustered distribution parameters are plotted as well as the average of the realization parameters.

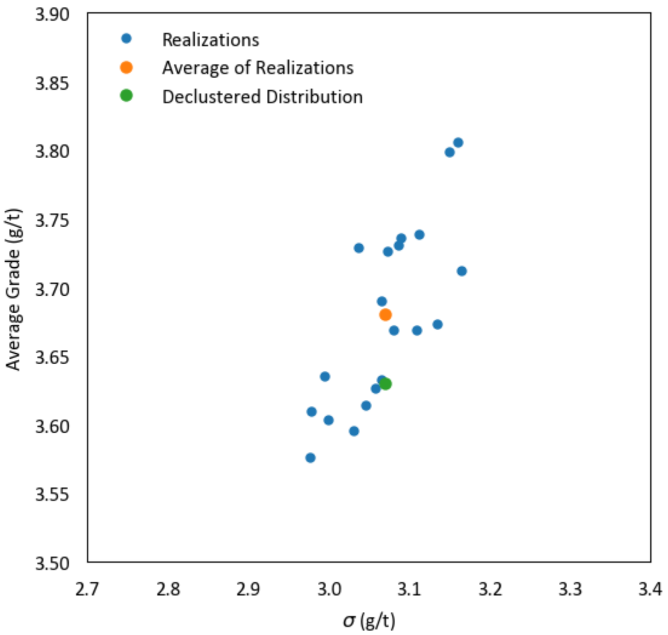


Fig. 4.19: Comparison of realization, average, and declustered distributions

Notably, the declustered and average standard deviations are the same at 3.07 g/t. The average mean grade of the realizations is 3.68 g/t, 1.4% higher than the declustered mean grade of 3.63 g/t. This is a reasonable reproduction of the distribution and acceptable for the purposed of this case study. The range of realization means is also reasonable, with the highest realization reporting an average grade 4.8% greater than the declustered mean, and the lowest-lying 1.5% below.

4.1.5.2 Variogram Reproduction

One property of SGS is that it reproduces spatial variability (Rossi & Deutsch, 2014). Consequently, variogram reproduction can be checked to assess the suitability of the input model. Here, GSLIB program *Gam* is used to calculate the experimental variogram for each direction that is modelled in Section 4.1.3. *Gam* calculates variograms for gridded data according to input block steps. Since the strike is aligned with the north axis, that direction is simple to reproduce, but the down-dip and perpendicular directions are approximated with block ratios to reproduce the modelled directions. The variograms for the down-dip direction are calculated at an azimuth of 90° and a -36.8° dip, and the cross dip at the same azimuth and a 45° dip. The cross-dip check is also attempted at a 51.3° dip, but the 32 m minimum lag length required to generate that angle is too long to generate a meaningful comparison. The variograms for each realization are compared to the input models and experimental points (Fig. 4.20).

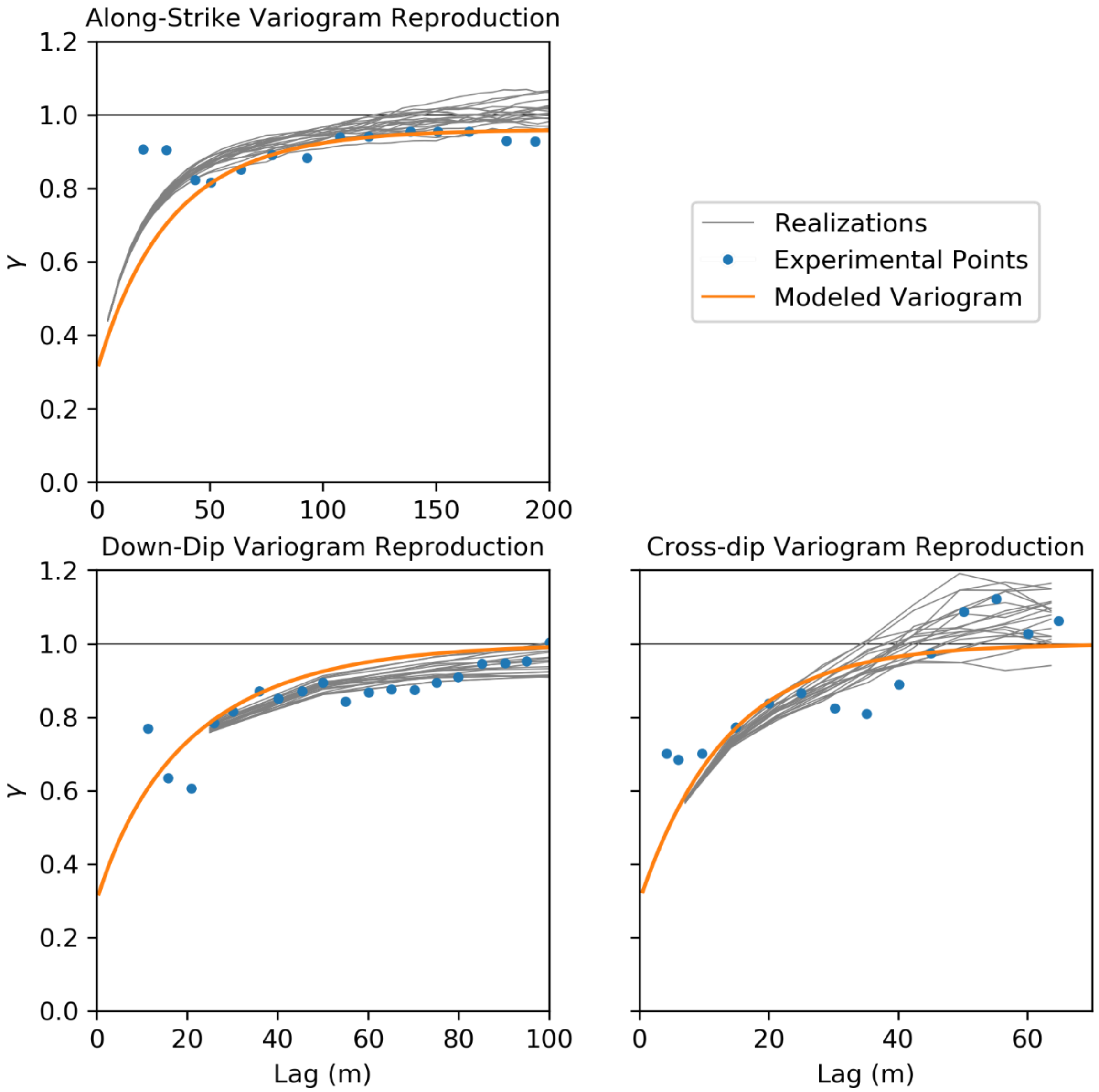


Fig. 4.20: Variogram reproduction assessment

The along-strike variogram appears to have inflated variance at low lags, which might be explained by the higher values seen in the initial low-lag experimental variogram points than the model. Similarly, in the down-dip direction, the realization variograms tend to lie between the variogram model and the experimental points, particularly at higher lags. Short lags are more challenging to assess as the minimum lag calculated is 25 m. The perpendicular variogram is also difficult to assess as there are no calculated pairs below a lag of 32 m if an approximately equivalent angle of -51° is used. Instead, a -45° dip is considered, allowing a lag of 1.41 m. The lags which are assessed are more varied between realizations than the other directions but are still relatively close to the modelled values. Overall, the variograms are reasonably reproduced.

4.1.5.3 Comparison to Kriged Model

The final check produced for the stochastic grade model is a comparison of the averaged stochastic model to a kriged model. The values at each location are cross plotted to allow a visual comparison (Fig. 4.21). The variables have a correlation coefficient of 0.89 and a mean square error of 0.085, suggesting a reasonable reproduction. A mean error of -0.004 indicates low bias in the results. This is considered acceptable, particularly because only 20 realizations are produced.

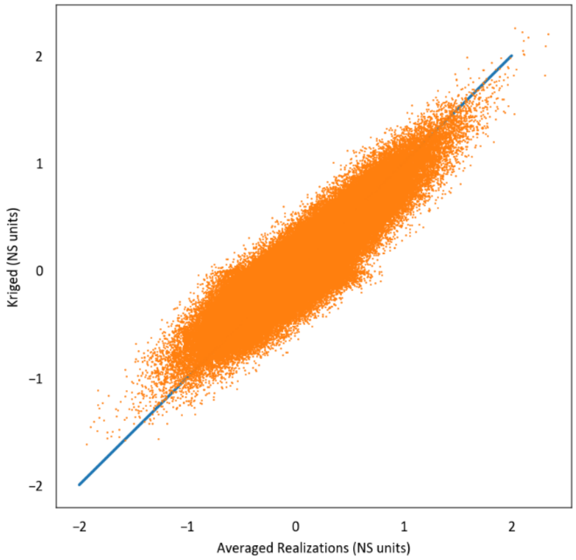


Fig. 4.21: Cross plot comparing average of realization values and kriged values at each location

4.1.5.4 Model Checking Conclusions

The model reasonably reproduces the data distribution and spatial variability, as determined in Sections 4.1.5.1 and 4.1.5.2. The block model section and plan views (Fig. 4.15, Fig. 4.16, Appendix A, Appendix B) additionally show good containment of simulated blocks within the high-grade structure. Finally, a comparison of a kriged grade model and the average of the realizations show a reasonable reproduction with a high correlation coefficient and low mean squared error. The bias of the errors is also shown to be very low. Together, these factors show that the simulated model is internally consistent and a reasonable interpretation of the spatial grade distribution.

4.2 Stope Optimization

The grade model produced in Section 4.1 is considered for optimization with the proposed stochastic stope optimizer. The optimization is run with various risk penalty factors, and the resulting layouts are compared in a decision-making framework. The top layouts are scheduled using the stochastic simulated annealing scheduler, which accounts for time-based risk and differential access costs to better compare and finalize the layouts.

To assess the impact of non-linear calculations on the optimization, results with no risk discount are compared to a layout found by applying the recoveries both by block and by stope on an e-type model, ignoring the non-linear impacts of stope-based recoveries.

4.2.1 Parameters

Parameters representative of the operation are provided to facilitate the optimization of subsurface extraction with an open stoping method. This includes stopes and pillar sizes, recovery ranges, mining cost and metal price estimates. Tab. 4.2 shows the parameters input for optimization.

Tab. 4.2: Parameters for stope optimization

Parameter	**Symbols**	**Values**
Stope size minima (blocks)	$xmin, ymin, zmin$	(1,4,4)
Stope size maxima (blocks)	$xmax, ymax, zmax$	(5,8,8)
Model size	$Xdim, Ydim, Zdim$	(165,249,166)
Model origin	$Xmin, Ymin, Zmin$	(123342.5,259887.5,422.5)
Block sizes	Xsz, Ysz, Zsz	(5,5,5)
Pillars	$xpil, ypil, zpil$	(1,1,1)
Risk scaling factor	α	0 to 0.3 by 0.1
Mining cost per block	C_m	\$75/t
Processing cost per block	C_p	\$25/t
Metal price	P	\$38.59/g
Density	γ	2.7 t/m^3
Recovery	Function	Eq. 17

Recovery is assumed to behave non-linearly with a range between 85% to 95%. To form a recovery function, 5.0 g/t is taken as the optimal grade for processing, with a 95% recovery. A piecewise parabolic curve is fit with a discontinuity at 5.0 g/t. Initially, the portion of the curve below 5.0 g/t is set to have a 0% recovery at 0 g/t. An economic cutoff is back-calculated as 3.5 g/t, returning a recovery of 85%, which is within the given range. The function above 5 g/t calculated to have a recovery of 90% at 50 g/t. The full recovery function is given in Eq 17.

$$r = \begin{cases} \frac{19}{20} - \left(\frac{19}{500} \times (g-5)^2\right) & , \quad g \leq 5 \\ \frac{19}{20} - \left(\frac{3}{125000} \times (g-5)^2\right), & \quad g > 5 \end{cases} \qquad 17$$

Stope placement is limited to between 0 and 485 m above the base of the simulated model due to the requirement for a crown pillar below the existing excavation. The required pillar is assumed to be approximately equal to the width of the pit at the pit bottom.

4.2.2 Implementation

The stochastic stope optimization algorithm is implemented with a greedy secondary algorithm as there is no requirement for continuous longitudinal or transverse pillars. Each level is optimized for every considered risk discount in succession to reduce the reading and calculation of level data. The dynamic level combination adds continuous sill pillars, which are identified as a requirement for this project. It is indicated that the pillars should be different in height depending on the width of the above excavation, but stope size-dependent pillar dimensions are not implemented for this algorithm. This implementation considers all sill pillars to be one block tall, the minimum stope width.

The algorithm initially takes about 6 days to produce the layouts associated with the 4 risk discount factors, indicating a need for efficiency improvements. The algorithm is adjusted to reduce the data input to any call of the 1-dimensional dynamic optimization to include only strips bounded by the outermost positive stopes, limiting the memory and iterations required to reach a solution. This greatly improved the speed of the algorithm, with the improved implementation taking 4.5 hours to produce results for the 97-level stope optimization problem at 4 discount levels. The 5 considered stope dimensions in each direction lead to 5^3 or 125 possible stope sizes.

4.2.3 Results

Four stope layouts are produced for α values between 0 and 0.3. The layout produced for the $\alpha = 0$ case is shown in Fig. 4.22.

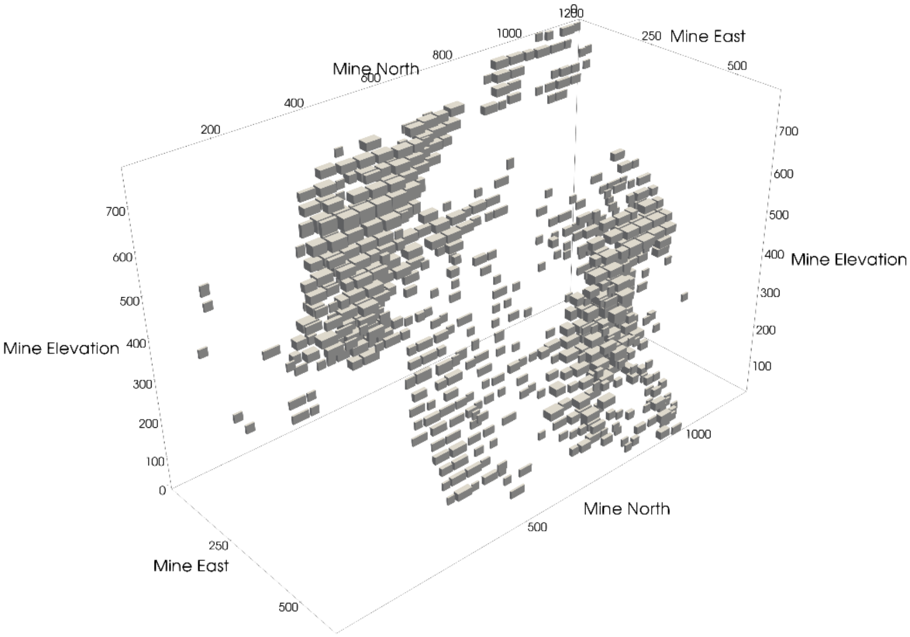


Fig. 4.22: Schematic view of $\alpha = \ 0$ *results*

Note that some stopes are set apart from the main valuable area reducing the likelihood that those stopes are economic if individual access costs are assessed.

As the risk discount is increased, the number of stopes, value, and standard deviation of the layout is reduced. Fig. 4.23 shows the changes in the number of stopes and value by risk discount. The number of stopes appears to decrease linearly as the risk discount increases, while the value inflects at $\alpha = 0.2$. Additionally, the stope number has a broader range with values between 422 and 265, a 37% reduction. The value ranges from \$560 M to \$540 M, a difference of only 3.5%.

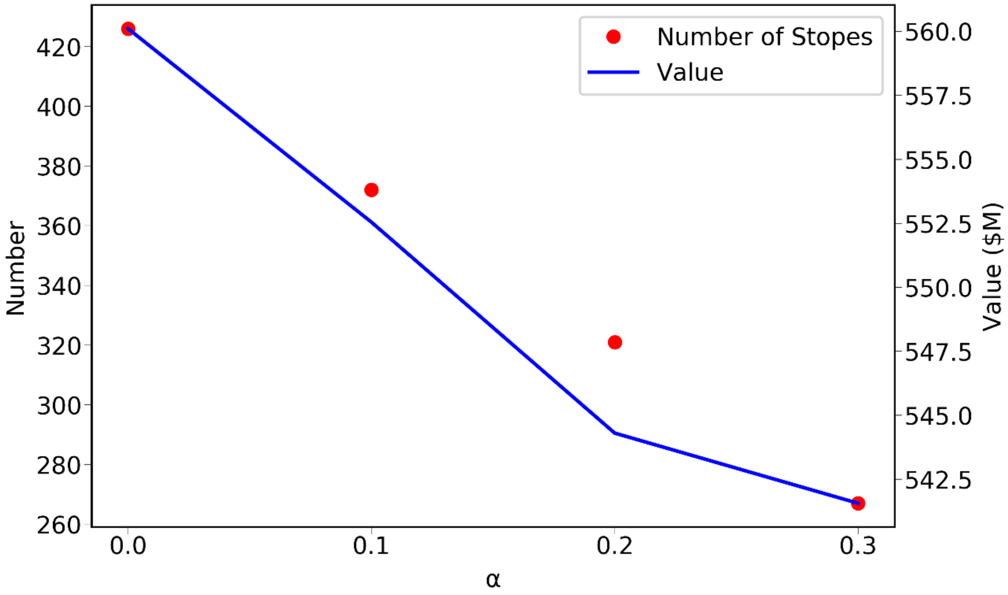


Fig. 4.23: Stope number and value by risk discount

When the results are plotted in value-risk space (Fig. 4.24), they do not form a convex efficient frontier as might be expected, most noticeable when considering the $\alpha = 0.3$ risk penalty optimization; however, none of the solutions appear dominated, and therefore all remain viable solutions.

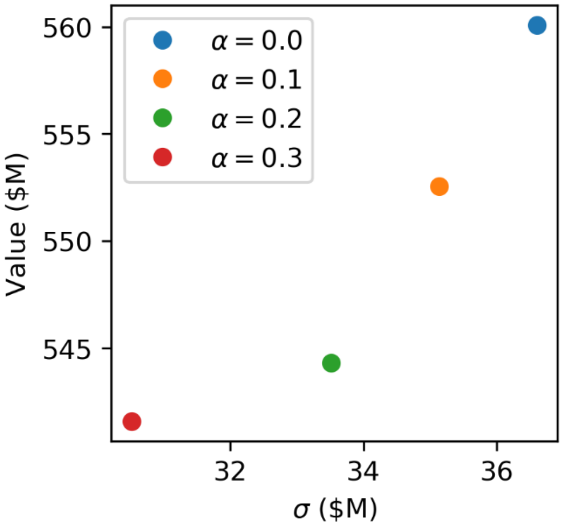


Fig. 4.24: Solutions in value-risk space

As the best layout is not apparent from the value-risk plot, alternative assessments are required to choose between them. The first method in the process is to assess the alternatives for stochastic dominance. An alternative is stochastically dominated if its value is lower than that of another layout at every quantile of its distribution (Rossi & Deutsch, 2014). The cumulative histograms are plotted for all the optimized layouts to assess whether any are stochastically dominated (Fig. 4.25).

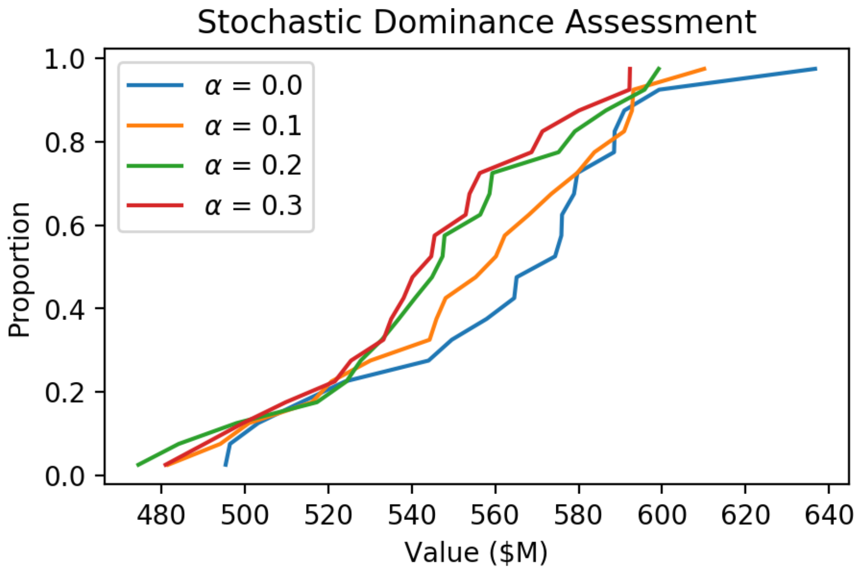


Fig. 4.25: Assessment of stochastic dominance between optimized layouts

In this case, one alternative is dominated: the layout at $\alpha = 0.3$ is dominated by that at $\alpha = 0$. The remaining layouts are not dominated. Visually, the layout for $\alpha = 0$ outperforms the others on most quantiles, with the $\alpha = 0.1$ outperforming it on two quantiles and the $\alpha = 0.2$ on one. With the higher upside and downside outcomes, it appears that the $\alpha = 0$ layout should be chosen in most cases. This is likely because of the low relative error in even the riskiest optimized layout. The remaining layouts, at $\alpha = 0, 0.1,$ and 0.2, are considered for scheduling to account for time-dependent risk when selecting the final layout.

4.2.4 Scheduling

The layouts remaining after optimization and assessment in Section 4.2.3 are further analyzed through a stochastic scheduling procedure. Each layout is considered in stochastic annealing schedule optimization using the same risk discount as in the corresponding layout optimization. The schedules are optimized to maximize a risk-rated NPV by level, assuming a vertical development cost of $10000/m and a discount of 8% per period. The resulting range of returns is shown undiscounted by period in Fig. 4.26 and overall in Fig. 4.27.

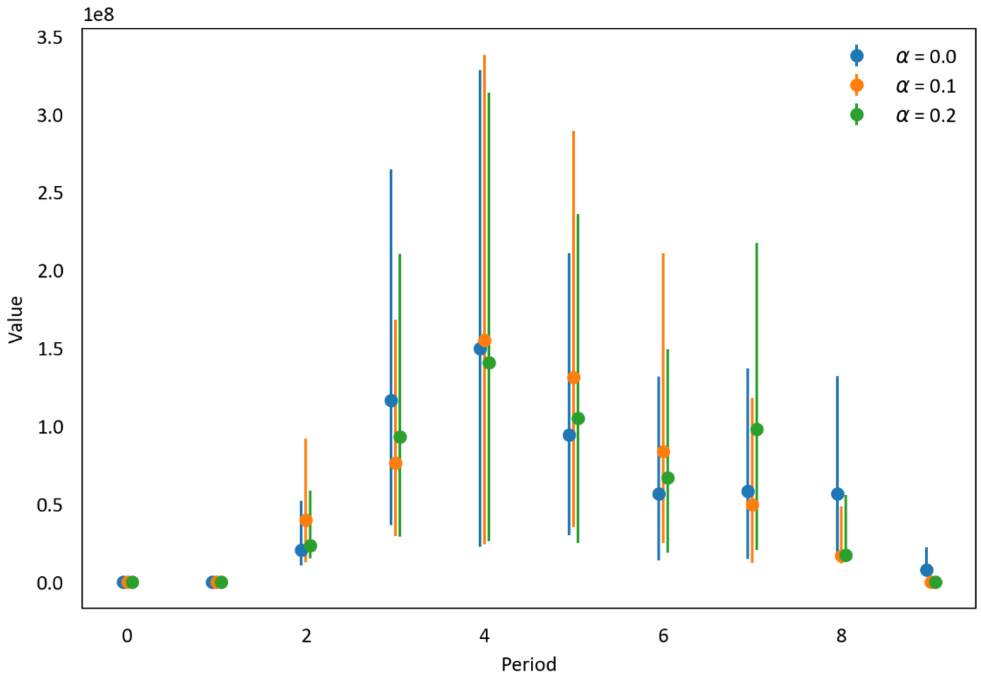


Fig. 4.26: Scheduling results with 90% confidence interval

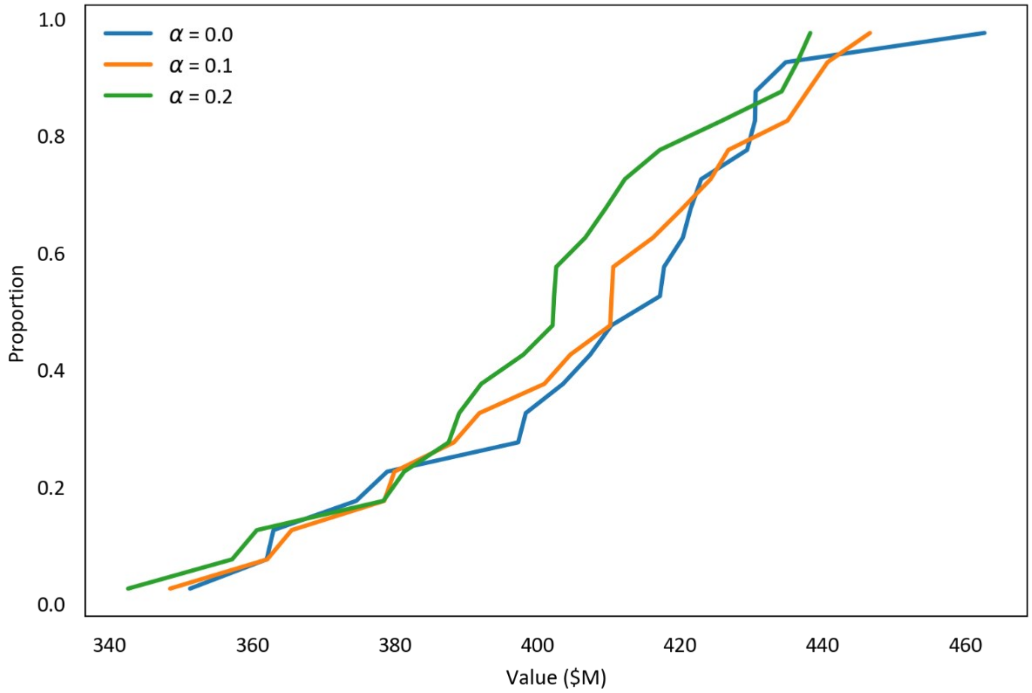


Fig. 4.27: NPV histogram by risk discount to assess stochastic dominance

The undiscounted period distributions clearly show value being preferred early in the schedule. The general correlation of high value and high risk is also apparent here. There is no stochastic dominance between the solution distributions after scheduling (Fig. 4.27). The $\alpha = 0.2$ case, however, is clearly worse than the other options, only competing in low quantiles but generally producing visibly worse results. The remaining solutions are reasonably similar, with the α= 0 case displaying a large jump in value at the high tail. While more analysis could be pursued, such as a utility theory comparison, the $\alpha = 0$ case is selected due to the better mid-range performance compared to that with $\alpha = 0.1$. The order of extraction by level for the $\alpha = 0$ layout is shown in Fig. 4.28. The resulting layout is reasonable, with levels mostly being mined as they are accessed, with some exceptions where more stopes are accessed in a period than can be mined. This is most notable looking at period 6, during which previously omitted levels are mined.

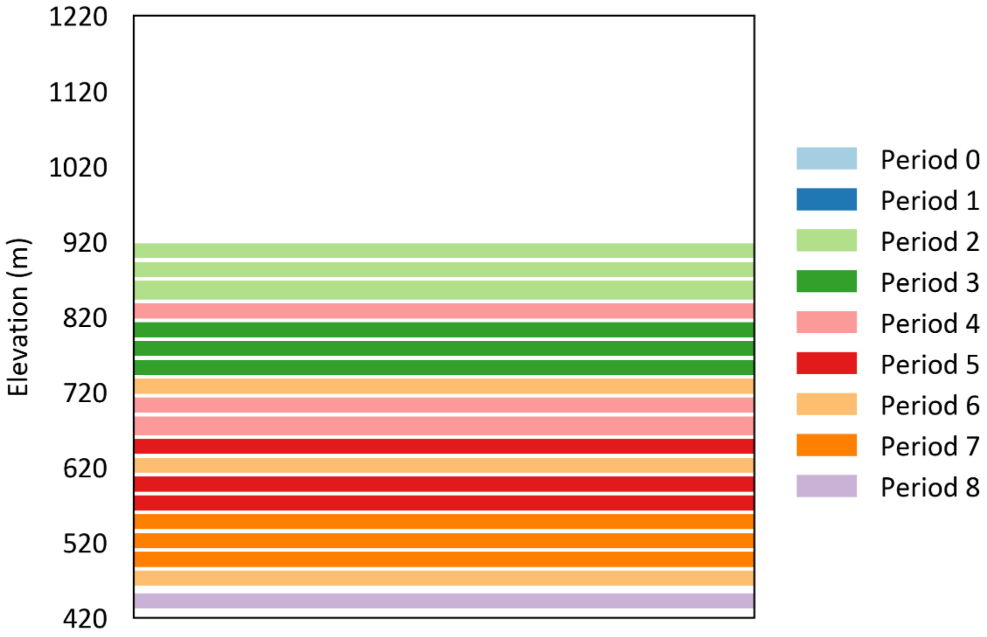


Fig. 4.28: Order of extraction by level for chosen layout

In a case where a decision is less clear, expected utility or regret theory can be utilized to decide between remaining options (Bleichrodt & Wakker, 2015). The maximization of expected utility or minimization of maximum regret is considered in this case. This requires the development of a utility function that considers risk according to personal or corporate risk tolerance. The minimization of maximum regret, alternatively, reduces the impact of underperformance on lower value realizations. In an industry where meeting or exceeding targets is an indication of success, adopting a regret-based decision-making paradigm is worth pursuing.

4.3 Non-Linearity Analysis

The optimization is run on an e-type model of the simulated grades to assess the impact of considering non-linearities by implementing stochastic optimization. Recoveries are first applied by block and then by stope before comparing both results to the initially optimized layouts from Section 4.2.3. Figs. 4.29 and 4.30 show the stope-based e-type and initial results in value-risk space and as full distributions respectively, facilitating a direct comparison.

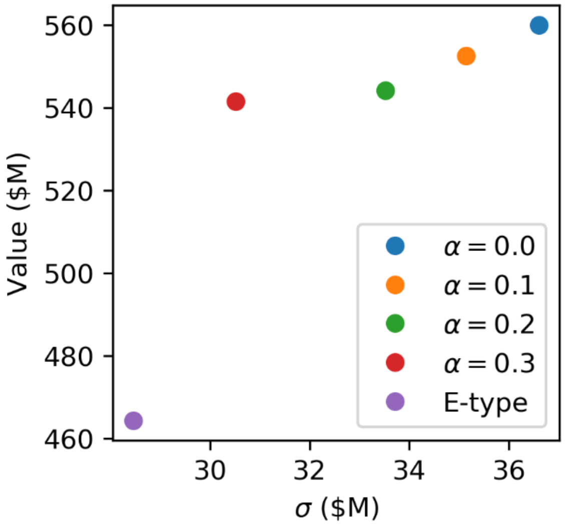


Fig. 4.29: E-type comparison in value-risk space

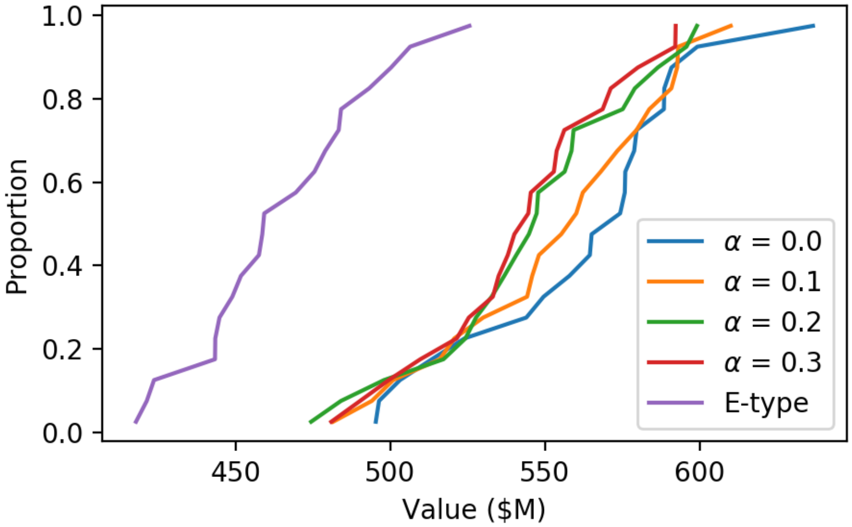


Fig. 4.30: Stochastic comparison of e-type optimization with recovery by block

In both cases, the e-type model severely underperforms all the stochastically optimized layouts, falling well below the apparent efficient frontier and is stochastically dominated by all alternatives. The average value of $464 M is 17% below the value of the $\alpha = 0$ layout. This shows the magnitude of the benefit realized when considering non-linearities in value calculation for stope optimization. Fig. 4.31 shows results for an additional optimized plan with recovery applied to each stope in an e-type model. This plan performs very poorly with an average value of $43.6 M and one realization on which it returns a value less than $0, further supporting the necessity of optimizing on a stochastic model to include non-linearities in calculations.

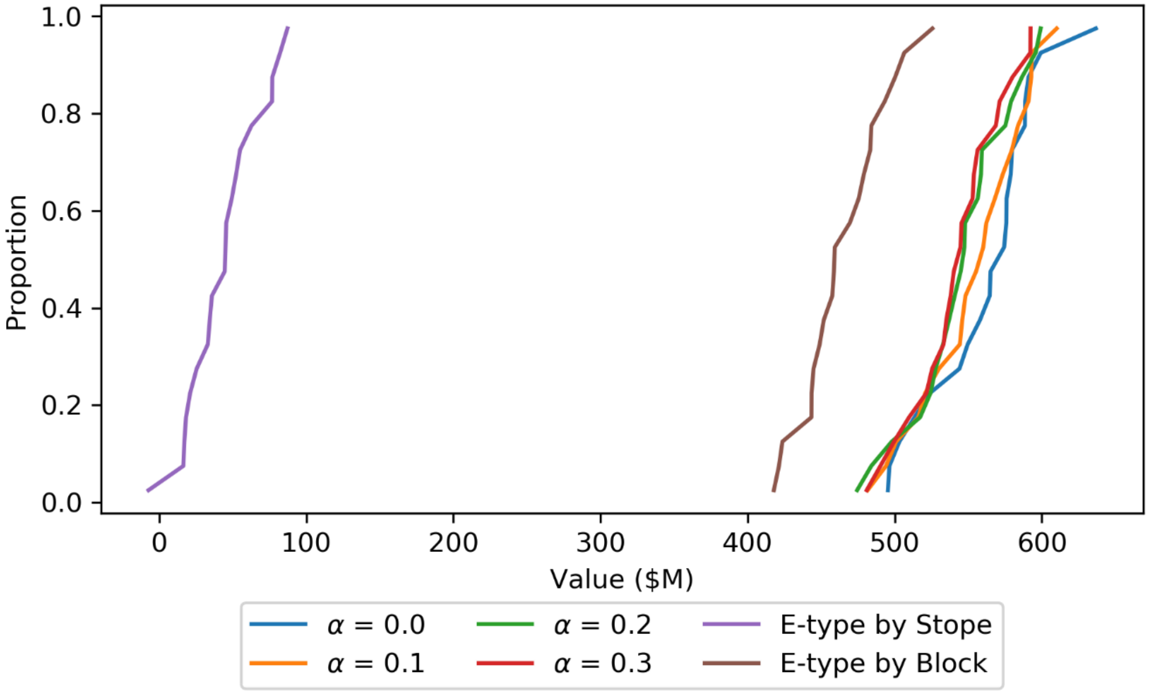


Fig. 4.31: Value distributions including e-type optimization with recovery applied by stope

The gap between the two optimized e-type layouts and the stochastically optimized solutions is not as drastic in practice as it appears in Fig. 4.31. This is because the information effect causes the uncertainty to become lower at the time of mining (Chiquini, 2018). There is often recourse available through the changing of mine plans over time, allowing the realized value of the e-type layouts to become closer to those of the stochastic optimizations. However, stochastic methods also do not consider the information effect directly and their value similarly improves as mining proceeds. The non-linearity example in Chapter 2.6 indicates that the stochastic interpretation is correct.

An additional consideration is that this comparison assumes that it is correct to calculate metal recovery at the stope level. Realistically, ore does not move as a unit from a stope to the processing plant. Instead, ore is often combined from various sources, impacting the head grade at the processing plant at any point in time. While this combination would ideally be considered, the time-independent nature of stope layout optimization makes it infeasible.

4.4 SSO Comparison

The ore body is optimized using SSO as implemented in Maptek Vulcan to compare the developed optimizer with existing deterministic methods (*Vulcan Envisage*, 2018). The slice method is used, as discussed in Section 2.5.1. A cutoff grade of 3.5 g/t is used for the optimization, and substoping is used to enforce x and z pillars. It is difficult and computationally costly to produce scenarios with identical constraints between SSO and the heuristic stochastic stope optimization algorithm as substoping can only define x and z pillars as a ratio to the stope width. It is difficult to maintain constant pillar sizes as they are constrained to block multiples. Two scenarios are considered to attempt to reach a reasonable comparison:

1. The x and z dimensions are constrained to the minimum values (1 and 4 blocks, respectively)
2. The z dimension is constrained to its minimum height, and x is varied between 5 and 10 metres.

Fig. 4.32 compares the stochastic stope optimizations with the value distribution of the SSO results.

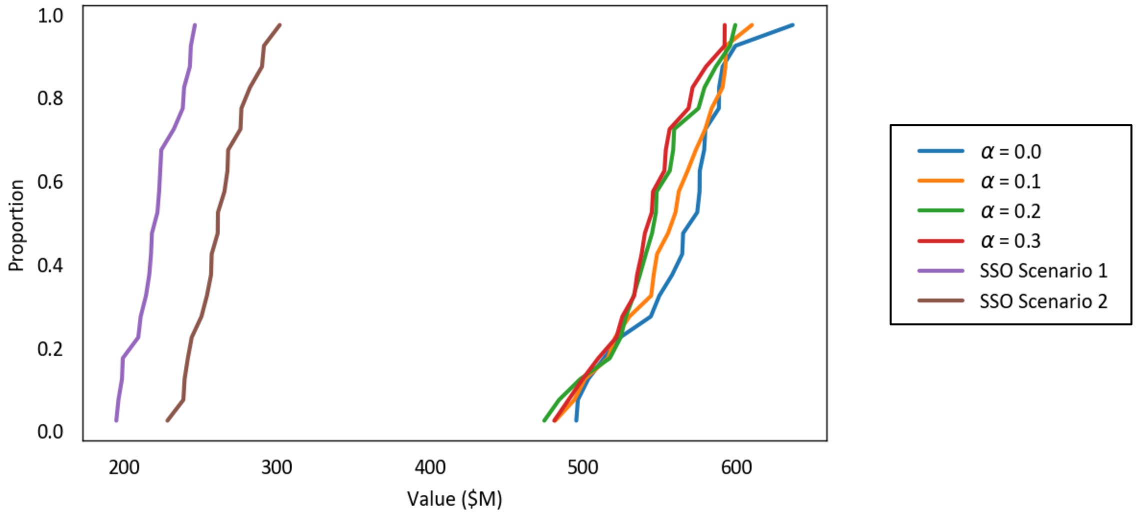


Fig. 4.32: Novel optimizer compared to SSO implementation

The SSO clearly underperforms the heuristic optimizer in the evaluated cases. The notable improvement when considering the heuristic stochastic stope optimizer increases confidence in the quality of the novel optimizer presented in this thesis. The quality of the comparison is assessed by evaluating stope dimension distributions in the x, y, and z dimensions. (Figs. 4.33 and 4.34)

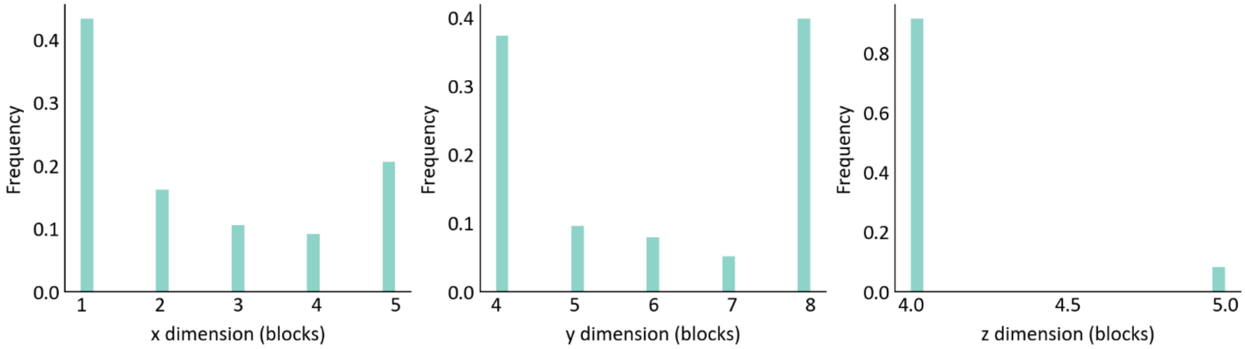


Fig. 4.33: Heuristic optimizer stope dimension distributions at $\alpha = 0$

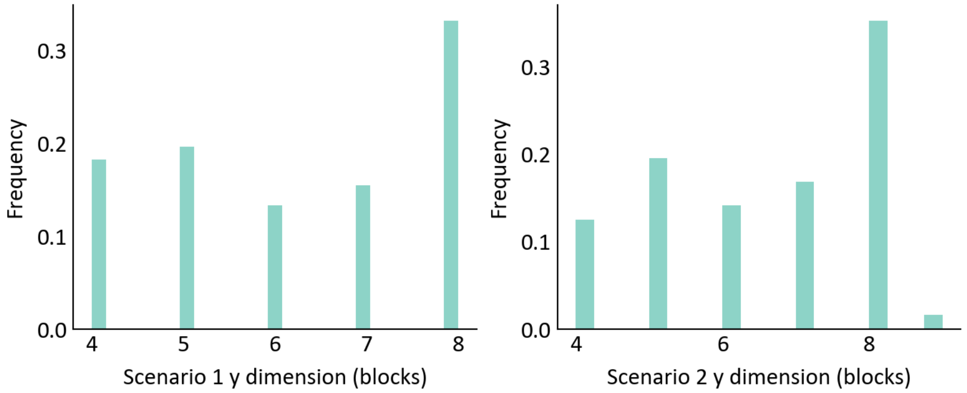


Fig. 4.34: Scenario 1 (left) and 2 (right) y dimension distribution

The x and z dimensions are fixed in scenario 1, at 1 and 4 blocks, respectively. In scenario 2, the z dimension was constrained at 4 blocks. All stopes are 2 blocks in the x dimension even though 3 block stopes are permitted. The largest differences between the results are the lower frequency of low y stopes and restriction of x dimensions in both scenarios. These results support the statement that the produced heuristic optimizer truly outperforms SSO in this case.

It should still be recognized that this is not a perfect comparison, and alternate implementations of the SSO or the heuristic stochastic algorithm may obtain different results. Additionally, comparing on this case study benefits the presented algorithm as the widely varying stope sizes cannot be reproduced using SSO. This limitation should be considering when comparing the algorithms.

4.5 Discussion

The algorithm shows promising results when implemented on a real-world case study, accessing risk-rated results and accounting for non-linearities in value calculations. Of the four optimized layouts, three remain for further analysis as one is stochastically dominated. Schedules are optimized for each layout with a simulated annealing approach, considering the same risk discount used for the layout optimizations. The layout optimized with no risk discount is found to outperform the alternative solutions marginally and is selected as the overall solution.

An additional comparison is completed to assess the benefit of accounting for non-linearities though stochastic optimization, showing significant improvement by applying the non-linear transfer function on each realization before averaging the response. The e-type optimization returns a stope layout that preforms 17% worse than the layout produced with no risk penalty considered. This leads to the conclusion that stochastic methods are necessary for the optimization of stope layouts.

Implementing the stochastic stope layout optimization as presented accounts for grade uncertainty but does not directly consider boundary uncertainty. Boundary uncertainty can take varying forms with different implications for deposit assessment. The main contributing factor is the scale of the uncertainty in comparison with the scale of the deposit. If boundary uncertainty is small in scale, it is likely sufficient to consider its contribution to grade uncertainty in those blocks. However, there are also cases, particularly in vein deposits, where the positional uncertainty can be considerable compared with the width of the vein (Fig. 4.35).

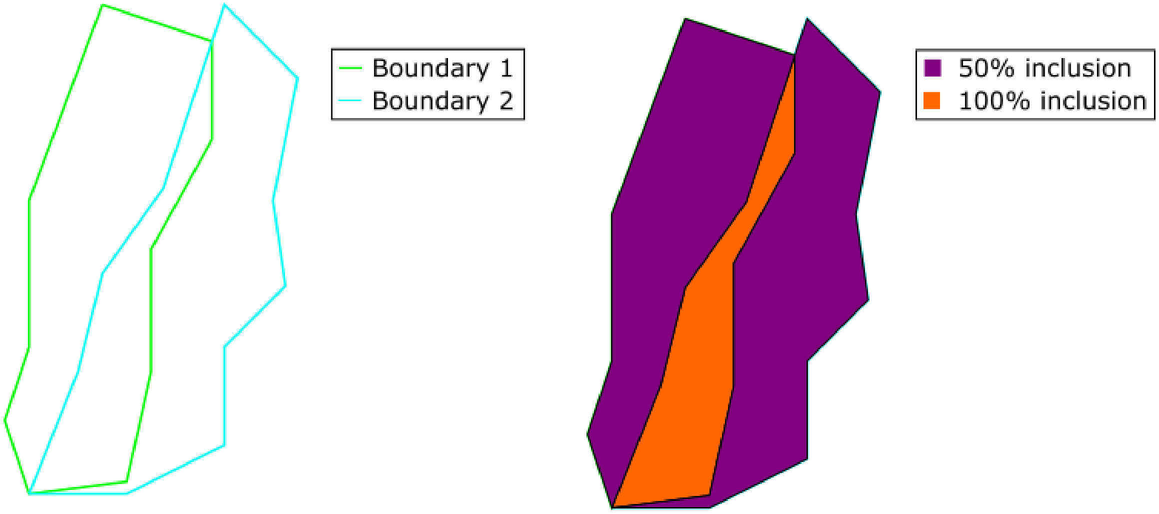


Fig. 4.35: Boundary uncertainty causes pessimistic assessments due to stope averaging process as only a small number of blocks may be included in most or many interpretations

In these cases, the current averaging process in the algorithm is not appropriate as only a small subset of the blocks might be profitable over all realizations, and the stopes found will be limited around these blocks. However, it is clear when evaluating any number of realizations that a larger, more inclusive stope is possible. In many cases, the resolution of data increases and the boundary locations of the veins will be further informed before the time of mining. Correct assessment in these scenarios requires implementing an information effect workflow (Chiquini & Deutsch, 2018; Neufeld et al., 2007).

The general idea of this workflow is to simulate the level of information known at the time of mining and create separate simulated models for each resampled case (Neufeld et al., 2007). Each set of realizations is then assessed with planning tools, such as the developed stope optimizer. This will not result in a single stope layout output but will generate a range of returns from a plan as expected at the time of mining. Maps can also be produced to show areas that are more likely to be mined than others.

Chapter 5: Conclusion

This thesis develops a practical heuristic stochastic stope optimization algorithm that enables the consideration of sets of geostatistical realizations in the optimization process. The algorithm allows for the calculation of non-linear transfer functions and the consideration of risk in decision-making. An existing deterministic algorithm is adapted to accept realizations as input, applying a defined objective function on each realization before averaging the response. A workflow is developed in which several layouts are optimized with different risk penalties, and the results are compared to determine an overall layout.

5.1 Contributions

The main contribution of this research is an algorithm with the ability to optimize stope layouts considering a suite of realizations concurrently heuristically. Two outcomes are achieved by this configuration: 1) non-linear transfer functions are correctly accounted for in the calculation of layout response; 2) integration of risk into the objective function for active consideration of risk in optimization; 3) demonstration of algorithm performance on a real deposit.

5.1.1 Non-linearities

It is well known that non-linearities in any calculated response cannot properly be accounted for if the input is averaged before the response is calculated (Rossi & Deutsch, 2014). Configuring the algorithm to calculate response on each realization before averaging the values is the correct approach to optimization. Logically, optimization can only reach the correct solution when the objective calculation properly accounts for any aspects that could create substantive discrepancies; otherwise, what appears to be an optimal solution may not be optimal.

5.1.2 Risk Integration

Risk is an inherent aspect of mine planning, optimization, decision-making, and investment. When choosing between two possible plans, both the average value of the plan and the uncertainty in that value must be considered. The implementation of value calculation by realization allows for not only the assessment of plans with uncertainty but the optimization of plans with different levels of uncertainty, generating a variety of planning options fitting differing risk tolerances. This leads to the development of an efficient frontier of results which can be chosen between based on an assessment of stochastic dominance and a determination of risk tolerance. This improves the decision-making process when optimizing stope layouts.

5.1.3 Demonstrative Case Study

A case study is presented to demonstrate the algorithm as it is applied to real-world data. A set of gold data is modelled into 20 realizations with SGS, and an open stoping layout is optimized at four risk discount levels. The layouts are compared considering stochastic dominance, reducing the field to 3 cases. After stochastically scheduling all levels using a simulated annealing approach, it is found that the distribution of NPVs of the $\alpha = 0$ case dominates both others and is selected as the optimal layout. The algorithm is also applied to an e-type model. The results indicate the impact of the non-linearities on the optimization, with the e-type layout having a value 17% below the $\alpha = 0$ layout. The value distribution for this layout is also dominated by all alternatives and lies well below the efficient frontier.

5.2 Limitations and Future Work

While the work presented represents a real and valuable contribution to the body of research on stope optimization, the algorithm, as presented, has limitations in its implementation from both mining and programming perspectives.

5.2.1 Mining Method Scope

The scope of the optimizer has been limited primarily to the open stoping mining method. The mining controls are limited to stope sizes and pillars, allowing it to be adapted in its current form to any method which can be defined by those parameters. However, other stoping methods can require vertical alignment between stopes or stope edges, such as in the case of sublevel stoping, which reports blasted ore to a common extraction level. This alignment is not currently possible in the stochastic stope optimization algorithm. Additionally, any stopes produced are vertical, with no existing method to add or optimize inclination.

Pillar recovery is also difficult to account for in the stochastic optimization as pillars are either defined based on time of extraction or a set pattern. Pillar recovery requires that adjacent stopes are backfilled, often with a cement mix, and allowed to set imposing cost and scheduling differences. The mining and processing recoveries of the pillars are reduced from primary stopes due to the difference in scale and dilution from those adjacent and the relatively soft backfill.

5.2.2 Dimensionality Reduction

The dimension reduction leveraged to reduce the computational demand of the optimization at the expense of some completeness. Specifically, when a 1-dimensional strip is optimized, no direct consideration is given to where pillars are placed within strips that are adjacent to it horizontally or vertically (Fig. 5.1).

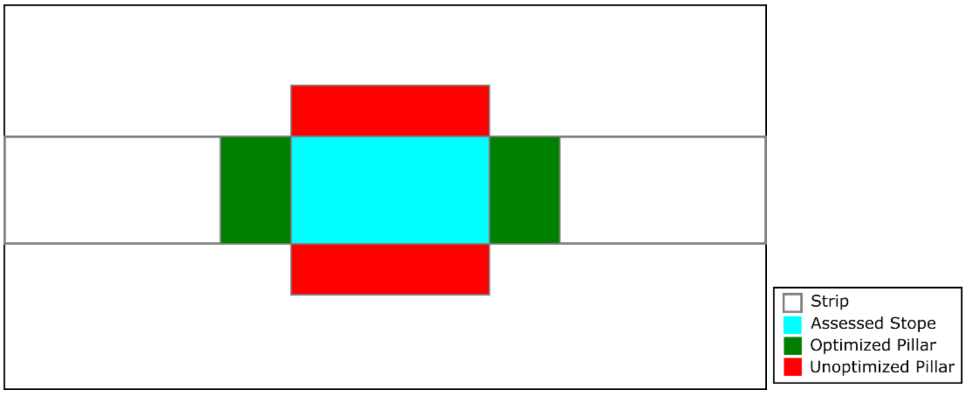


Fig. 5.1: Pillar optimization considerations. Red segments are not considered jointly with the possible stopes they overlap

The optional tabu optimization module (Fig. 3.11) reduces some impact of this issue by changing the order of strip selection and, consequently, pillar placement. However, it is possible that an optimal plan is not found with an alternate stope configuration as, theoretically, the first selected stope strip could need alternate stopes selected to reach the optimum.

5.2.3 Minability of Results

The layouts which result from this algorithm have some positionally outlying stopes that are not likely to be profitable at the time of mining. These stopes are those that fall on the outer extents of the optimized area, far from other groups of stopes. This distance leads to higher access costs and uneconomic extraction. The algorithm does not discriminate between stopes by the specific costs of accessing them in a mining scenario as that would require scheduling. The inclusion of outlying stopes is misleading, but the algorithm or workflow could be changed to remove the outliers. Without changing the algorithm itself, the practitioner can limit the workflow to act on panels of the deposit, ignoring areas where only outlying stopes are found. An ad-hoc alternative would be to post-process stopes, building clusters of stopes that are a maximum distance from each other, with a minimum number of stopes forming each cluster. This functionality could also be built-in as an optional procedure in the algorithm.

5.2.4 Additions to Algorithm

Several additions are suggested for future improvement to the algorithm, some more impactful than others. These include methods to increase the optimality of the results, better reflect real mining scenarios, add flexibility, and either combine or account for multiple aspects of planning during optimization.

The first suggested improvement is the addition of a method to account for variable mining and access costs. This can be implemented by, for example, referencing a function for distance from a set point, or including a rock-type with variable cost and recovery calculations. This addition would still not reproduce the exact costs of optimized access infrastructure, but it is worth investigating whether the implementation of an analogue for access cost improves the optimality of the layout when scheduling is considered.

A more direct approach is to consider multiple aspects of underground mining optimization simultaneously. It is suggested that a partial combination, where some aspects are considered less thoroughly than others, could lead to an improved solution overall compared to the typical iterative approach where each portion is considered separately. This could include considering a scheduling analogue during the dynamic level selection portion of the algorithm, accounting for an approximation of differential access costs and the impact of mining order on the optimal layout. A potential issue with accounting for scheduling in stope layout optimization is stopes mined later would be given less weight in the objective function if a typical discounted value weighting such as net present value is considered, possibly leading to a depressed nominal value in the final layout. This could be relieved by considering the order of extraction to define the precedence of infrastructure while only considering discounted value after the optimization is completed. Further investigation is required to determine the best way to combine these optimizations while maintaining reasonable computational complexity.

Another shortcoming of the algorithm is the limitation to vertical stopes. An immediate solution to this problem is to input a model that is rotated to a dip that is assumed appropriate for the stopes in question. As an attempt to optimize the dip, multiple angles could be considered, and the results compared. However, this would still lead to all stopes following the same dip, which is not always practical. Other methods to evaluate varying stope dips should be determined and evaluated.

It is noted that in some cases, positive stopes are indicated in an optimal solution that are spatially separated from the bulk of the other stopes. In this case, it is unlikely that the incremental value of the layout would increase when a stope of this type is added due to the cost of access. This should be accounted for in either the subsequent scheduling ad access optimizations or through general proximity limits to cull these external stopes.

Finally, the current implementation of the algorithm is programmed in the Python programming language within the Jupyter development environment due to the ease of implementation. Because of the looping required, Python is not the most efficient language for this application, and the development of an executable program would be preferable if the program is adapted for commercial purposes. It is recommended that for continued use, a FORTRAN implementation as standard for GSLIB programs is pursued for the heuristic stochastic stope optimizer.

5.3 Summary

In the above sections, a heuristic optimization algorithm is developed. The benefits of the algorithm are proven both in terms of the correct calculation of non-linearities in objective functions and the added ability to directly consider risk during optimization. This improves a critical aspect of mine evaluation and planning, leading to a better realization of extraction value.

A case study is completed with the novel algorithm, showing the ability for risk evaluation and the improved calculation of non-linear objectives with a 17% increase in value when calculation over realizations is considered.

The thesis statement from Chapter 1 states: "*The development and implementation of a stochastic stope optimization algorithm allow for the correct consideration of non-linear functions and the direct inclusion of risk in the optimization process, improving decision-making when designing stope layouts.*" The body of this work develops a heuristic algorithm, which represents a real improvement in the ability for the consideration of non-linearities and risk in stope layout optimization, achieving all indicated goals. This is shown through the case study implementation where improvements due to the correct assessment non-linearities are detailed, and the consideration of risk is leveraged to implement a decision-making workflow that would not be possible without the generation of risk-rated layouts.

References

Acorn, T., & Deutsch, C. V. (2018). Optimizing Pit Shells in the Presence of Geologic Uncertainty with a Heuristic Algorithm. *Transactions*, *344*(1), 38–47. https://doi.org/10.19150/trans.8747

Alford, C. (2013). *Mineable Shape Optimizer*. Alford Mining Systems. http://alfordminingsystems.com/

Alford, C., Brazil, M., & Lee, D. H. (2007). *Handbook Of Operations Research In Natural Resources*. *99*(March 2015). https://doi.org/10.1007/978-0-387-71815-6

Bleichrodt, H., & Wakker, P. P. (2015). Regret theory: A Bold Alternative to the Alternatives. *Economic Journal*, *125*(583), 493–532. https://doi.org/10.1111/ecoj.12200

Bootsma, M. T., Alford, C., Benndorf, J., & Buxton, M. W. N. (2018). Cut-Off Grade Based Sublevel Stope Mine Optimisation. In *Advances in Applied Strategic Mine Planning* (pp. 537–557). Springer International Publishing. https://doi.org/10.1007/978-3-319-69320-0_31

Bullock, R. L. (2011). Comparison of Underground Mining Methods. In P. Darling (Ed.), *SME Mining Engineering Handbook* (3rd ed., pp. 385–403). Society for Mining, Metallurgy, and Exploration (SME). https://app.knovel.com/hotlink/khtml/id:kt008JZC6B/sme-mining-engineering/comparison-underground

Chiquini, A. P. (2018). *Mineral Resources Evaluation with Mining Selectivity and Information Effect* [University of Alberta]. https://doi.org/https://doi.org/10.7939/R3H41K333

Chiquini, A. P., & Deutsch, C. V. (2018). Underground Mining Considerations on Information and Mining Selectivity Effects and Case Study. *CCG Annual Report*, *20*. http://www.ccgalberta.com/

Deutsch, C. V. (2018). All Realizations All the Time. In B. S. Daya Sagar, Q. Cheng, & F. Agterberg (Eds.), *Handbook of Mathematical Geosciences* (pp. 131–142). Springer International Publishing. https://doi.org/10.1007/978-3-319-78999-6_7

Dimitrakopoulos, R. (2018). Stochastic Mine Planning---Methods, Examples and Value in an Uncertain World. In R. Dimitrakopoulos (Ed.), *Advances in Applied Strategic Mine Planning* (pp. 101–115). Springer International Publishing. https://doi.org/10.1007/978-3-319-69320-0_9

Dimitrakopoulos, R., & Grieco, N. (2009). Stope Design and Geological Uncertainty: Quantification of Risk in Conventional Designs and a Probabilistic Alternative. *Journal of Mining Science*, *45*(2), 152–163. https://doi.org/10.1007/s10913-009-0020-y

Elton, E. J., Gruber, M. J., Brown, S. J., & Goetzmann, W. N. (2014). *Modern Portfolio Theory and Investment Analysis* (9th ed.). Wiley.

Erdogan, G., Cigla, M., Topal, E., & Yavuz, M. (2017). Implementation and Comparison of Four Stope Boundary Optimization Algorithms in an Existing Underground Mine. *International Journal of Mining, Reclamation and Environment*, *31*(6), 389–403. https://doi.org/10.1080/17480930.2017.1331083

Godoy, M. (2018). A Risk Analysis Based Framework for Strategic Mine Planning and Design---

Method and Application. In R. Dimitrakopoulos (Ed.), *Advances in Applied Strategic Mine Planning* (pp. 75–90). Springer International Publishing. https://doi.org/10.1007/978-3-319-69320-0_7

Journel, A. G. (2018). Roadblocks to the Evaluation of Ore Reserves---The Simulation Overpass and Putting More Geology into Numerical Models of Deposits. In R. Dimitrakopoulos (Ed.), *Advances in Applied Strategic Mine Planning* (pp. 47–55). Springer International Publishing. https://doi.org/10.1007/978-3-319-69320-0_5

Journel, A. G., & Huijbregts, C. J. (1978). *Mining Geostatistics*. Academic Press.

Khalokakaie, R., Dowd, P. A., & Fowell, R. J. (2000). Lerchs–Grossmann algorithm with variable slope angles. *Mining Technology*, *109*(2), 77–85. https://doi.org/10.1179/mnt.2000.109.2.77

Khosrowshahi, S., Shaw, W. J., & Yeates, G. A. (2018). Quantification of Risk Using Simulation of the Chain of Mining---Case Study at Escondida Copper, Chile. In R. Dimitrakopoulos (Ed.), *Advances in Applied Strategic Mine Planning* (pp. 57–74). Springer International Publishing. https://doi.org/10.1007/978-3-319-69320-0_6

Lane, K. F. (2016). *The Economic Definition of Ore: Cut-off Grades in Theory and Practice*. COMET Strategy Pty Ltd.

Little, J., Knights, P., & Topal, E. (2013). Integrated Optimization of Underground Mine Design and Scheduling. *Journal of the Southern African Institute of Mining and Metallurgy*, *113*(10), 775–785. https://www.onemine.org/document/document.cfm?docid=208704&docorgid=9

Loomes, G., & Sugden, R. (1982). Regret Theory: An Alternative Theory of Rational Choice Under Uncertainty. *The Economic Journal*, *92*(368), 805–824. https://doi.org/10.2307/2232669

Manchuk, J. G. (2007). *Stope Design and Sequencing*. University of Alberta.

Manchuk, J. G., & Deutsch, C. V. (2012). A Flexible Sequential Gaussian Simulation Program: USGSIM. *Computers & Geosciences*, *41*, 208–216. https://doi.org/https://doi.org/10.1016/j.cageo.2011.08.013

Markowitz, H. (1952). PORTFOLIO SELECTION*. *The Journal of Finance*, *7*(1), 77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

Maybee, B., Fava, L., Dunn, P. G., Wilson, S., & Fitzgerald, J. (2010). *Toward Optimum Value in Underground Mine Scheduling*. Canadian Institute of Mining, Metallurgy and Petroleum.

Menabde, M., Froyland, G., Stone, P., & Yeates, G. A. (2018). Mining Schedule Optimisation for Conditionally Simulated Orebodies. In R. Dimitrakopoulos (Ed.), *Advances in Applied Strategic Mine Planning* (pp. 91–100). Springer International Publishing. https://doi.org/10.1007/978-3-319-69320-0_8

Monkhouse, P. H. L., & Yeates, G. A. (2018). Beyond Naïve Optimisation. In R. Dimitrakopoulos (Ed.), *Advances in Applied Strategic Mine Planning*. https://doi.org/10.1007/978-3-319-69320-0_1

Nehring, M., & Topal, E. (2011). Production schedule optimisation in underground hard rock mining using mixed integer programming. *Australasian Institute of Mining and Metallurgy Publication Series*. https://www.researchgate.net/publication/43478546_Production_schedule_optimisation_in_underground_hard_rock_mining_using_mixed_integer_programming

Nelson, M. G. (2011). Evaluation of Mining Methods and Systems. In P. Darling (Ed.), *SME Mining Engineering Handbook* (3rd ed., pp. 341–348). Society for Mining, Metallurgy, and Exploration (SME). https://app.knovel.com/hotlink/khtml/id:kt008JZ9G1/sme-mining-engineering/evaluation-mining-methods

Neufeld, C., Leuangthong, O., & Deutsch, C. V. (2007). A Simulation Approach to Account for the Information Effect. *Centre for Computational Geostatistics*, *21*. http://www.ccgalberta.com/

Nhleko, A. S., Tholana, T., & Neingo, P. N. (2018). A Review of Underground Stope Boundary Optimization Algorithms. *Resources Policy*, *56*(January), 59–69. https://doi.org/10.1016/j.resourpol.2017.12.004

Nikbin, V., Ataee-pour, M., Shahriar, K., & Pourrahimian, Y. (2018). A 3D Approximate Hybrid Algorithm for Stope Boundary Optimization. *Computers & Operations Research*, *115*, 104475. https://doi.org/10.1016/j.cor.2018.05.012

Nikbin, V., Ataee-pour, M., Shahriar, K., Pourrahimian, Y., & MirHassani, S. A. (2019). Stope Boundary Optimization: A Mathematical Model and Efficient Heuristics. *Resources Policy*, *62*(October 2018), 515–526. https://doi.org/10.1016/j.resourpol.2018.10.007

Noble, A. C. (2011). Mineral Resource Estimation. In P. Darling (Ed.), *SME Mining Engineering Handbook* (3rd ed.). Society for Mining, Metallurgy, and Exploration (SME). https://app.knovel.com/hotlink/khtml/id:kt008JZ3P3/sme-mining-engineering/resource-estimation-methodology

Pakalnis, R., & Hughes, P. (2011). Sublevel Stoping. In P. Darling (Ed.), *SME Mining Engineering Handbook* (3rd ed.). Society for Mining, Metallurgy, and Exploration (SME). https://app.knovel.com/hotlink/pdf/id:kt008K10G1/sme-mining-engineering/sublevel-stoping-requirements

Ramazan, S., & Dimitrakopoulos, R. (2018). Stochastic Optimisation of Long-Term Production Scheduling for Open Pit Mines with a New Integer Programming Formulation. In R. Dimitrakopoulos (Ed.), *Advances in Applied Strategic Mine Planning* (pp. 139–153). Springer International Publishing. https://doi.org/10.1007/978-3-319-69320-0_11

Rosado, R. C., Costa, J. F. C. L., & Saldanha, A. A. (2019). A Parallel Composite Genetic Algorithm for Mine Scheduling. In C. Mueller, W. Assibey-Bonsu, E. Baafi, C. Dauber, C. Doran, M. J. Jaszczuk, & O. Nagovitsyn (Eds.), *Mining goes Digital* (pp. 286–291). CRC Press. https://doi.org/https://doi.org/10.1201/9780429320774

Rossi, M. E., & Deutsch, C. V. (2014). *Mineral Resource Estimation*. Springer Netherlands. https://doi.org/10.1007/978-1-4020-5717-5

Sandanayake, D. S. S. (2014). *Stope Boundary Optimisation in Underground Mining Based on a Heuristic Approach* (Issue August). http://hdl.handle.net/20.500.11937/1725

Sari, Y. A., & Kumral, M. (2019). Incorporating grade uncertainty into sublevel stope sequencing. In C. Mueller, W. Assibey-Bonsu, E. Baafi, C. Dauber, C. Doran, M. J. Jaszczuk, & O. Nagovitsyn (Eds.), *Mining goes Digital* (pp. 323–327). CRC Press. https://doi.org/https://doi.org/10.1201/9780429320774

Sotoudeh, F., Ataei, M., Kakaie, R., & Pourrahimian, Y. (2019). Application of Sequential Gaussian Conditional Simulation to Underground Mine Design Under Grade Uncertainty. *Journal of Mining and Environment*. https://doi.org/10.22044/jme.2019.7333.1582

Topal, E., & Sens, J. (2010). A new Algorithm for Stope Boundary Optimization. *Journal of Coal Science and Engineering (China)*, *16*(2), 113–119. https://doi.org/10.1007/s12404-010-0201-y

Verhoeff, R. L. A. (2017). *Using Genetic Algorithms for Underground Stope Design Optimization in Mining: A Stochastic Analysis*. *January*. uuid:d626cc7b-4fb6-40f7-8f0e-b42b3d9d753f

Villaescusa, E. (2014). Geotechnical Design for Sublevel Open Stoping. In *Geotechnical Design for Sublevel Open Stoping* (1st ed.). CRC Press. https://doi.org/10.1201/b16702

Villalba Matamoros, M. E., & Kumral, M. (2018). Underground Mine Planning: Stope Layout Optimisation Under Grade Uncertainty Using Genetic Algorithms. *International Journal of Mining, Reclamation and Environment*, *0930*, 1–18. https://doi.org/10.1080/17480930.2018.1486692

Vulcan Envisage (11.0.2). (2018). Maptek. https://www.maptek.com/vulcan11/

Whittle, J. (2018). The Global Optimiser Works---What Next? In R. Dimitrakopoulos (Ed.), *Advances in Applied Strategic Mine Planning* (pp. 31–37). Springer International Publishing. https://doi.org/10.1007/978-3-319-69320-0_3

Wilson, B., Acorn, T., & Boisvert, J. (2019). Pit Optimization on Clustered Realizations: Identifying Functional Scenarios. *Mining, Metallurgy & Exploration*, *36*(6), 1221–1234. https://doi.org/10.1007/s42461-019-0101-7

Zarate, E. U., Pourrahimian, Y., & Boisvert, J. (2020). Optimizing Block Caving Drawpoints Over Multiple Geostatistical Models. *International Journal of Mining, Reclamation and Environment*, *34*(1), 55–74. https://doi.org/10.1080/17480930.2018.1532866

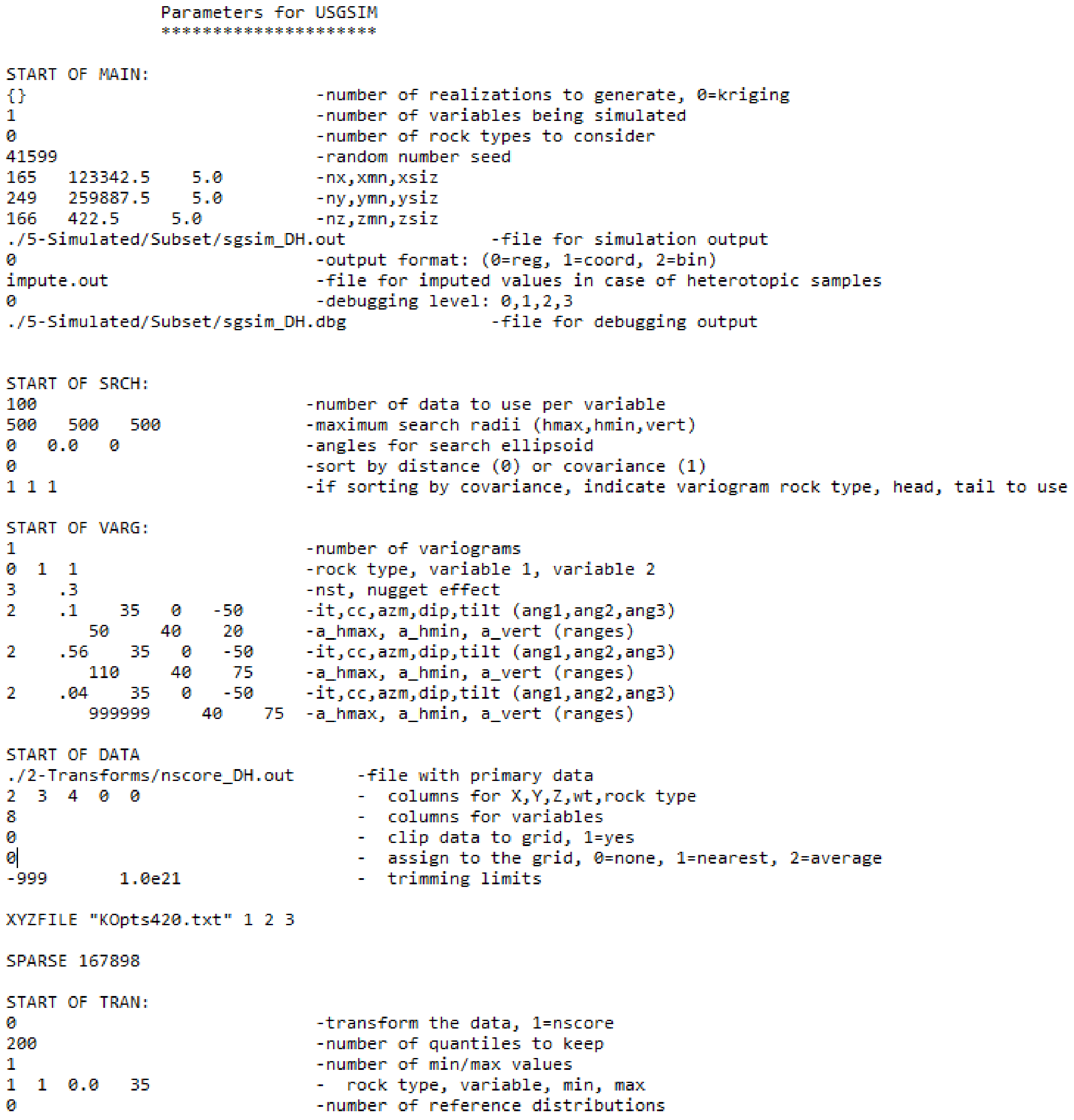
Appendix A

Description of input parameters

Parameter	Symbol(s)	Description
Stope size minima (blocks)	$xmin, ymin, zmin$	Minimum stope dimensions – determined by equipment restrictions
Stope size maxima (blocks)	$xmax, ymax, zmax$	Maximum stope dimensions – determined by geotechnical constraints
Model size	$Xdim, Ydim, Zdim$	Dimensions of the input geostatistical grade model
Model origin	$Xmin, Ymin, Zmin$	Dimensional origin of the input model
Block sizes	Xsz, Ysz, Zsz	Size of blocks in each dimension
Pillars	$xpil, ypil, zpil$	Required dimension of pillars in each dimension required for geotechnical stability
Risk scaling factor	α	Multiplicative penalty factor α applied in objective calculations
Mining cost per block	C_m	Cost required to mine a block – converted from cost per tonne
Processing cost per block	C_p	Cost of processing a single block – converted from cost per tonne
Metal price	P	Projected metal price for optimization
Density	γ	Value for density of all blocks in model
Recovery	Function	Function defining process recovery

Appendix B

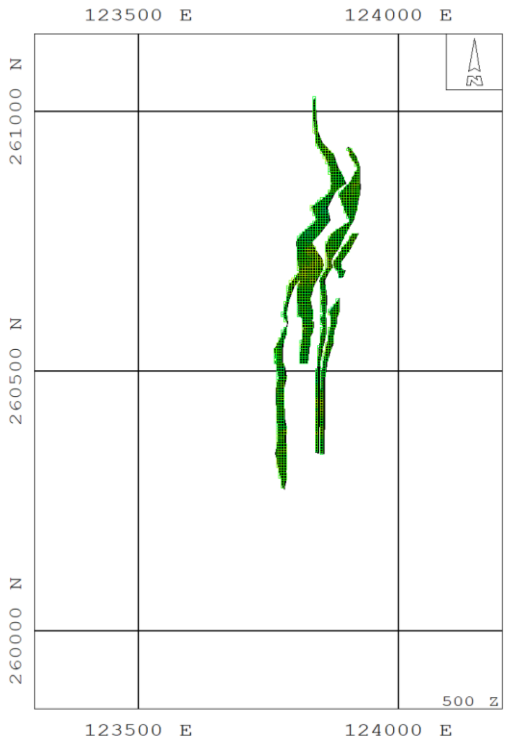
USGSIM Parameter File



```
               Parameters for USGSIM
               *********************

START OF MAIN:
{}                            -number of realizations to generate, 0=kriging
1                             -number of variables being simulated
0                             -number of rock types to consider
41599                         -random number seed
165   123342.5    5.0         -nx,xmn,xsiz
249   259887.5    5.0         -ny,ymn,ysiz
166   422.5     5.0           -nz,zmn,zsiz
./5-Simulated/Subset/sgsim_DH.out              -file for simulation output
0                             -output format: (0=reg, 1=coord, 2=bin)
impute.out                    -file for imputed values in case of heterotopic samples
0                             -debugging level: 0,1,2,3
./5-Simulated/Subset/sgsim_DH.dbg              -file for debugging output


START OF SRCH:
100                          -number of data to use per variable
500   500   500              -maximum search radii (hmax,hmin,vert)
0   0.0   0                  -angles for search ellipsoid
0                            -sort by distance (0) or covariance (1)
1 1 1                        -if sorting by covariance, indicate variogram rock type, head, tail to use

START OF VARG:
1                            -number of variograms
0  1  1                      -rock type, variable 1, variable 2
3    .3                      -nst, nugget effect
2    .1    35   0   -50      -it,cc,azm,dip,tilt (ang1,ang2,ang3)
        50     40    20      -a_hmax, a_hmin, a_vert (ranges)
2    .56   35   0   -50      -it,cc,azm,dip,tilt (ang1,ang2,ang3)
        110     40    75     -a_hmax, a_hmin, a_vert (ranges)
2    .04   35   0   -50      -it,cc,azm,dip,tilt (ang1,ang2,ang3)
        999999     40    75  -a_hmax, a_hmin, a_vert (ranges)

START OF DATA
./2-Transforms/nscore_DH.out      -file with primary data
2  3  4  0  0                     -  columns for X,Y,Z,wt,rock type
8                                 -  columns for variables
0                                 -  clip data to grid, 1=yes
0                                 -  assign to the grid, 0=none, 1=nearest, 2=average
-999       1.0e21                 -  trimming limits

XYZFILE "KOpts420.txt" 1 2 3

SPARSE 167898

START OF TRAN:
0                             -transform the data, 1=nscore
200                           -number of quantiles to keep
1                             -number of min/max values
1  1  0.0   35                -  rock type, variable, min, max
0                             -number of reference distributions
```

Supplementary Figures

Plan Views:



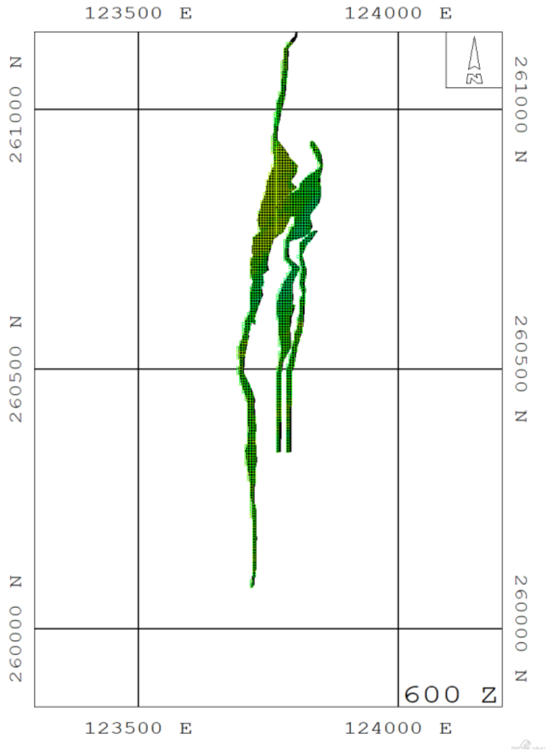
LEGEND

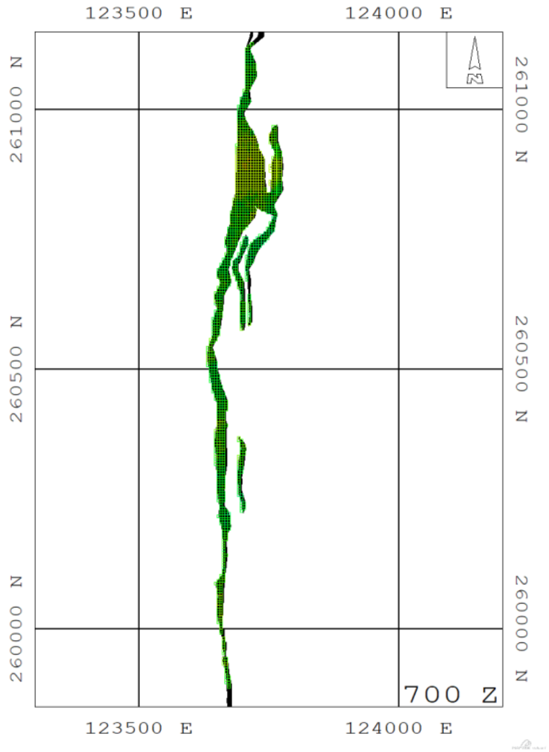
Block model

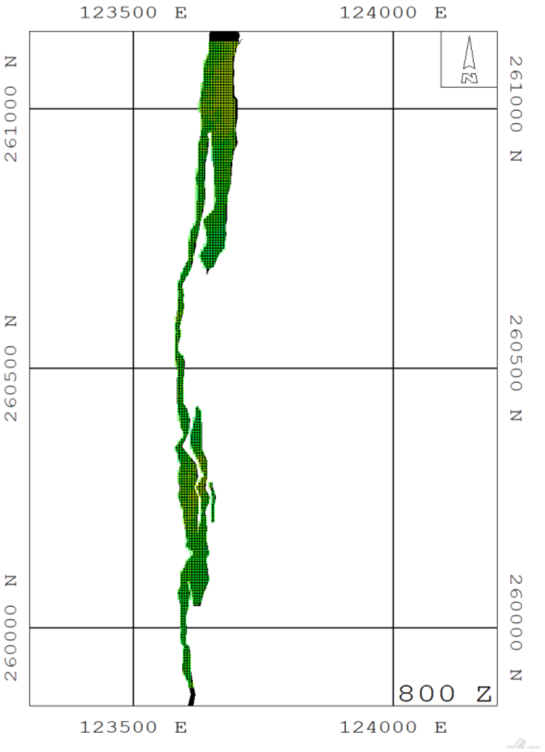
File MODCHKNS.BMF

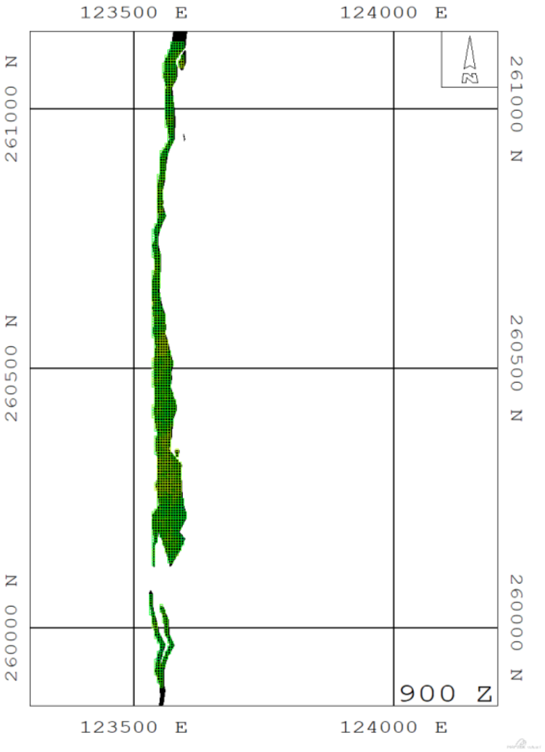
Block scheme
Scheme ASSAY

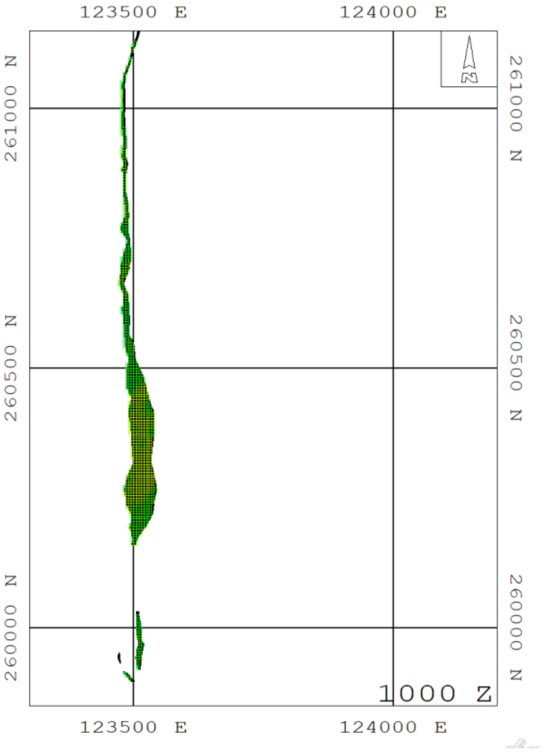
0.000		0.500
0.500		1.000
1.000		2.000
2.000		3.000
3.000		4.000
4.000		5.000
5.000		10.000
10.000		20.000
20.000		30.000
30.000		40.000
40.000		100.000

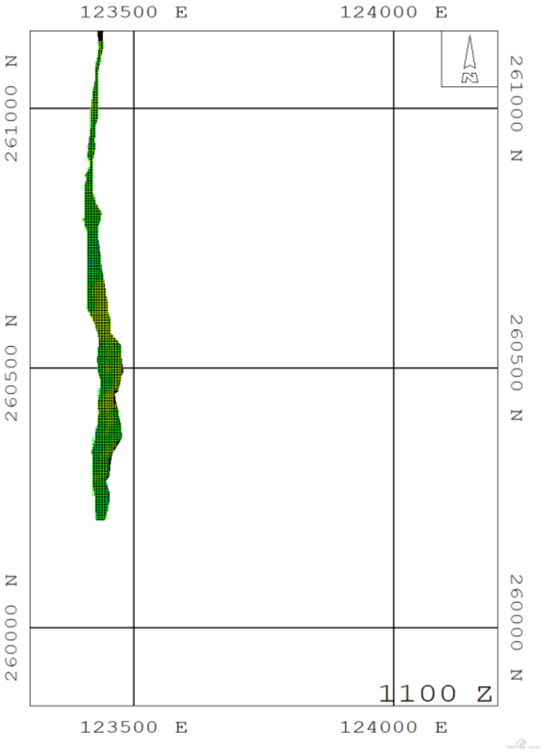




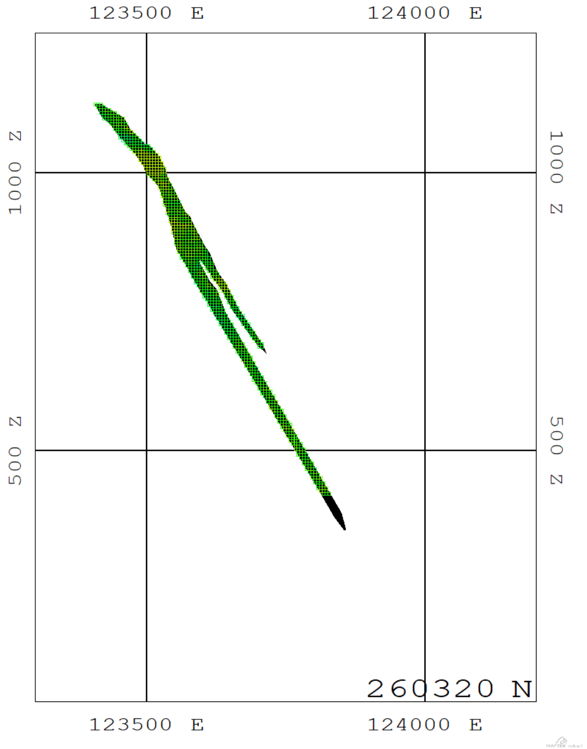

123500 E
124000 E
261000 N
260500 N
260000 N
800 Z

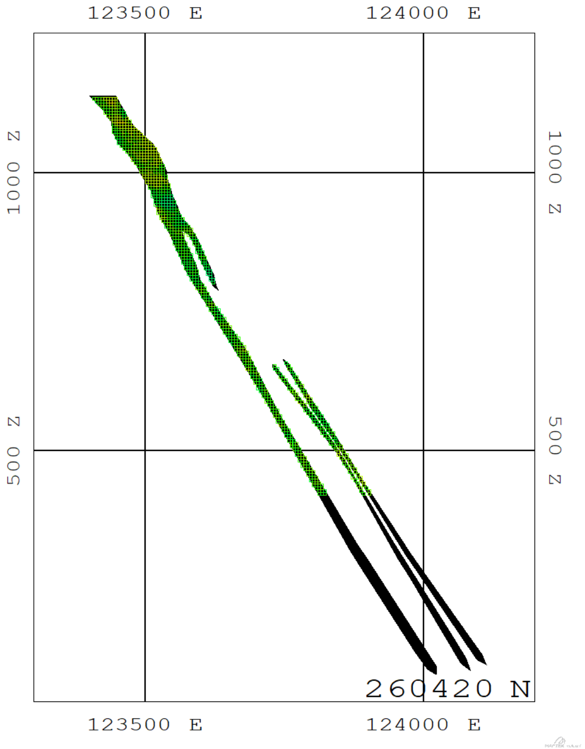

123500 E
124000 E
261000 N
260500 N
260000 N
900 Z

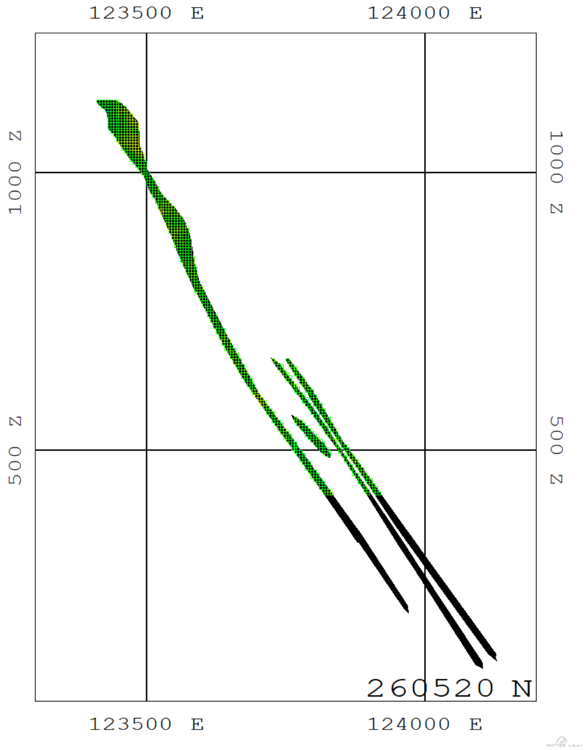

123500 E
124000 E
261000 N
260500 N
260000 N
1000 Z

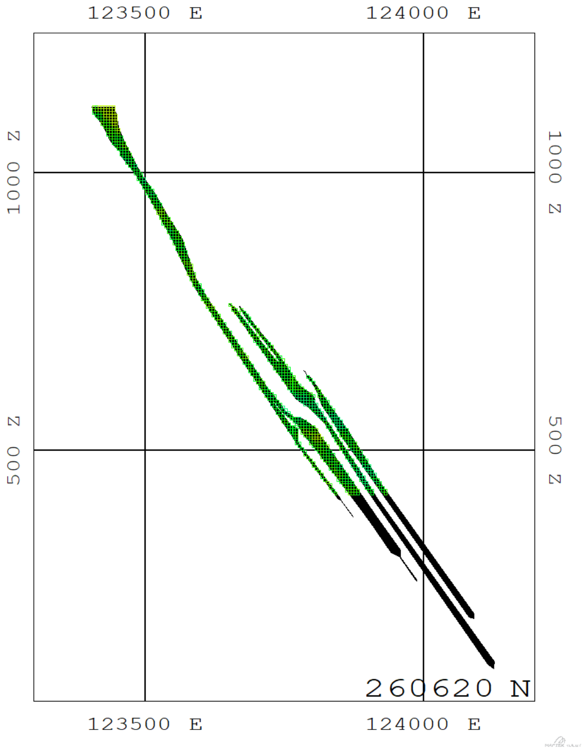

123500 E
124000 E
261000 N
260500 N
260000 N
1100 Z

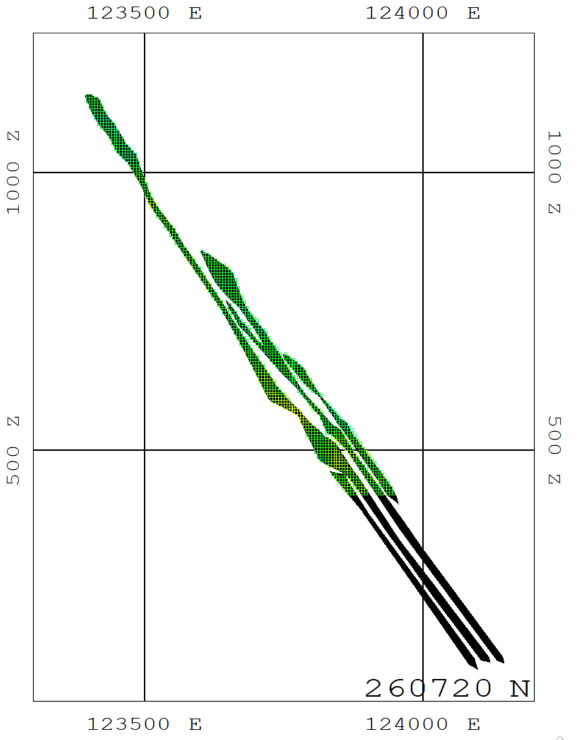
Section Views:

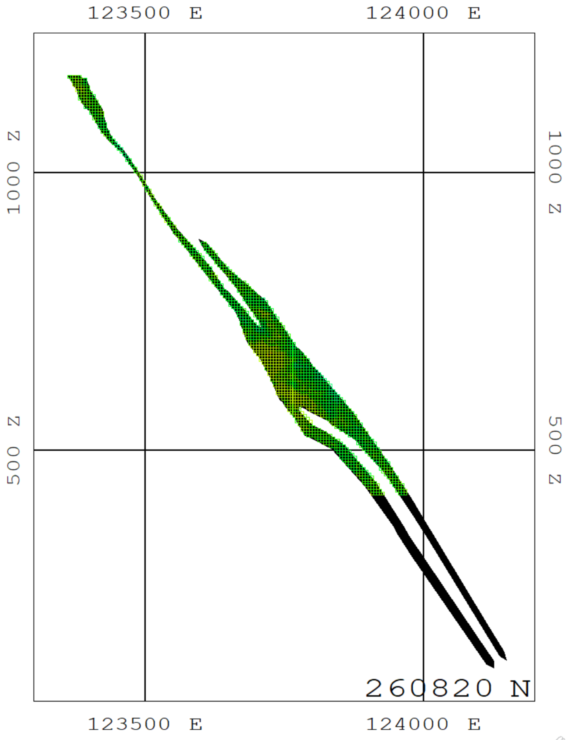









123500 E
124000 E
1000 Z
500 Z
1000 Z
500 Z
260720 N
123500 E
124000 E


123500 E
124000 E
1000 Z
500 Z
1000 Z
500 Z
260820 N
123500 E
124000 E