

Using Genetic Algorithms for Underground Slope Design Optimization in Mining

A Stochastic Analysis

by

R.L.A. Verhoeff

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on April 24, 2017

Student number: 4237471
Project duration: February 2016 – January 2017
Thesis committee: Prof. J. Benndorf TU Bergakademie Freiberg
Dr. M.W.N. Buxton TU Delft
A. Matthaus Msc RWTH Aachen

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Mine design and mine planning are essential to a mining operation as they do not only dictate the outline of the mine, but essentially determine the financial robustness and success of the mining operation. However, as mining is a conservative business mine design is still mostly done by hand. To indicate the effect of uncertainty in mine design, Vallee (2000) [29] reported that 60% of the reviewed mines had an average production rate that was 30% lower than their designed capacity. To provide in better mine design researchers and companies have developed and proposed number of guidance tools in recent years to help optimize mine planning. However their efforts have mainly focussed on open-pit mine design whilst largely ignoring underground mining, which is more versatile and therefore more difficult to assess. Furthermore, the optimization algorithms that have been developed are mostly based on average-type or interpolated resource models, which do not allow assessing uncertainty. Several stochastic approaches have been proposed by Dimitrakopoulos [36], [24], [37] and others [30], as a tool to involve stochastic techniques in mine design optimization as a way to assess uncertainty. However, few practical stochastic solutions have been developed for underground mine design. To contribute to the recent efforts made by the TU Delft in the 'Horizon 2020 – Real Time Mining' European Union Research and Innovation Programme, this research has focussed on combining underground mine design optimization with several stochastic analyses.

A stope layout optimizer was developed in Matlab based on standard genetic algorithms, a sub-group of evolutionary algorithms. This model was tested on two validation data sets where it showed good optimization performance. A stochastic optimization module was added and tested, but as the validation data set was linear, it showed no added value. Using a resource model of a copper / zinc VMS deposit a number of stochastic approaches were tested. By performing a minimal downside risk / maximum upside potential analysis [36] a stochastic optimization and elimination approach was tested under different economic scenarios by using 20 unconditional economic block model simulations. A stochastic risk analysis [37] using the same 20 economic block models was done to study the effect of stochastic probability on economic value, ore tonnage, waste tonnage, average ore grade and average arsenic content. Lastly, a stochastic design optimization was done using 100 unconditional economic block models and its optimization performance was compared to that for a traditional non-stochastic optimization.

Both the minimal downside risk / maximum upside potential analysis and the stochastic design analysis showed better optimization results than traditional optimization using average type orebody models. However, as in the min / max approach emphasis was on robustness instead of profitability, it was unable to show the same degree of improvement over traditional optimization

Contents

List of Figures	ix
List of Tables	xi
1 Introduction and problem statement	1
1.1 Research context	1
1.2 Problem statement	3
1.3 Motivation	3
1.4 Aim and objectives	4
1.5 Hypothesis and research questions	4
1.6 Report outline.	4
2 Description of problem case	7
2.1 General geologic scenario	7
2.2 Typical mining methods	8
2.2.1 Bench-and-fill	8
2.2.2 Drift-and-fill	8
2.3 Problem statement	9
2.4 Optimization considerations	9
2.4.1 Key optimization parameters	9
2.4.2 Objective function	10
2.4.3 Constraints	10
3 Theoretical background	11
3.1 Global Mathematical Optimization	11
3.1.1 Metaheuristics	11
3.2 Genetic algorithms	13
3.2.1 Why genetic algorithms?	13
3.2.2 History.	13
3.2.3 Overview of general genetic algorithm	14
3.2.4 Solution representation (candidates)	14
3.2.5 Selection and measure of optimality	15
3.2.6 Recombination (Cross-over)	16
3.2.7 Mutation.	17
3.3 Conditional resource modelling and simulation	19
3.3.1 Geostatistical Interpolation and kriging	19
3.3.2 Conditional Simulation of orebody models	20
3.3.3 Stochastic modelling.	20
3.3.4 Methods of stochastic simulation	20
3.3.5 Turning bands simulation	20
3.3.6 Sequential simulation	21
3.3.7 Stochastic simulations for risk analyses	21
4 Model	23
4.1 Optimizer model	23
4.1.1 Model characteristics	23
4.1.2 Optimization assumptions.	23
4.1.3 Coding language	24
4.2 Description of the resource model	24
4.2.1 Method of orebody modelling	24
4.2.2 Delivered results	25

4.3	Model Schematic	25
4.4	Model data Input	27
4.4.1	Excel Import	27
4.4.2	ASCII Import	27
4.4.3	CSV Import	28
4.5	Economic block model creation.	28
4.5.1	Development costs	29
4.5.2	Arsenic content penalty function	29
4.5.3	Application of cut-off grades	29
4.5.4	Cost calculation parameters	30
4.6	Parent Generation	30
4.7	String Evaluation and selection	31
4.8	Cross-Over Algorithm	31
4.9	Mutation Algorithm	32
4.10	Constraint function Algorithm	33
4.11	Elitism	33
4.12	Input Parameters	33
5	Model validation and validation results	37
5.1	Model validation	37
5.1.1	Aim of validation.	37
5.1.2	Validation procedure.	37
5.2	Model validation results.	39
5.2.1	Validating using Excel import	39
5.2.2	Validating using ASCII import	42
5.2.3	CSV File adaptation	47
5.2.4	Zinc cost analysis	51
6	Methodology	53
6.1	Introduction	53
6.2	Analysis scope	53
6.3	KPPI's	53
6.4	Maximum upside potential / Minimal downside risk optimization	54
6.5	Stochastic risk analysis	54
6.6	Stochastic design optimization	55
7	Results	57
7.1	Stochastic analyses	57
7.1.1	Maximum Upside / Minimum Downside Analysis	58
7.1.2	Base-case	58
7.1.3	Price-sensitivity analysis	59
7.1.4	Penalty value impact analysis	60
7.2	Stochastic risk analysis	61
7.2.1	Stope inclusion probabilities.	62
7.2.2	Risk Profile analyses	64
7.3	Stochastic Design Optimization.	67
8	Discussion of results	71
8.1	Stochastic analyses discussion	71
8.1.1	Minimum Downside Risk / Maximum Upside Potential Analysis	71
8.1.2	Stochastic Risk Analysis	72
8.1.3	Stochastic Design Optimization	75
8.2	Comparing the Min / Max analysis with the Stochastic analysis.	75

9 Conclusion	79
10 Recommendations and further work	81
Bibliography	83
A Flowchart Legend	87
B Flowcharts block model creator	89
C Flowcharts optimizer model	91
D Model parameters and variables	101
E Economic baseline blockmodels analysis 1 and 2	105
F Stope layout analysis 1 and 2	109
G Excel Files	113
H Overview of economic block calculations	119

List of Figures

2.1	Overview of VMS, from [10]	7
2.2	Bench-and-fill mining, from [11]	8
2.3	Drift-and-fill mining, from [2]	9
3.1	Taxonomy of global optimization algorithms, from [55]	12
3.2	Metaheuristics classification, from [8]	12
3.3	Roulette wheel, from [15]	16
3.4	Single point cross-over, from [16]	17
3.5	Two point cross-over, from [18]	17
3.6	Global and local maximum, from [13]	18
3.7	Flip-Bit Mutation, from [14]	18
3.8	Swap Mutation, from [14]	18
3.9	Scramble Mutation, from [14]	19
3.10	Inversion Mutation, from [14]	19
4.1	Schematic flowchart of real structured genetic algorithm	26
5.1	Orebody	40
5.2	parent grid 1	40
5.3	Performance of individuals best and worst	40
5.4	Ranked individual 1 (Best)	41
5.5	Ranked individual 40 (Worst)	41
5.6	Performance of individuals - no elitism	41
5.7	Performance of individuals - single elitism	42
5.8	Performance of individuals - double elitism	42
5.9	Real scenario elevation model	43
5.10	Elevation simulation 128	43
5.11	E-type (200) elevation model	43
5.12	Best individual solution - real case	44
5.13	Real scenario elevation model	44
5.14	Performance of individuals - best and worst	44
5.15	Best solution E-type	45
5.16	Performance graphs	45
5.17	Stochastic display of best individual solution projected over all orebody simulations	46
5.18	Histograms and normal distributions E-type and stochastic case	47
5.19	Normal distributions E-type and stochastic case	47
5.20	Independent data-layers	49
5.21	economic block model	50
5.22	No arsenic penalty	51
5.23	Penalty value 2500	51
5.24	Zinc price x 10	52
5.25	Zinc price x 100	52
7.1	Stochastic solutions for 20 random simulations	58
7.2	Stochastic solutions for 5 best random simulations	59
7.3	Metal price +25%	60
7.4	Metal price +50%	60
7.5	Metal price -25%	60
7.6	Metal price -50%	60

7.7	Arsenic penalty x2	61
7.8	Boxplot arsenic penalty x2	61
7.9	Arsenic penalty x5	61
7.10	Boxplot arsenic penalty x5	61
7.11	Stope inclusion probability map	62
7.12	Stope inclusion probability filters	63
7.13	Economic value	64
7.14	Ore tonnage	65
7.15	Waste tonnage	65
7.16	Average copper grade	66
7.17	Average arsenic grade	67
7.18	Histograms and normal distribution for all cases	68
7.19	Boxplot analysis 3	68
7.20	Normal distributions	69
8.1	Metal price –25%	72
8.2	Metal price –50%	72
8.3	Comparison Min/Max analysis with E-type analysis and stochastic analysis	76
A.1	Flowchart legend	88
B.1	Flowcharts block model creator	90
C.1	Model overview	92
C.2	Excel Import	93
C.3	ASCII Import	94
C.4	CSV Import	95
C.5	Parentstring Generator	96
C.6	Cross-Over Algorithm	97
C.7	Mutation	98
C.8	Constraint Function	99
G.1	Analysis 1 Cross value stope envelopes 20	113
G.2	Analysis 1 Cross value stope envelopes 20	113
G.3	Analysis 2 Average grade arsenic risk	114
G.4	Analysis 2 Average grade copper risk	115
G.5	Analysis 2 Economic risk	116
G.6	Analysis 2 Ore Tonnage risk	117
G.7	Analysis 2 Waste Tonnage risk	118

List of Tables

4.1	Neves global statistics of As (ppm), Cu and Zn by TB block simulations	25
4.2	Cost calculation parameters	30
4.3	Explanatory list of input parameters	34
4.4	Input parameters economic	35
5.1	Parameter values for model validation	39
5.2	Parameter values for calculating the economic block models	48
5.3	Penalty values	51
6.1	Key Project Performance Indicators	54
7.1	Optimizer model	57
7.2	Economic block model creator	58
7.3	Changed parameters	59
8.1	comparing mean, standard deviation and percentile	76
D.1	Explanatory list of input parameters	101
D.2	Explanatory list of model output variables	102
D.3	Explanatory list of model output values	103
D.4	Input parameters economic	104

Introduction and problem statement

1.1. Research context

In general, mine planning can be described as designing the optimal mine-layout that tries to maximize ore production at minimal costs. Proper mine planning is essential as it not only dictates the outline of the mine but essentially determines the financial robustness and success of the mining operation. The main input for mine planning is the geological interpretation of the orebody as this will dictate the scale of the mining operation and the method of extraction. Other subsequent inputs are generally: a geotechnical orebody model, mining equipment and equipment costs and processing constraints. Furthermore, commodity price is important as different market scenarios greatly affect the optimal layout and extraction sequence.

In traditional mine planning all processes leading to the final mine design and schedule are done manually by a team of geologists and mining engineers. For most of the bigger mining operations, companies have aimed to improve the decision making process by introducing mine design and planning software. These have been around since the early 1980's, but it was not until the introduction of windows 95 in the mid 1990's that they became widely available [9].

Where open pit mining is concerned the pit design algorithms are almost without exception based on the Lerchs & Grossmann algorithm. One of the most notable software derivatives of this concept being Whittle 3D [40].

For underground mining it was not until 1995 that Ovanic and Young presented a mixed integer programming formulation for stope boundary optimization [48]. This was quickly followed by Alford's floating stope concept for stope optimization [20] which has since been widely applied and extended.

However, as mining software tends to be expensive and the mining business generally conservative, software applications are not used in all mining operations. Especially in underground mining, where the mines tend to be more complex and their design is usually unique, software solutions are not always used on a large scale yet.

Underground mine design in general revolves around the placement of stopes. These stopes are units (voids) that can be mined in one series while staying within the ore body's geotechnical constraints. They can be approached as the smallest mineable unit given the mining equipment while ignoring infrastructure works. In order to maximise project value, it is therefore essential to optimize the stope layout.

Over the years a number of algorithms have been developed to solve the stope layout problem. These algorithms can either be classified as exact, meaning they are supported by mathematical proof, or heuristic. In contrast to exact algorithms, heuristic models do not have mathematical proof but aim to solve the given problem by restricting the search area using several constraints [55]. Some notable previous efforts on stope layout optimization are discussed below.

Ovanic and Young (1995) [48] Present a mixed integer programming formulation optimizing the stope boundaries along a single dimension. The algorithm therefore offers a partial solution to the stope layout problem and was one of the first developed for this purpose alone.

Alford (1995) [20] Developed and presented the floating stope concept for stope optimization. This approach has since found multiple applications in stope optimization software. However, the stopes generated by this algorithm are overlapping and therefore need post-optimization manual adjustment. The final outcome will therefore be sub-optimal.

Cawrse (2001) [32] introduced an extension to Alford's floating stope algorithm and recommends a multiple pass floating stope process (MPFSP). Even though the results show improvement, the model fails to overcome the shortcomings of Alford's model.

Ataee-pour (2000 and 2004) [21] [22] presents a heuristic method based on the maximum value neighbourhood (MVN). This algorithm offers a robust solution to the stope optimization problem and mitigates the overlapping stopes. However, the algorithm fails to produce equal results when the starting location within a block model is changed.

The MVN algorithm has been widely implemented in commercially available mining software and can therefore be used as a benchmark.

Sens and Topal (2009) [51] came up with a heuristic approach to stope layout optimization in 3D. In contrary to the floating stope algorithm it eliminates overlapping stopes. Its downside is that it only generates one unique set of stopes while, in reality, multiple solutions exist and the optimal solution should be selected among them.

Sandanayake, Topal and Asad (2015) [49] proposed a heuristic algorithm. Using an ore body model and its relevant economic parameters as an input, it translates the ore body model into an economic ore body block model. From this it develops the optimal stope layout by maximizing the total economic value subject to the geotechnical constraints. Unlike the floating stope model it does not allow for overlapping stopes. Furthermore the starting location should not influence the outcome. In tests the algorithm has shown to generate substantially higher economic potential than the commercially available MVN model from Ataee-pour.

A key characteristic for mine design methods, such as the ones described in the previous sections, is the use of complex optimization algorithms that assume a certain input. This input is in general an estimated and sometimes smoothed representation of the orebody, obtained through interpolation of drill hole data. Coupled with the uncertainty and in-situ variability allocated to those models these have been shown to be highly inaccurate. For example, Vallee (2000) [29] reports that 60% of the surveyed mines had an average production rate that was 30% lower than their designed capacity. These discrepancies between the actual production and the planning expectations arise through the uncertainty about the orebody in term of ore grade, tons and quality.

In the last decades it has been shown that implementing stochastic optimization techniques in mining can lead to a more realistic and robust mine design. Furthermore stochastic optimization techniques applied to mining can help in finding the optimal project NPV and production targets.

Dimitrakopoulos and Ramazan (2004) [35] propose a new optimization formulation based on linear programming for multi-element open pit mine production scheduling which integrates orebody uncertainty. Furthermore, the model also takes into account risk quantification, mobility restrictions and various operational requirements. With this approach they were able to produce better results than traditional approaches.

Dimitrakopoulos, Martinez and Ramazan (2007) [36] introduce the maximum upside potential / minimal downside risk approach where they focus on open pit mine optimization based on the assessment of geological uncertainties. Their approach allows the use of conventional optimization tools. By stochastically analysing different equally probable mine designs and their schedules they are able to select a final model that is best suited to handle uncertainties, while at the same time meeting several production targets. However, as they mention, the approach might be operationally tedious in the case of larger ore bodies.

Leite and Dimitrakopoulos (2007) [46] propose a multi-stage framework for open pit mine scheduling at a low-grade disseminated copper deposit. The framework considers geological uncertainty in order to minimize the risk of project target deviation. Using conditionally simulated ore body models ultimate pit limits and production schedules are generated. Using a simulated annealing algorithm a single production schedule is then generated based on the probability that a mining block belongs to a given scheduling period. Their results indicate that conventional approaches tend to wrongly estimate ore tonnages and NPV. Furthermore, the stochastic schedule has low chances to significantly deviate from project targets.

Benndorf and Dimitrakopoulos (2013) [24] present a stochastic integer programming based mine production scheduling approach which considers jointly multi-element geological uncertainty. This approach was tested on a large Australian open pit iron ore deposit. The results demonstrate the ability of the stochastic approach to control production target deviation risk. Furthermore, the stochastic schedule increases the probability of meeting production targets, therefore decreasing project risk and increasing project value.

Even though most previous researches have focussed on stochastic planning solutions for open pit mines, some efforts have been done to connect the technique to underground mine planning. The two most notable attempts being as follows.

Grieco and Dimitrakopoulos (2009) [37] propose a MIP programming approach for stope design in underground operations that determines the optimum location, size and number of stopes based on the concept of acceptable risk in a design they use an application on a Canadian base metal mine to demonstrate its practical aspects. These aspects include risk quantification for contained ore tonnes, grade and economic potential. They do note that as a result of mitigating risk, it might prove difficult to meet production demands and therefore requires waste blending.

Carpentier, Gamache and Dimitrakopoulos (2016) [30] propose a stochastic integer programming model to optimise long-term scheduling of underground operations while considering geological uncertainty. The two-stage model considers a variable cut-off grade and accounts for maximum development, material handling, flow conservation in a mine and mill capacities. Through the results they show the benefits of risk control such as an increased NPV and a shorter mine life. Furthermore they suggest using zones of variability in a flexible manner as to control risk while avoiding high-grading.

1.2. Problem statement

As the literature study indicates, mine design optimization has been the subject of research. Even though the sector is conservative, the recent economic crisis has contributed to the awareness that there is room for optimization in mining engineering and applying new solutions is crucial. As mining is a capital intensive business, small improvements can potentially save large amounts of investment costs.

However, most of the research done so far has been solely aimed at open pit mining, ignoring underground operations. As a result underground mine design is mostly still done by hand, and when computer aids have been introduced they still heavily rely on user guidance.

Another problem is that the current software solutions tend to optimize using single input. Usually a resource model derived through kriging or averaging is used as representation of the resource deposit. Even though this model might be a good enough representation of reality, it does not incorporate the uncertainty that is naturally present in those models.

The effects of uncertainties in mining design have been noted by Vallee and Bouchard (2000) [29], who found that 60% of the mines surveyed had an average production that was 30% below their designed capacity.

The capital intensive nature of mining means that failing to meet targets can have severe financial consequences. Furthermore, mining should strive to keep its environmental impact as small as reasonably possible. This would require mining only the valuable resource while leaving un-economic zones undisturbed. These goals can only be achieved if the amount of uncertainty in the design phase is decreased.

1.3. Motivation

As part of the Horizon 2020 Project, the biggest European Union Research and Innovation programme to date, the *'Real-Time Mining'* programme was started in 2015. The overall aim of Real-Time Mining is to develop a real time framework to decrease environmental impact and increase resource efficiency in the European raw material extraction industry. The project consists of 13 European partners from 5 countries and is led by the Resource Engineering Section of the Delft University of Technology assisted by an international External Expert Advisory Board. Among other sub-projects, it will include:

“research and demonstration activities integrating automated sensor based material characterization, online machine performance measurements, underground navigation and positioning, underground mining system simulation and optimization of planning decisions, state-of-the art updating techniques for resource/reserve models.” [3]

As part of this project investigations are to take place on optimal long-term planning and mine design.

This research will focus on combining mine design optimization with several stochastic analyses to quantify grade-risk and to explore the potential benefit of these techniques.

For optimization, the principles of genetic algorithms will be used to develop a stochastic stope layout optimizer in Matlab. Quantification of stochastic optimization will be done in regard to a number of Key Project Performance Indicators widely used for assessing mining operations.

1.4. Aim and objectives

The aim of this research was defined as:

“To introduce and evaluate an optimization approach for underground stope mining, based on genetic algorithms, taking into account grade uncertainty and to quantify grade uncertainty risk by stochastic analysis.”

To achieve this, the following objectives were defined:

- Develop a 2D stope layout optimizer based on a binary genetic algorithm.
- Convert the algorithm to incorporate stochastic input.
- Develop a module to convert the available stochastic ore-grade simulations into economic block model.
- Use the developed model to evaluate the orebody stochastically and investigate its potential benefits.

1.5. Hypothesis and research questions

The aforementioned research goal and objectives are part of the central hypothesis:

Hypothesis: *“Simulation based stochastic optimization of underground mine design will lead to better results than optimization based on customary average grade models.”*

The main research questions following from this hypothesis, which are to be answered in this research, have been stated as follows:

1. How can metaheuristic optimization be used in underground stope geometry optimization?
2. How can stochastic optimization of stope geometries, using metaheuristic optimization, be used to manage and eliminate uncertainty in underground mine design?

1.6. Report outline

This report consists of 4 main sections divided over 9 chapters.

- I The first section consists of chapter 1, 2 and 3. Chapter 1 introduces the research context by a summary of previous research, states the research motivation and goal, lists the aim and objectives and presents the hypothesis and research questions Chapter 2 describes the problem case by introducing the scenario on which this research was conducted and states what parameters were derived from this case. Chapter 3 provides a theoretical background on genetic algorithms, geological simulation and stochastic optimization.
- II The second section consists of chapter 4 and chapter 5. Chapter 4 states the model characteristics and scope, states the model assumptions and modelling input. This is followed by an extensive model description in which justification is given for the various modelling decisions. In chapter 5 the model validation procedure is documented.

III Section three consists of chapters 6, 7, 8 and 9 Chapter 6 contains the methodology. Here the analysis scope is stated and three independent analyses are introduced and their procedure listed. Chapter 7 contains the results of the three distinct stochastic analyses Chapter 8 discusses the highlights of the results and provides explanation for the different results. Chapter 9 concludes the research done by answering the research questions, reflecting on the previously stated research goals and hypothesis.

IV Section 4 consists of the recommendations (Chapter 10) and the appendices.

Description of problem case

2.1. General geologic scenario

The mineral deposit considered for this research is classified as volcano-sedimentary massive sulphides (VMS).

These deposits form on, or below, the ocean floor and are typically associated with volcanic rocks. They typically form during periods of rifting. As rifting causes extension and thinning of the crust this allows hot sub-volcanic magmas to rise and form metal-bearing hydrothermal cells. The location of the lenses are often controlled by volcanic faults and fissures which can permit a discharge of hydrothermal fluids on the seafloor (black smokers).

These deposits typically occur as lenses of poly-metallic massive sulphides rich in copper, zinc, tin and/or lead. The lenses are commonly underlain by sulphide-silicate stockwork vein systems, although the stockwork systems may extend into the hanging wall strata above the massive sulphide lenses. VMS deposits are often found to have a strong metal zonation showing as a segregated zones of various metal bearing sulphides throughout the deposit. In general chalcopyrite (copper sulphide) forms in the central part of the deposit. Zinc and lead sulphides (sphalerite and galena) tend to form in the more outlying deposit parts. Generally non-economic iron sulphides like pyrite and pyrrhotite occur within the base metal sulphides.

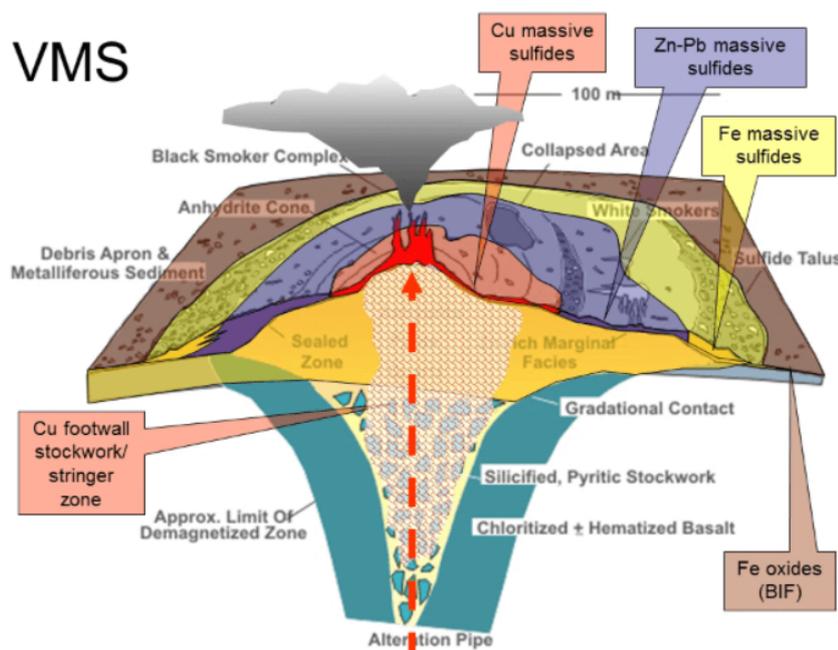


Figure 2.1: Overview of VMS, from [10]

Base metal grade distributions within the massive copper/zinc sulphide lenses typically show good internal continuity but can laterally terminate abruptly in barren pyrite. The massive sulphide deposits are generally very large, regular, continuous and predictable. However, the geometry of high-grade zinc and cop-

per zones within them can be complex due to geological alterations. In many cases boundaries between ore grade mineralization and barren pyrite may be almost parallel to the stratigraphic contact of the sulphide lenses.

2.2. Typical mining methods

Due to the large, but locally complex, high grade ore zones in VMS deposits they are usually mined using underground stoping techniques as this allows for a reasonable degree of selectability. In VMS orebodies with average geotechnical conditions and average to high grade copper or zinc ore the most common methods are bench-and-fill and drift-and-fill.

2.2.1. Bench-and-fill

Bench-and-fill mining is commonly used where the mineralization is of sufficient thickness and continuity. The method is highly productive and had relatively low operating costs compared to other stoping methods. It is generally applied in orebody regions where the thickness exceeds 20m.

The bench-and-fill stopes are accessed through the footwall with drives driven along the strikes in 20m vertical intervals. Upper and lower access crosscuts are driven across the orebody. The top access is opened up to the full stope width and a slot raise is opened at the hangingwall side of the stope. Vertical sections are then blasted on retreat to the footwall with material being recovered from the lower access drive.

Primary stopes have been mined up to 120m long, but are more typically broken up into sections of 30 – 40m. Primary stopes are usually backfilled with cemented paste fill and hydraulic sand fill. Smaller secondary stopes are filled with either waste rock or paste fill with a low cement content.

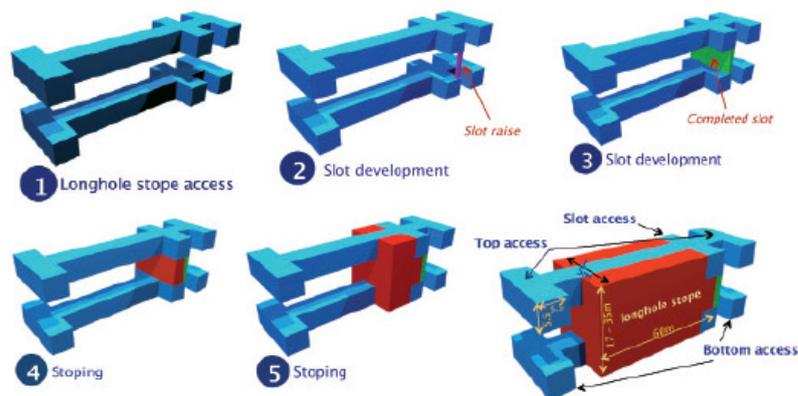


Figure 2.2: Bench-and-fill mining, from [11]

2.2.2. Drift-and-fill

Although the method has relatively low production rates and high unit costs when compared to bench-and-fill, it is usually chosen because it is highly flexible and can achieve high recovery rates in high grade orebodies with complex and flat dipping geometry.

Drift-and-fill stopes are usually accessed through the footwall ramp with footwall access drives driven along the orebody stroke at 20m vertical intervals. Access crosscuts are driven down into the orebody and a horizontal slice is mined using drifts developed either longitudinally or transversely in sequence. Standard drift dimensions are 5-by-5 meters. Following completion of a drift it is tightly backfilled with hydraulic sand fill before the drift alongside is mined. When a complete 5 meters high orebody slice is mined and filled, the back of the access drive is slashed down and mining recommences on the level above.

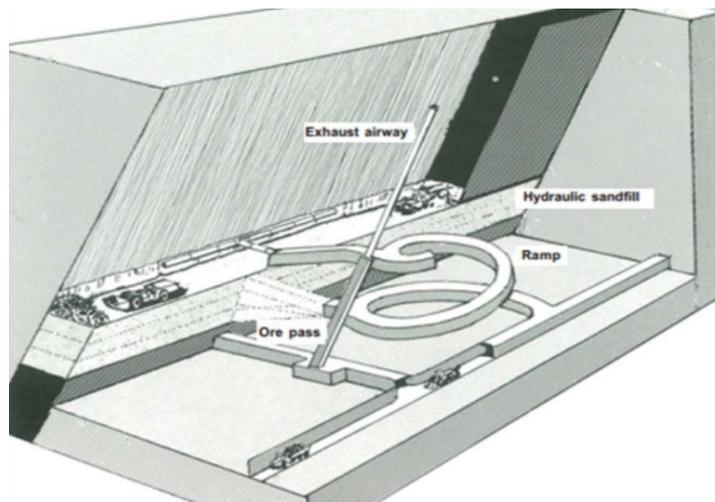


Figure 2.3: Drift-and-fill mining, from [2]

2.3. Problem statement

The stockwork and lens nature of the VMS deposit mean uncertainty plays a role. Where previously stated that most mines fail to meet their target this is especially an issue with VMS deposits. In contrast to for instance coal mining, where deposits are predictable, the combination of high variability and the need for selective mining pose challenges in the mine design phase.

When creating a traditional blockmodel using interpolation techniques such as kriging, the combination of uncertainty and high variability will result in a smoothed resource model. This model cannot be used for detailed stope positioning and target estimations as it would result in a highly inaccurate mine design.

Another option would be to create a number of orebody simulations, each equally probable. However, due to the variability in the VMS deposit, these have a high probability to be an unfit representation of the resource deposit. Choosing one of those simulations is impossible as each of them are equally probable to accurately describe the real situation. Averaging the simulations would lead to a similar issues as when using interpolation methods, as it would lead to mine design being based on an unfit, smoothed, representation of the actual orebody.

Stochastic techniques deal with uncertainty in the way that they help to make a more accurate uncertainty estimation. By using multiple orebody simulations of equal probability, and taking each into account as equally probable orebody representations, the orebody uncertainty can be more accurately dealt with.

Where little uncertainty is present this will not lead to large benefits. However, the high variability within VMS deposits, and the uncertainty this leads to in orebody modelling, mean more significant gains can be expected.

2.4. Optimization considerations

In this research a stochastic mine design optimization will be done on a copper and zinc bearing VMS deposit. The goal will be to find the optimum size and location for stopes to be mined. In the section below the key optimization parameters, the objective functions and constraints that will be considered are listed.

2.4.1. Key optimization parameters

Optimization will be done based in the Economic value of the stopes. The aim is to place the stopes in such a way that they capture the full economic orebody potential by mining as much profitable parts of the deposit as possible. This will be done while taking into consideration a number of constraints.

Furthermore, in selecting the optimum mine design, in some analyses other key drivers will be considered in addition to the economic value:

- The inclusion of penalty elements.
- Average ore grades.

- Ore tonnage.
- Waste tonnage.

The size of pillars and stopes used for the optimization will be closely related to the bench-and-fill mining method, however theoretical values are used. As different VMS deposits are mined using different mining methods due to varying geo-technical characteristics, maximum and minimum stope and pillar sizes differ per operation. To allow different scenarios to be evaluated in the future it was decided not to focus on one particular mining method. Instead stope and pillar sizes were used that allow the deposit variability to play a role.

The economic parameters used for transforming the resource block model into an economic block model were based on real mining operations where copper/zinc VMS deposits are mined using underground stoping.

The aim of the model is not to accommodate one certain operation, but to allow multiple mining operations to be simulated and stochastically analysed. To do so all economic and mining related parameters have remained accessible in the model so that they can be tailored to a desired scenario.

2.4.2. Objective function

The objective function will be to **Maximize** the economic value of the selected stopes.

This will be done based on an economic block model, where we aim to include as many profitable blocks as possible within the constraint limits, whilst leaving out blocks that do not contribute economically to the objective function.

2.4.3. Constraints

Several constraints will be taken into account.

- **Geotechnical constraint:** To ensure the stability of the stopes a pillar boundary should be placed around them. This pillar cannot be mined, even after a stope has potentially been backfilled. The width of this pillar will be fixed during optimization but can be changed within the model.
- **Regularity constraint:** To ensure the mineability of the stopes they will have to be rectangular in shape. A constraint is in place to adjust the individual stopes and ensure a rectangular shape.
- **Stope size constraint:** To ensure geotechnical stability of the stopes and keep their size within the desired range, both a minimum and maximum stope size will be defined. The desired maximum and minimum length and width of the stopes can be adjusted within the model.
- **Ore Grade constraint:** By working with a cut-off grade the inclusion of low grade material (under cut-off grade) will be un-linearly penalized.
- **Ore Quality constraint:** To accommodate the current smelter demands, the inclusion of penalty elements will be constrained. A soft constraint is in place for low grade penalty element inclusion, a hard constraint for above-threshold values.

Theoretical background

3.1. Global Mathematical Optimization

Optimization, or mathematical optimization is “The selection of a best element (with regard to some criterion) from some set of available alternatives” [5].

In the simplest case an optimization problem consists of maximizing or minimizing a function (objective function) by systematically choosing input values from within an allowed set of values, and calculating the outcome value of the function. Its goal is to find the best fitting value which will result in the highest (maximizing) or lowest (minimizing) outcome for the objective function [6].

Generally optimization algorithms can be divided in two groups: deterministic and probabilistic algorithms.

Deterministic algorithms are mostly used if there is a clear relation between the characteristics of the possible solution and its utility for a given problem exists. In that case a search space can efficiently be explored using for instance a divide and conquer scheme [1]. If the relation between an input variable and the outcome of the objective function are not obvious, too complicated, or the search space is too big, it becomes harder to solve a problem deterministically. It would simply take too much computational time.

Probabilistic algorithms are better at dealing with such problems as they do not necessarily try to calculate the actual best fit value, but use the result of a random process instead. One of the main groups of probabilistic algorithms are the Monte Carlo based approaches which trade in guaranteed correctness of the solution for an ‘as good as possible’ solution to decrease run-times.

An important sub-group of probabilistic Monte Carlo based approaches is Evolutionary Computation [1], which groups all algorithms based on a set of multiple solution candidates.

Figure 3.1 shows a global overview of the taxonomy of optimization algorithms.

3.1.1. Metaheuristics

Heuristics are parts of an optimization algorithm that use the information gathered by the algorithm to decide which candidate solution should be tested next, or how the next individual can be composed [1]. Metaheuristics are a higher level heuristic designed to provide not a perfect solution, but a sufficiently good and efficient solution, by treating the problems as a black-box-procedure. Metaheuristics are often performed stochastically by utilizing statistics obtained from search space samples or are based on physical processes or natural phenomenon.

As part of the probabilistic algorithms, metaheuristics do not guarantee a globally optimal solution will be found.

The main advantages of metaheuristics are [7]:

- They move relatively quickly towards very good solutions and thus provide an efficient way of handling large or complicated problems.
- They are useful in cases where other methods get stuck at local optima as they are capable of escaping them.
- They combine properties of different good solutions to generate new solutions.

Figure 3.2 provides an overview of the different classifications of metaheuristics.

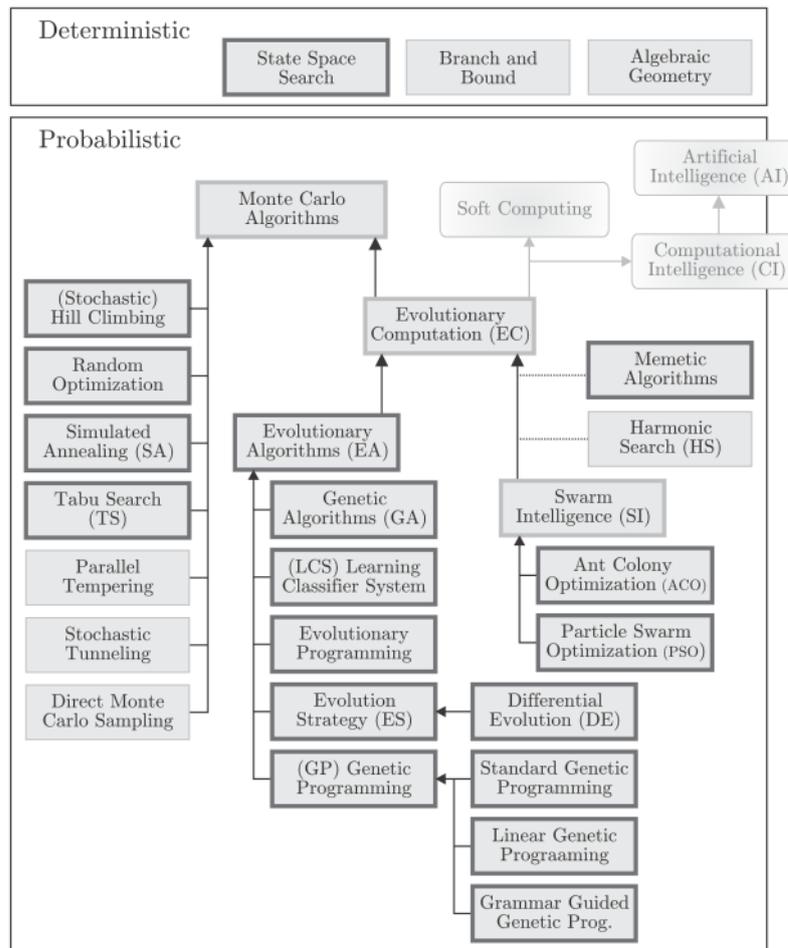


Figure 3.1: Taxonomy of global optimization algorithms, from [55]

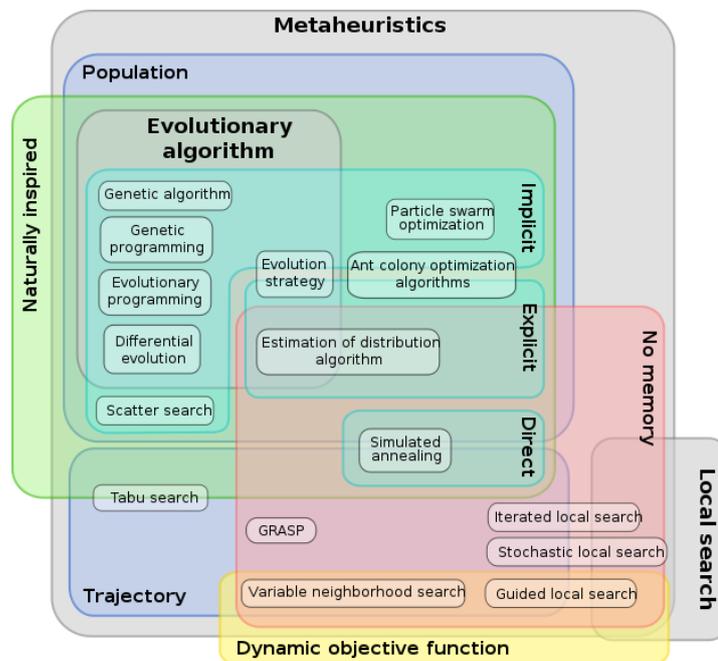


Figure 3.2: Metaheuristics classification, from [8]

3.2. Genetic algorithms

Genetic algorithms are a form of machine learning algorithms. This is a group of algorithms capable of learning - or 'fine-tuning' - while running, without being extensively programmed. These algorithms make decisions based on their input data. Despite being one of the most impractical forms of machine learning it is one of the best general-purpose learning algorithms. Even though for most problems there are more efficient solving algorithms, genetic algorithms do a very good job at solving cases with a large number of non-correlating parameters [source].

Being a form of meta-heuristics Genetic algorithms do not guarantee an optimal solution, as would an exact method, however they do a great job in approaching the optimal solution relatively quickly. As the time to find the exact solution, if one exists at all, is several orders of magnitudes bigger than the heuristic one, heuristic algorithms tend to be used for complex problems or problems with large datasets. Due to the latter stochastic problems are usually approached in a heuristic manner [31].

Genetic algorithms have proven to be both a successful and powerful problem solving algorithm demonstrating the potential powers of evolutionary concepts. They are applied in a large range of fields solving difficult problems with large sets of non-correlating parameters. The solutions they produce are usually more complex and efficient than a human engineer could come up with by hand [31].

3.2.1. Why genetic algorithms?

For this research it was decided to use genetic algorithms as a mean to optimize a stope layout problem for underground mining of VMS deposits. In addition the above mentioned benefits of genetic algorithms, the main decision drivers have been:

- Genetic algorithms are intrinsically parallel. Where most other algorithms can only explore the solution space to a problem in one direction at a time, genetic algorithms can work in multiple directions at once. This means that where other algorithms get stuck and have to start over, Genetic algorithms can easily eliminate the wrong search path and continue in a more promising direction.
- Genetic algorithms work well for problems where the fitness landscape is complex (discontinuous fitness function, changes over time, many local optima) or very large
- Genetic algorithms need no information on the problems they need to solve. Instead of requiring domain specific information they make random changes to the candidate solutions and use a fitness function to determine whether those changes produce an improvement.

While the model presented in chapter 4 does not use a complex input in the optimization analyses done for this research, it was designed to handle larger data sets and to be extended in the future. Where orebody models are complex domains, using genetic algorithms should be able to decrease run-times and reduce the probability of it getting stuck on a local optimum.

Furthermore, as they need limited information the algorithm can be easily adjusted and extended.

3.2.2. History

Since the 1950's scientists have studied artificial intelligence and evolutionary computation, trying to write programs able to simulate natural processes. In 1953 Dr. Barricelli [23] started studying artificial intelligence at Princeton, using one of the first computers to try and mimic natural reproduction and mutation. His research was not aimed at solving optimization problems or recreating biological evolution, but tried to create artificial intelligence. In 1954 his research resulted in the first computer program related to genetic algorithms.

His work was followed by dr. Fraser, an English biologist. He was the first to attempt creating a computer model for biological evolution, as this could not be observed in nature [39] [38].

The first genetic algorithm was invented by John Holland in the 1960's [41] and further developed by him and his students and colleagues at the university of Michigan during the 1960's and 70's. Holland's goal was to study the phenomenon of evolution and adaptation as occurs in nature and to explore how these concepts could be integrated in computer systems.

In the book *Adaptation in Natural and Artificial Systems*, released by Holland in 1975 [42], he presented

the algorithm as an abstract form of biological evolution. Also he gave a framework for adaptation of the Genetic Algorithm. His method for moving from one population to a new population incorporates all the basic techniques; cross-over, mutation and inversion. Previous works had only used mutation as evolutionary driver. In his model individuals are represented by binary strings (e.g. 1's and 0's) where the more successful individuals are allowed to produce more offspring than lesser ones.

Holland's attempt was the first to successfully incorporate evolution in computational science and until in the 1990's his work was the basis of almost all theoretical work on genetic algorithms. Furthermore his book remains popular as it demonstrates the mathematics behind evolution.

Research on genetic algorithms increased in the 1970's and 1980's driven by advancing computer technology. Furthermore researchers also became increasingly aware of the shortcomings of traditional optimization techniques for solving complex problems. It was found that genetic algorithms could tackle problems that traditional methods so far had been unable to solve.

3.2.3. Overview of general genetic algorithm

In basic terms a Genetic Algorithm is a programming method using biological evolution as a strategy to solve a certain problem. Given the problem to solve, the input of the model is a number of encoded potential solutions which are each quantitatively evaluated by a certain fitness function. The input, or candidates, might be solutions already known to work in order to optimize them or select the best one. Usually however these starter candidates have been generated at random.

After the generation of the first generation, the algorithm then evaluates and ranks each candidate according to the fitness function. After ranking most candidates will turn out to be no good solutions and are subsequently ignored or deleted. A few however might show potential or improvement. These will be allowed to reproduce, usually called cross-over.

Multiple copies or 'offspring' are made of the first set of candidate solutions. These copies however are not the same as the first parent generation as alterations and random changes are made. Apart from cross-over this usually also includes some form of mutation. These offspring solutions go on to form the second generation of candidate solutions and are subsequently evaluated by the fitness function. Again the offspring that has worsened is deleted. However, some candidates of the offspring generation might show improvement that has made them better or more efficient solutions.

This process is repeated over a certain number of generation steps under the assumption that each new generation, through a new round of mating and mutations will encounter some better solution candidates than present in any of the previous generations [31].

3.2.4. Solution representation (candidates)

In order for a Genetic Algorithm to be able to work on solving any problem, the potential solutions need to be encoded in a form that can be processed by the program. Almost all algorithms work with a data-string in which each position resembles a certain variable. The problem itself defines the size of the string, which, in a problem with few variables can be short. Usually however strings can be several thousand values long.

The most common encoding scheme is binary coding. Here each possible candidate is represented by a string of 1's and 0's. and each string position refers to a variable or characteristic that can be either turned 'on' or 'off'. The string itself is one-dimensional but can easily be transposed to two or more dimensions.

Similar approaches are those where the string is not binary but contains integers or decimal numbers. This would not change how the algorithm work massively but allows more complex models and greater precision. Furthermore it can be seen as more intuitive programming as it is easier to relate to string positions.

A third method is grammatical encoding. Here each individual is represented by a string of letters, where each can, again, stand for an aspect of the solution.

A benefit of the above listed methods is that they make it easy to quickly encode random candidates. Also they allow for a quick evaluation of the effects of mutation and cross-over. Furthermore, defining the operators referring to the final solution is made relatively easy.

A fourth method has been developed by Koza at the Stanford University [45]. In his algorithm the candidates are not represented by a string of data but as branching data structures. These structures – or trees – are a set of branched nodes in which each node contains a certain value. Apart from the genetic algorithm being able to change the node values, also branches can be swapped or altered.

3.2.5. Selection and measure of optimality

Selection is the process of choosing two parent chromosomes from the population to create offspring with. Its main purpose is to emphasize fitter individuals in the hope that their offspring has a higher fitness. The main idea is that by taking two good chromosomes with different traits, the resulting offspring has the positives of both parents and thus has a higher solution value. By repeating this over numerous generations eventually this will lead to the optimal solution.

The selection pressure is defined as the degree to which the better individuals are favoured [52]. The higher the selection pressure, the greater the chance that the individual is picked for creating offspring. The magnitude of the selection pressure also greatly influences the convergence rate of the genetic algorithm. If the selection pressure is too low, the convergence rate will be slow. However, a too high selection pressure can result in the model getting stuck on a local optimum and thus will not result in finding the optimal solution.

Two selection schemes can typically be distinguished; proportionate based selection and ordinal based selection.

The first picks out the individuals based on their fitness value relative to the fitness of other individuals in the population. An example is roulette wheel selection, one of the most traditional selection methods. Ordinal based selection picks individuals, not based on their fitness value, but upon their rank within the population.

3.2.5.1. Random selection

In random selection the parent chromosomes are chosen not based on any ranking or fitness but are just taken out of the population at random.

3.2.5.2. Fitness proportionate selection

Fitness proportionate selection, also known as roulette wheel selection is the most common operator for selecting potentially useful solutions for reproduction. As in all selection methods it assigns a fitness value to all individuals, which is then used to assign to it a probability of being chosen for reproduction. The selection probability is calculated by using equation 3.1.

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad (3.1)$$

Where:

- p_i = probability
- f_i = the fitness of individual i
- N = the number of individuals in the population

This concept can be envisioned similar to a roulette wheel in a casino, where each candidate solution represents a pocket on the wheel with a size relative to its fitness. The wheel is played until the fitness proportionate values add up to the maximum combinatory value for that generation.

Even though it favours the stronger solutions this way, it does not eliminate the possibility of a bad solution being chosen. This prevents favourable genetic properties disappearing due to being in an overall worse solution string.

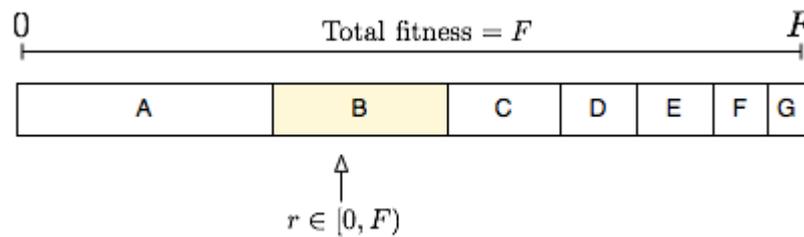


Figure 3.3: Roulette wheel, from [15]

3.2.5.3. Other selection methods

Roulette wheel selection is not always the preferred selection mechanism. When the fitness of the chromosome string varies strongly, and where there is a big difference in fitness between the best and worst solution, roulette wheel fails to perform well. Furthermore other selection methods might be quicker or easier to implement [27].

Popular alternatives are tournament selection and stochastic universal sampling.

In tournament selection repeatedly two strings are randomly chosen for a ‘tournament’ where the better solution is selected for reproduction. The amount of pairs selected depends on the tournament size, which is equal to the amount of chromosomes demanded for crossover.

Stochastic universal sampling is more similar to roulette wheel selection, but uses intervals in selecting chromosomes, thus increasing the chance of weaker solutions being selected.

3.2.6. Recombination (Cross-over)

Recombination is the process of generating offspring from a set of parent solutions. It follows after the parent solutions have been valuated and ranked based on their fitness. The idea is that it takes two chromosome strings and implements a cross-over operator to create offspring with genetic properties of both parents. The aim is to combine two good solutions so that the offspring has the beneficial properties of both parents. Usually two parent chromosomes result in two offspring chromosomes. However, variations in which two parent strings create only one chromosome have also been used.

The two most common methods for cross-over are listed below.

3.2.6.1. Single point cross-over

The most common method in which recombination is applied is single point cross-over. For this method two parent chromosomes are selected and a random or semi-random cut-point is chosen. At this point both parent strings are cut in half. These halves are then recombined to form 2 new chromosomes both consisting of genetic material from the two parent strings.

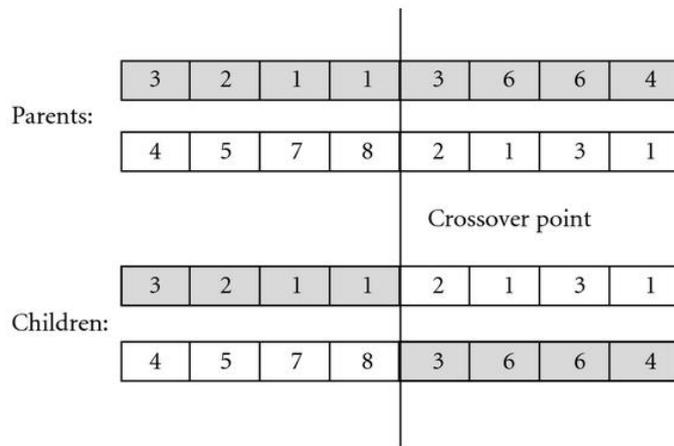


Figure 3.4: Single point cross-over, from [16]

3.2.6.2. Dual point cross-over

Dual point cross-over works similar to the above mentioned single point cross-over. However, here each parent string is not cut in two, but three pieces which are then recombined to form the offspring chromosomes.

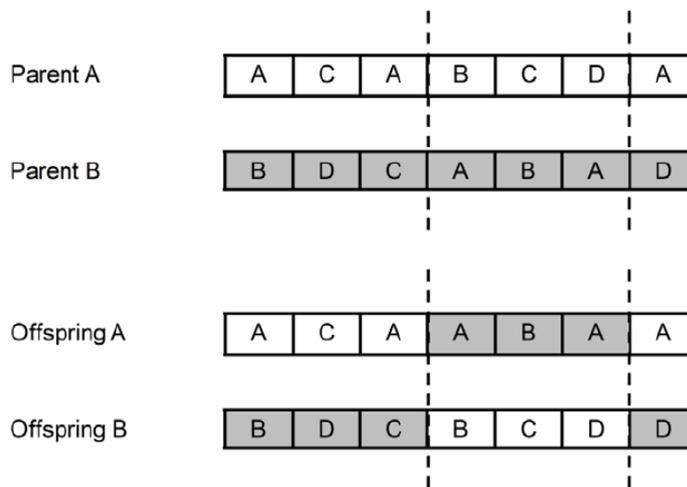


Figure 3.5: Two point cross-over, from [18]

The idea behind dual-point or multi-point (more than 2 crossover points) is that the part of the genetic chromosome contributing most to the objective function may not necessarily be contained in the substring selected for cross-over [28]. Increasing the number of cuts, and therefore the number of genomes to be swapped, should increase this probability. Furthermore, multi-point cross-over results in more emphasis being put on the search space. This prevents the optimizer converging early, but decreases the chances of getting stuck in a local optimum [53].

3.2.7. Mutation

Mutation is generally performed after cross-over. It introduces a random change in the genetic string in order to prevent the genetic algorithm from getting stuck at a local optimum solution. A local optimum can be seen as a solution that is optimal within a certain part of the domain or dataset, but is not the best possible solution over all possible solutions. This is illustrated by figure 3.6. Furthermore it is a way of recovering lost genetic material and introducing new solutions. Hence it covers for good genetic characteristics getting lost due to being in an overall bad genetic string.

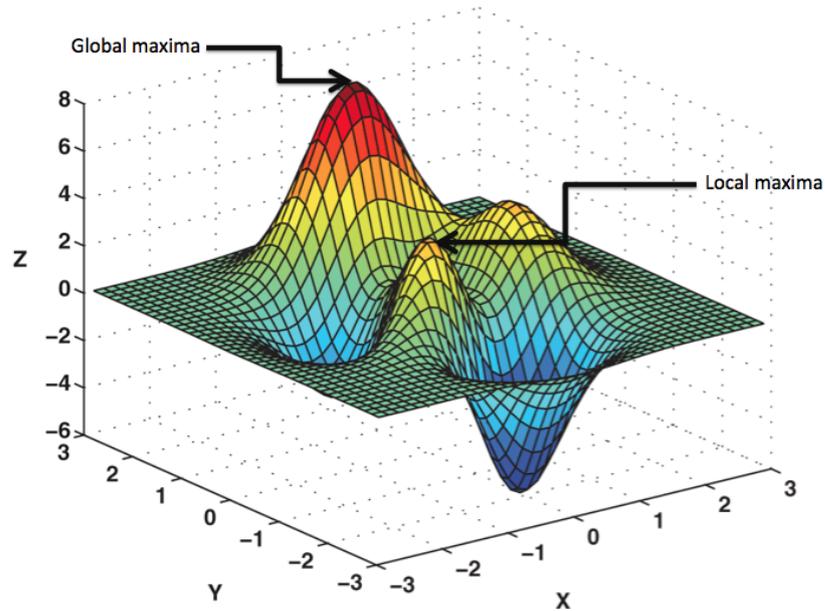


Figure 3.6: Global and local maximum, from [13]

Mutation is supposed to cover the full genetic string so that it can influence all model parameters and should have access to all genetic strings in a population. The probability of a mutation happening is usually taken as $1/L$, where L is the length of a chromosome (genetic string). However this is not a given as different applications require different mutation probabilities to converge efficiently to the global optimum. However, care should be taken that the mutation probability is not too high, as the algorithm would then change to random search.

Another option could be to implement a hill-climbing mutation operator that only allows mutation if it improves the solution. It is possible to accelerate the optimizer this way, however, it might also reduce the population diversity. This could then lead to the optimizer getting stuck at a local optimum.

Below several of the most common mutation operators are described.

3.2.7.1. Flip-Bit Mutation

Goes for binary strings. One or more genes are selected and reverted. If the specific gene houses a 1 it is changed in a 0 and vice versa.

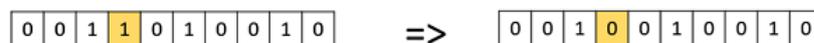


Figure 3.7: Flip-Bit Mutation, from [14]

3.2.7.2. Swap Mutation

In swap mutation two gene locations are selected and swapped. In the case of binary chromosomes a change is not guaranteed to happen if both genes have the same value. On non-binary coded algorithms this method can have bigger effect.

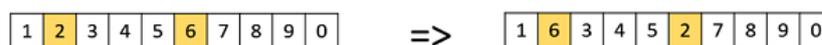


Figure 3.8: Swap Mutation, from [14]

3.2.7.3. Scramble Mutation

In scramble mutation a number of adjacent genes are selected and shuffled randomly. The number of genes to select depends on the algorithm and chromosome length.

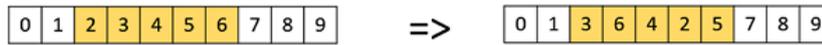


Figure 3.9: Scramble Mutation, from [14]

3.2.7.4. Inversion Mutation

Here, like in scramble mutation, a number of adjacent genes are selected. However instead of randomly shuffling them, their order is simply reversed.

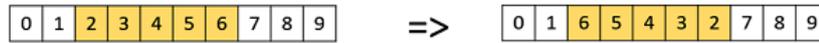


Figure 3.10: Inversion Mutation, from [14]

3.2.7.5. Displacement Mutation

Displacement mutation re-locates a number of adjacent genes within a chromosome and shifts the genes that were originally on this location in the direction of the mutation location.

3.3. Conditional resource modelling and simulation

The underground geologic conditions on which a mine design is based are commonly defined as the ‘state of nature’. This state is poorly known as relatively little information can be provided by drillhole data. Getting more and more accurate data through additional drilling can reduce the geologic uncertainty but is expensive. The systematic and random errors accumulated during sample collection, preparation and assaying decrease accuracy and increase developmental risk. The information is obtained at specific data points and must be interpreted to develop the three-dimensional geologic model necessary for mine planning and thus adds another level of uncertainty. The quality of the model is depended, not only on the drilling data, but also on the method used for interpolation and the expertise of the developers. The model is never better than the information used for its development.

The process of modelling is complicated as it is required for estimating how the mining method will influence the tonnage and grade of ore sent to the processing plant. It should also be applicable both to the evaluation of selective mining, as well as bulk mining. Underground mining requires a differently modelled representation than open pit operations and the geostatistical methods used must accommodate to that.

Deposit modelling required the estimation of discrete features such as geologic zones and continuous variables such as grades, penalty elements and metallurgical properties. Knowledge of uncertainty in the location of these discrete features can be quantified by analysing different possible interpolations of the same data. Uncertainty in discrete features can be quantified through the use of geostatistical simulation techniques.

3.3.1. Geostatistical Interpolation and kriging

Originally, in spatial statistics, geostatistics was synonymous with kriging, a statistical version of interpolation. In the last decades this definition has widened through the development of other interpolation techniques. In mining however kriging related interpolation methods are still preferred and used in almost every operation.

The basic idea of kriging is to predict the value of a function at a given point (e.g. ore grade) by computing the weighted average of the known neighbourhood point values. The method is mathematically closely related to regression analysis as both theories derive a ‘Best Linear Unbiased Estimator’, most commonly referred to as ‘BLUE’. The BLUE is based on the assumptions of covariances. Furthermore kriging makes use of the ‘Gauss-Markov theorem’ to prove independence of the estimate and error [33] [4].

$$z_0^* = \sum_{i=1}^n \lambda_i z_i \quad (3.2)$$

Where:

- z_0^* = the value at an unsampled location to be estimated from a linear combination of n values of a regionalized variable z_i
- λ_i = the weight of the regionalized variable z_i at a given location

- z_i = the regionalized variable at a given location

Several forms of kriging have been developed as depending on the properties of the random field (stochastic) and the various degrees of stationarity assumed, different methods for calculating neighbourhood points can be used. The most common kriging methods used in mining related geostatistics are:

- Ordinary kriging, which assumes a constant unknown mean over the search neighbourhood
- Simple kriging, which assumes stationarity of the first moment over the entire domain with a known mean.
- Universal kriging, which assumes a general polynomial trend model (e.g. linear trend model)
- Co-kriging, which uses information of several variable types.

A known downside of kriging is that it smooths out the original data as it overestimates the low point values and underestimates the highs. This characteristic results in a smoothed orebody model. This is one of the main reasons of numerous cases where the mine was not able to meet its targets by not being optimally designed to the actual orebody.

Because of this, in the last decades alternatives have been studied.

3.3.2. Conditional Simulation of orebody models

Deterministic methods like kriging, whose output is a single unique solution, do not attribute to the actual variability. The smoothing characteristic of any interpolation algorithm replaces local detail with a good average value. However, the mining industry, and specifically the mining engineer are more interested in finer-scaled details as these are needed to provide an optimal mine design.

Unlike deterministic methods, simulation methods preserve the variance observed in the data. Their stochastic approach allow for calculation of many equally probable solutions, which can be used to quantify and assess uncertainty.

Like the traditional deterministic approach, stochastic methods like conditional simulation preserve hard data where known, and use soft data where informative. The kriged solution in itself is the average of numerous realizations, and the variability in the different outcomes is a measure of uncertainty at any location. The standard deviation of all simulated values generated for each grid point is the quantification of uncertainty.

3.3.3. Stochastic modelling

In general, these stochastic simulation techniques (i.e. conditional simulation) require that the main input parameters (spatial models such as variograms) and the sample value distribution (i.e. cumulative distribution function) remain constant over all simulations within a given geologic interval. The structural and stratigraphic model (resource outline) remains fixed.

As each realization begins with a different random seed number, each are calculated following a unique path, guaranteeing the uniqueness of the outcome. The random path provides the simulation algorithm with the specific order of cells to be simulated in. Therefore, the results are different at all un-sampled locations, producing local changes in the property distribution. This also means that selecting the same random seed will always reproduce the same path, allowing for reproduction.

3.3.4. Methods of stochastic simulation

A large number of stochastic simulation approaches have been developed over the last decades. Most of which are also often referred to as Monte-Carlo simulation techniques. Most notable of these approaches are turning bands simulation, one of the earliest simulation methods and sequential simulations.

3.3.5. Turning bands simulation

In turning bands simulation, first the data are kriged after which unconditional simulations are created using a set of randomly distributed bands (or lines). The general procedure as described by Mantoglou and Wilson [47] is as follows:

1. Raw data values are kriged to a regular grid.

2. Numerous random lines (bands) with various azimuths are generated around a centroid located at the grid or volume center. The modeler controls the number of lines.
3. Unconditional simulations of normal-score transformed data are performed along each line using the transformed-data histogram and variogram.
4. Values along the lines are linearly interpolated to grid nodes—the more lines, the less interpolation.
5. Unconditional interpolated values are back interpolated to well locations.
6. Unconditional interpolated values and well locations are kriged.
7. The grid of unconditional interpolated values is subtracted from the grid of unconditional results in step 5. This creates a residual map with a value of zero at the well locations.
8. The residuals are back interpolated from the y-space to the z-space.
9. The back-interpolated residuals from step 8 are added to the original kriged map from step 1.
10. The result is a grid or volume of data values that reproduce both the mean of the raw data and the variance.

3.3.6. Sequential simulation

Sequential simulation is a collective name for three different techniques using the same basic algorithm. These techniques are:

- Sequential Gaussian Simulation (SGS), used to simulate continuous variables such as ore grades. Is usually applied when geological and mineralogical properties are being modelled.
- Sequential Indicator Simulation (SIS), which simulates discrete (binary) variables using the same methodology as SGS.
- Bayesian Indicator Simulation (BIS), a descendant from SIS, uses a combination of classification and binary indicator methods.

The general procedure according to Deutsch and Journel [34] is as follows:

1. Perform a normal-score transformation of the raw data.
2. Randomly select a node that is not yet simulated in the grid.
3. Estimate the local conditional probability distribution function (lcpd) for the residuals at the selected node. The residuals can be calculated by subtracting the grid of an unconditional simulation from a kriged grid of the unconditional values sampled at the geographic coordinates of the wells.
4. Create a newly simulated value by adding together the randomly drawn residual value and the mean of the transformed data.
5. Include the newly simulated value in the set of conditioning data, within a specified radius of the new target location. This ensures that closely spaced values have the correct short-scale correlation.
6. Repeat until all grid nodes have a simulated value.

3.3.7. Stochastic simulations for risk analyses

As using stochastic simulation techniques to generate orebody models are a way to capture the grade uncertainty resulting from sparse exploratory drilling, these stochastic simulations can be used to analyse and quantify the present uncertainty. By using a large number of stochastic simulations, and by studying the effect each of these realizations would have on the project case risks can be made visible. Now, instead of creating one model and using that as representation of reality, the range of possible realistic outcomes is unveiled. This range of possible solutions can be used for several risk analyses. Risk can then be decreased by finding a solution (mine design) that is robust towards a wide range of possible outcomes.

4

Model

4.1. Optimizer model

For this research a genetic algorithm has been coded to stochastically optimize stope locations within one of the ore bodies of the Neves-Corvo mine in Portugal. As no similar model exists, the model was built up from scratch based on genetic algorithm fundamentals. An overview of the model and an explanation on the working of its individual components can be found in later sections. The scope of the model, prior to programming, and the key assumptions made during this stage are described in the following chapters.

4.1.1. Model characteristics

The main scope of the stochastic mine-design algorithm is to optimize stope geometries within the Neves resource deposit as part of the Neves-Corvo mine. A genetic algorithm forms the basis of the stope location optimization, which is value driven based on economic value. Even though the model was built based on the Neves-Corvo dataset, the aim has been to keep it dataset independent. For this, several import possibilities were taken into account and an import function for Excel, ASCII- and CSV files is present.

In order to perform value optimization, a metal grade model was converted to an Economic block model. Different independent models (one for each element grade) have been combined and the main penalty element (Arsenic) has been taken into account.

Geotechnical characteristics were accounted for by taking into account pillar barriers. These un-minable blocks are independent of the stope sizes and are to prevent stopes from becoming oversized.

The model is able to work with an unlimited amount of simulations to allow stochastic optimizing. However, it is also able to work with a single model. As the model was used to compare E-type and stochastic optimization different realizations are compared while running.

4.1.2. Optimization assumptions

Several assumptions were made during this research. By far most of these were made while coding the algorithm. As the aim of the research was not to create an industry ready product, but instead explore an optimization concept this is justified. Also making assumptions was sometimes necessary to keep model runtimes within reasonable ranges or to compensate for missing information. Some of the most critical assumptions are listed:

- The model is limited to 2 Dimensions only as adding a third dimension would greatly increase the required run-time for each problem. However, the model was designed so that adding a third dimension in a later stage would be possible.
- The stope geometry optimization is done on the basis of economic block values. These are calculated taking into account the three main elements: copper, zinc and arsenic. Other elements such as silver are not taken into consideration due to the absence of data or the fact that they do not contribute enough to the economic block values.
- For the optimization process the amount of stopes are not taken into consideration. There is no option available for limiting or capping the amount of stopes for the final design.
- Blending is not considered for the value model. In reality ore from the different resource deposits is blended during upgrading and processing. As only data from one ore body has been made available it was not possible to take this into consideration.

- As blending was not considered and few detailed information on processing and concentration could be found, a number of parameters used for calculating the economic block model were averaged or estimated.
- Flexible pillar sizes have been made possible, however, these were not coded as a function of the individual stope sizes. Instead they act as a firm barrier between the individual stopes. This does not have an effect on the stope placement possibilities.

4.1.3. Coding language

For the programming done during this research Matlab will be used. Matlab is a numerical programming environment and fourth-generation programming language, being developed by Mathworks since 1984. It initially started off as a wrapper on Fortran libraries for linear algebra but has since evolved in a coding language on its own. Some of its key features are:

- Good language for scientific and engineering computing.
- Clear desktop environment.
- Well integrated graphics and add-ons for visualizing data.
- Numerous tool boxes for a wide variety of engineering and scientific applications.
- Good compatibility to C/C++, Java, .NET, Python, SQL and Microsoft Excel.

As a language it has grown organically over the years, and there are some flaws baked in if you look at it just as a programming language. However, if you look at it as an environment for doing research in it has a number of strengths.

The notation is simple and the implementation relatively fast. It is good in generating plots and there are a large number of toolboxes for particular tasks. Furthermore there is a large community of users that share numerical codes apart from the company offering a large database of documentation. Hence, despite not being one of the more advanced coding languages, it remains very popular at educational institutions, engineering departments and industries.

Some of the specific reasons to use Matlab during this research are that, as it has a lot of built-in functions it allows for quick coding and testing and has good compatibility with most standard data processing software. This makes it easy to write a flexible code with various import possibilities and makes it easy to distribute.

4.2. Description of the resource model

As part of the H2020 Real Time Mining WP& program Delft University of Technology has requested Geovariances, A geostatistical software developer and consultancy bureau, to perform a consultation aimed at generating a geostatistical resource estimate of the Arsenic, Copper and Zinc mineralized orebodies of the NevesCorvo deposit based on drill hole and chip sampling information.

The Neves and Corvo orebodies had previously been built for a single domain by Geovia. Geovariances used this solid as the estimation zone.

Following that, Geovariances have tested several methods regarding data integration. These included kriging, co-kriging and kriging with a variance of the measure of error. This was done with the aim of identifying the most adapted method for producing the target Ground Truth Model (GTM).

In addition to this GTM estimate, a platform of conditional simulations has been produced allowing the selection of separate realizations, each fully characterizing the spatial variability of the key elements (copper, zinc and arsenic) [25].

4.2.1. Method of orebody modelling

The simulation realizations were performed independently for the available arsenic, copper and zinc variable. The simulations are turning-bands block simulations.

With the turning bands method, conditional simulations are obtained using a two-step procedure:

1. Non conditional simulation of the specified variogram model. The number of turning bands is the only parameter involved at this level and should be large enough to ensure good quality of the resulting simulations
2. Conditioning the data with Co-Kriging.

The simulations were created by taking the following steps as described by Geovariances:

1. The main data set from Geovia was imported in Isatis (the main Geostatistical software) and the exploration drill hole samples and chip samples were combined.
2. The raw drill hole and chip sample data was transformed into their Gaussian equivalent through Gaussian Anamorphosis.
3. Sample co-kriging was done on the Gaussian drill hole and chip sample data.
4. Panel kriging was then done on the Gaussian data at the data location of the main variables. The variance used here was the measurement error.
5. This information was then transformed into a block model.
6. From this blockmodel then 100 realizations were created using turning bands simulations. The variances used were calculated from the measurement error derived from the Co-Kriging and the local mean.

After creating the conditional simulation realizations, a statistical validation process was performed. For this basic statistics of the main statistical parameters, plots and histograms were calculated. This analysis did not expose any issues, such as erroneous values, with the realizations produced.

4.2.2. Delivered results

For both the Neves and the Corvoorebodies 100 conditional simulation realizations were produced for each of the three key elements, resulting in a total of 600 conditional simulations.

Concerning the quality of the conditional simulations Geovariances has however stated that:

- The distributions of the arsenic, copper and zinc grade simulations have turned out smoother than the raw variable distributions. However, the distributions are close to the raw variable, with more dispersion for the Corvo orebody when it comes to the arsenic and zinc grades.
- For the Neves data, the simulated global mean and the mean of each realization are close to the mean of the exploration data.

In this research only the simulations of the Neves orebody will be used.

The following table presents the mean of the global statistics of the arsenic, copper and zinc conditional simulations within the Neves dataset.

Table 4.1: Neves global statistics of As (ppm), Cu and Zn by TB block simulations

	Variable	Count	Minimum	Maximum	Mean	Std. Dev.	Variance
Neves	As	7206000	333	50503	7291	5047	25468023
	Cu	7527300	0.354	16.37	2.47	1.53	2.34
	Zn	7501200	0.0400	8.31	0.622	0.628	0.394

4.3. Model Schematic

The mine design optimization is done by a genetic algorithm scripted in Matlab. Figure 4.1 shows the build-up of the optimizer. In great lines it follows the structure of a standard genetic algorithm. However, due to the 2D nature of the optimizer and the possibility of changing optimization settings it incorporates some extra features. The next section contains a brief overview of the complete model. In the following chapters the independent modules will be described.

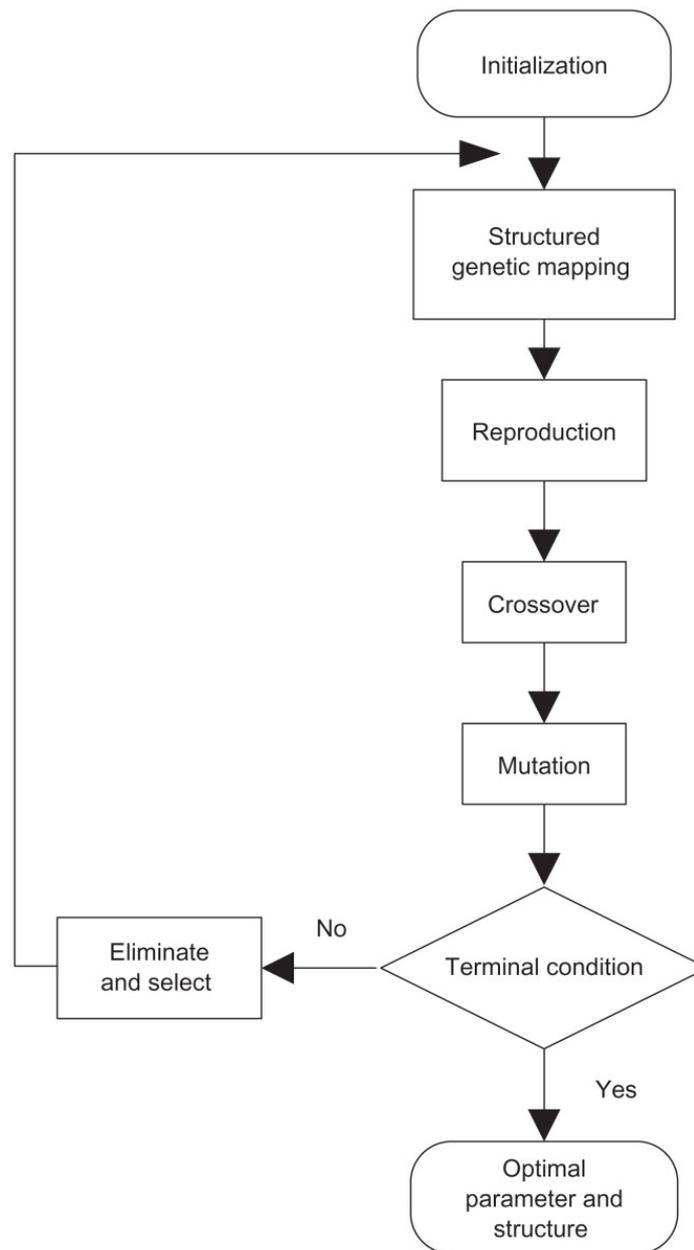


Figure 4.1: Schematic flowchart of real structured genetic algorithm

The model starts by importing the dataset. Apart from being able to import the main orebody models provided by Geovariances as .csv files it also has the possibility to load in ASCII or Excel files of varying sizes.

After the dataset is imported all model parameters are set. This is done after importing the data as several variables are data size dependent or use dimensional data in their calculation.

Next the first generation population is created. These parent chromosomes are created by filling a matrix of the same dimensions as the block model layer with randomly placed stopes. The stopes are defined in a binary one-dimensional string and are compliant with the defined minimum and maximum allowable stope size. Furthermore they are placed so that they do not directly touch or overlap.

The solution value of the first generation chromosomes is calculated next and the strings are ranked. The value is calculated by multiplying each first generation string with an economic representations of the two dimensional block models.

Based on their solution value the proportional fitness values for later roulette wheel selection are assigned. This step is the first within the multi-generational loop and is later repeated for each consecutive generation. After the solutions are ranked the best solutions are then taken out for later eliting.

The value calculation is followed by the standard steps in genetic algorithms: cross-over (Fitness proportionate or random) and mutation (flipbit or random). As these processes will have mixed up the binary strings they are not compliant to the constraints anymore. Hence a series of constraint operators are used to ensure that all strings represent a viable stope distribution. These strings represent the provisional next generation of chromosomes. The new generation is finalized after by applying the last step in the eliting process.

After each new generation is created termination is considered. The model is terminated either when no improvements have been measured over a certain number of generations, or if the maximum amount of previously defined iterations is reached. As long as one of these two termination criteria are not fulfilled the evolution process is continued.

After termination the last generation of binary strings, including the best optimized stope design, are saved along with their economic values and other run data. These can then be used for visualization and analysis.

4.4. Model data Input

The optimizer model is capable of importing data of different sources, being: Excel, ASCII and CSV files. Making the model suited for treating different inputs was done to make it independent from the NevesCorvo data sets and allows other similar models from other cases to be optimized. Furthermore, this also allowed quick validation of the separate optimizer modules as a smaller dataset is quicker to run.

In the upcoming sections each of the three import methods is described. The descriptions will be accompanied by a flow chart for each method.

4.4.1. Excel Import

The excel import, contrary to the other two methods later described, is not capable of loading in multiple simulation scenarios. The method however allows for importing relatively small datasets which are quicker to run. Because of this the input feature was heavily used during the model validation and the testing of separate optimizer modules. Figure C.2 shows a flowchart of the Excel file import process.

Using a built-in Matlab function the specified Excel file is loaded and its data imported as two dimensional matrix. If the data inside the excel function is irregular, that would mean not perfectly rectangular the missing data will translate in NaN filled cells in a rectangular matlab matrix. As empty cells would not allow further calculations these are replaced by a 'penalty-value'. This is to indicate they do not contribute to the optimization process. Filling the empty cells with zeros would mean they do not penalize the optimizer for selecting them and would therefore influence the stope geometries in an unrealistic manner.

The magnitude of this penalty value is irrelevant for later value determination as they would simply be characterized as waste and assigned a standard mining cost. However if used purely for validation without later value recalculation, negative penalty values of greater magnitude will aid the optimization process.

As matlab automatically determines the matrix size it is not necessary to specify them beforehand. The dimensional data is read out by an autonomous function and used for calculating later parameters and are directly responsible in determining the binary chromosome lengths.

4.4.2. ASCII Import

Alongside the Excel import module, a module was scripted to allow ASCII files to be imported. Figure C.3 shows an overview flow-chart of this module. Where the Excel function only allows one layer dataset to be used in each run, using ASCII files allowed for importing a set of conditional simulations. These were used for validating the stochastic model properties. For this an ASCII file containing a real-case scenario accompanied by 200 conditional simulations of the same scenario was received.

The data contained by the ASCII-file was organized as a number of columns containing cell values, each column representing a different scenario. After importing the data in the same format the number of sim-

ulations and dimensions are specified. Using these parameters each simulation is transposed to a 2D value matrix by assigning each individual value to a cell. The values are placed in the value matrix by using a double loop to fill the empty rows from right to left and bottom to top.

The first column of ASCII data, representing the 'known' case, is saved as a separate two dimensional matrix file for output. The simulations are stored in a separate three dimensional matrix where the first two dimensions are concerning the individual simulated layer values and the third dimension stores all individual simulations.

For later validation the 3D matrix containing the simulations is used to calculate a separate E-type layer. This two dimensional layer matrix contains for each coordinate cell the average value of all simulated values on that specific coordinate.

As the simulation values used for optimization were used to test the working of the optimizer on larger scale, these were used as-is. No cut-off grade was considered.

4.4.3. CSV Import

The Neves-Corvo data for the main analysis was prepared by Geovariances consulting in Isatis and delivered to the TU Delft as a number of .csv files. This included the copper, zinc and arsenic grade data needed for the stope geometry optimization. In order to use these datasets within the stochastic optimizer environment a .csv import module was scripted in Matlab. A flowchart for the import module is pictured in figure C.4. The following section contains a description of the module.

As the .csv files received for this research have been pre-organized they can be imported into matlab as a standard table. This table is then directly converted to an array to allow better data manipulation. The first four columns of data contain the sample number and X, Y and Z coordinates respectively. This is followed by a column of data contained through kriging with variance of measurement error which is not further considered in this research. These first meta-data columns are followed by the simulated grade data.

Next the layer data is isolated by filtering the z-coordinates on a certain depth. For each simulation then the grade data and x and y coordinates are stored in an empty 3D matrix. Based on the maximum and minimum coordinates recorded the model size is determined and exported. Each simulation is then, in turn, read out of the 3D matrix and plotted on a 2D mesh using the relative coordinates (lowest x or y coordinate corresponds to [0,0]). The meshes, containing grade data, are again bundled into a new 3D matrix containing all simulations. An E-type model is calculated by averaging the grade data of all meshed simulations over the z-axis and the first simulation is isolated and saved independently to feature as a 'real case' benchmark.

4.5. Economic block model creation

The resource model provided by Geovariances consists of separate models for the evaluated elements. In order to implement the information, the grade data on copper, zinc and arsenic are used to create a single representative model to be used for optimization. To allow value based optimization an economic block model had to be created as this would allow valuating the individual solutions. This is important as, in order to progress towards an optimum solution, fit individuals need to be selected for mating while less fit solutions are replaced. The module flowchart can be found in Appendix B.

Economic block models in mining are usually calculated by using a number of straight forward calculations. For this research it was decided to use a schematic comparable to the one proposed by W.A. Hustrulid and M. Kuchta in 'Open Pit Mine Planning and Design, Second edition; chapter 10.5' [43]. However, as this method does not take into account multi-ore deposits and contamination penalties it was extended.

The main calculation steps in determining the economic value of each minable block are as follows:

- Calculate the raw block tonnage
- Calculate the costs for mining the block
- Calculate the recoverable metal

- Calculate the metal revenue based on the metal price
- Calculate the amount of concentrate produced from this block
- Calculate the arsenic grade in the concentrate
- Calculate the concentrate value penalty related to the arsenic content
- Calculate the net value of the block if sent to the mill

The exact calculation scheme can be found in Appendix H.

The Neves-Corvo mine has independent production lines and facilities for treating copper and zinc ore. To accommodate for this, for each block two independent values are calculated: the value of the block if it were mined as copper ore (based on a block's copper grade) and the value of the block if it were to be mined as zinc ore. From these two values the highest is taken as net block ore value and the block is designated as either copper or zinc ore.

When ore grades are low however, processing and concentrating a block can be more expensive than mining it without further treatment, as waste. In that case, when the net value of a block is lower than the costs for mining, mining costs are taken as block value and the block is designated as waste.

4.5.1. Development costs

In calculating the block values development costs were not incorporated. This was done for two reasons.

First, mining scheduling was not intended for the stope layout optimizer presented here. If scheduling were to be done in a later stage, it would use the optimal stope layout - in combination with a starting location - to find the optimal mining sequence. In this additional optimization course, unviable stopes could then be excluded.

Secondly, if development costs were to be taken into account, it would be more suiting to do this in the string evaluation phase by adding discount or penalty factors based on the distance between stope and starting location.

4.5.2. Arsenic content penalty function

The ore of the Neves deposit has been classified as having high arsenic values. The report from Geovariances shows an average arsenic content in the ore of around 0.7%.

Arsenic is an unwanted element in copper and zinc concentrates as it is harmful to the metal smelting process. Metal smelters therefore apply a penalty fee if arsenic values in metal concentrate are too high. In general a 'no penalty' threshold of 0.2% is used. For all concentrate having higher arsenic concentrations a penalty fee needs to be paid. The highest value accepted is 0.5% arsenic content by Chinese smelters. As the Neves orebody contains such high arsenic grades it would render most of the ore unsellable. To overcome this, blending of ore with other - lower in arsenic content - ore is done before concentrating.

The high arsenic concentrations mean a penalty function should be implemented for creating the economic block model. However, no public information could be found on the specific methods applied by Neves-Corvo, nor on any penalty sums being paid. Therefore it was decided to use the data as is and tailor a penalty function around the available arsenic grades.

The developed penalty function itself is straightforward; any block containing within penalty limits arsenic content is penalized linearly for the 'over limit' arsenic concentrate. This means that if a block has an arsenic content of 0.3%, a penalty is paid for the 0.1% over limit. If arsenic content were to be 0.4% the penalty cost would be double. Ore above 0.5% is classified as waste directly with a fixed mining cost.

4.5.3. Application of cut-off grades

Due to varying arsenic grade throughout the deposit it would make little sense to use a fixed cut-off value. Where in reality the ore from different resource deposits is blended to keep the arsenic content below penalty grades, the single deposit used for this research has an average arsenic content far above the penalty limit. As ore blending is not considered, this would mean that for calculating the cut-off grade the average arsenic grade should be used. However, by doing so the spatial distribution of arsenic would be abolished and cannot

be used as a factor in optimizing stope locations.

To avoid this, a cut-off value is considered per block. By subtracting the processing costs, mining costs and arsenic penalty costs from the block revenue, a net block value is calculated. If this value is not positive the block is classified as waste and a fixed cost is assigned. Else the block is treated as ore and its value used in calculating stope profitability. Depending on mining considerations the threshold net block value can be changed.

4.5.4. Cost calculation parameters

Table 4.2 lists the economic block model input parameters and gives an explanation on how they were derived. The parameter origin is listed in the sources column.

Table 4.2: Cost calculation parameters

Parameter	Value		Description and remarks	Source
	Cu	Zn		
Refining Cost [€/tonne]	188	188	Cost of refining a tonne of ore	Infomine [17]
Processing Cost [€/tonne]	7.27	12.76	Cost of processing a tonne of ore – Based on old plant capacity	Neves-Corvo 2007 technical report[44]
Mining Recovery [%]	95	95	Percentage of ore recovered from a block when mined	Neves-Corvo 2013 technical report [54]
Processing Recovery [%]	86.94	64.98	during processing and treatment – for copper 2000-2012 average, for zinc 2006-2012 average	Neves-Corvo 2013 technical report [54]
Metal Price [€/tonne]	5422	2584	Price per tonne for high grade tradable metal	London Metal Exchange (December 2016) [12],[19]
Mining Cost [€/tonne]	35.53	27.11	Cost of mining a tonne of material	Derived from combined mining and processing cost
Arsenic Penalty [€/0.01%]	2500		Penalty cost for the inclusion of arsenic in resource blocks. More arsenic means higher penalty costs. Blocks with >0.5% arsenic are rejected.	Tailored to the dataset
Block Dimensions [m^3]	64		Dimensions of a block in the economic block model	Geovariances Statistical Report[25]
Rock Density [tonne/ m^3]	3.3		Average density of a cubic meter of rock within the deposit	Neves-Corvo 2013 technical report[54]

4.6. Parent Generation

As part of the initialization, together with importing the resource model and setting the parameters described in section 4.12, the first generation of parent chromosomes is created. Depending on the population size a number of possible solutions are created randomly. The parent chromosomes are coded as binary strings where a 1 indicates the mining of a block (inclusion of a block within a stope, and a 0 represents no mining. Despite the fact that the stope positioning in these first generation strings is random, the placement of the 1's and 0's is not. This is due to the geotechnical constraints pillar size and stope dimensions which require a logical placement of the binary numbers.

The following section describes the module for initiating the first generation of binary chromosomes and will elaborate on some of the choices made in its design. A schematic overview can be found in appendix C, figure C.5.

First an empty matrix is created for storing the different parent chromosomes. The size of this matrix depends on the amount of parents (rows) and size of the deposit represented by the amount of mineable blocks (columns).

Next, the random parent chromosomes will be generated one by one. This is done on a 2-D grid as it requires less coding constraints and is easier to overview. Based on the deposit layer dimensions, a 2-D grid is created in which all cells have a value of zero. A loop is then started to fill the empty base layer with viable stopes.

To start, the loop termination criteria are checked. These criteria are the number of stopes and the number of loop iterations. If either one of these is fulfilled, the loop is terminated. Next the stope location and stope size are determined. The location is selected by randomly generating an x and y coordinate, the stope size is set by randomly generating a length and width. The stope sizes are limited by the minimum and maximum allowable stope sizes selected as model input.

When both location and size are determined an attempt is made to place the stope on the base grid. To place the stope two criteria need to be met. A stope is not allowed to overlap another stope and there needs to be a certain spacing (pillar) in between the individual stopes. The pillar size itself is fixed for all stopes and independent on stope size. When met, the random stope is created by setting the grid values corresponding to the stope to one. If not met, the program will loop back to the termination criteria.

This process is repeated until either the desirable amount of stopes is reached, or the number of iterations are maxed.

After the base layer has been filled with stopes it is reshaped to a 1 dimensional string and stored in the matrix for parent chromosomes. It forms the basis of the next algorithm steps.

4.7. String Evaluation and selection

In standard form, string evaluation in binary genetic algorithms is straightforward. 2 strings are presented; one representing an individual (a binary chromosome), the other containing property values. When these two strings are multiplied it results in a new string containing property values where the chromosome is turned on, and zeros where the chromosome is turned off. Summing up the latter string results in the value of the respective individual. Coupling this value to the chromosome string allows for ranking the individuals.

The stope optimization algorithm scripted works similar to the case described above. In 2D matrix form. In the chromosomes, stopes are presented as a patch of 1's and pillars and other empty area's as zeroes.

The grade models available (Cu, Zn and As) are combined and an economic block model is formed in which the individual block values are defined. Either for one E-type model, or for a number of simulations. Both these matrices are then transposed to strings after which they are multiplied and summed. The value of each individual is used to rank the individuals for roulette wheel selection applied in the cross-over algorithm.

Ranking is based on selection probability, calculated using formula 3.1. Based on relative value and selection pressure the probability of selection is calculated. the better the individual solution provided, the higher this probability will be. The selection pressure acts as a measure to indicate the relative difference between good and bad solutions. A higher selection pressure will disproportionately favour good answers in assigning the probability of selection.

The solutions are then ordered based on the selection probability, with the highest ranking individual on top.

4.8. Cross-Over Algorithm

In the cross-over module of the algorithm individual chromosomes are cut open and recombined to create a series of new individuals. The number of strings selected for cross-over depends on the individual fitness score calculated for each string based on their performance relative to the other chromosome strings. Shuffling genetic information between strings aims to create better individuals without losing the genetic information present. The cross-over operator developed for this research is a single-point cross-over operator. A multi-point cross-over operator was considered, but it was decided to encourage search space in the mutation operator instead as this allows for more flexibility in search space emphasis. In the section below the structure of the cross-over operator will be explained in detail.

After importing the chromosomes, a maximum selection probability is determined. As strings are taken out for cross-over, their probabilities of selection are summed until a maximum is reached. Not only does this

value indirectly determine the number of strings selected for cross-over, but it also determines the diversity of chromosomes chosen.

Next it is determined whether the individuals will be cut horizontally or vertically, both equally probable, and the cutting range is determined. The cutting range determines the limits of where the chromosome matrix can be split, and usually ranges between 0.1 and 0.9 times the model dimension.

Two adjacent strings are then taken out and the summed selection probability is updated. A cutting point is then determined randomly and the selected chromosome strings are converted to matrices. Both matrices are then split on the cut point resulting in 4 halves (1.1, 1.2, 2.1 and 2.2). These are then recombined (1.1 + 2.2 and 1.2 + 2.1) to form two new chromosome matrices. The matrices are then converted back to strings and stored at their previous location within the matrix containing the individual chromosome strings.

Two versions of cross-over operator were coded. The first being a random roulette-wheel function, where the string pairs are taken out at random, not taking into account their ranking order apart from their selection probability. The other taking out the strings in order of their sorting.

A flowchart of the scripted cross-over algorithm can be found in appendix C, figure C.6.

4.9. Mutation Algorithm

The aim of mutation in genetic algorithms is to maintain genetic diversity from one generation to the next. It does so by introducing new 'random' traits to genetic chromosomes. Mutation occurs based on a certain probability, which is usually low. If set too high too much of the old information is overwritten and the algorithm will behave more like a random search operator. For the stope optimization algorithm two mutation operators were initially coded; a classis flip-bit operator and a custom random function.

In flip bit one or multiple random chromosome operators are selected and reverted; if the cell contained a 1 it is reverted into a 0 and vice versa.

In the model operator, all binary chromosomes are evaluated one by one. Each time one of the individuals is taken out the operator will pass through all binary cells drawing a random number. If this number is smaller than the mutation probability set the cell value will be changed from a zero to a one or the other way around.

The bit flip algorithm however was rendered insufficient the way it was coded as, when stope sizes larger than 1-by-1 block are selected, the algorithm would lead to non-uniform stopes. These in turn would trigger the function constraining uniform stope sizes which, while evening out anomalies, would eventually erase all stopes.

A better solution was found in the random mutation operator. This operator, instead of changing individual cell values, places random stopes directly over the existing stope pattern. The basic operator for placing random stopes was influenced by the parent chromosome operator and therefore carries a lot of resemblances.

The operator module starts by generating a random number to decide whether to perform mutation for the current generation. If the falls out of the mutation range the operator will be immediately terminated and the algorithm will finish its current generation without. If this is not the case the number of individuals to mutate will be decided. A minimum of individuals can be set to guarantee mutation will be performed at all. The strings to mutate will then be taken out in order of appearance and recomposed to a grid matching that of the deposit size. Based on the amount of mutation operations set to perform (the amount of random stopes to be placed), random coordinates and stope size are determined. Depending on the pillar size the stope patch is extended with a pillar boundary around it and placed on the randomly assigned grid location. If the set number of mutations have been executed the individual's grid is recomposed into a binary string and placed back in the original population matrix.

The number of mutation operations to perform on each individual can be set in the algorithm as fixed value. However, as different inputs have frequently been used, it was found to be more convenient to have this parameter dependent on the size of the block model and have it created automatically. Caution was taken here to make sure the number of mutational operations increased with larger data files.

Figure C.7 from appendix C presents a flowchart of the mutation algorithm.

4.10. Constraint function Algorithm

A constraint function was written to cope with the errors posed by the chromosome altering operators. In the creation of first generation chromosomes care was taken to ensure rectangularity of the stopes and to have a fixed pillar between them. However, the recombination of chromosome parts during cross-over violates both of these constraints. Also, where mutation guarantees the placement of feasible stopes and pillars, it does not consider the viability of its borders.

To avoid disturbing the randomness of both operators it was decided to, instead of altering both the cross-over and mutation operators, write a single constraint function eliminating all infeasibilities. Various attempts were made in the coding stage to design an operator capable of eliminating all possible errors. This was made more complex by the strain most early versions would cause to the model run times. The definitive constraint function uses a shadow grid to track the corrected area's decreasing the number of loops, and therefore the runtime required.

The constraint function starts by loading the dimension data, stope size constraints, preferred pillar size and the population of individual solutions. The strings are taken out one by one and converted to 2 dimensional matrices. Based on the size of the individual matrix, a matrix full of zeroes is summoned. This 'shadow matrix' is used to keep track of the corrections made and determines when an individual is fully corrected. For the individual chromosome matrix a starting position is chosen and the first stope is located by walking from the starting point in x and y directions until no stope has been encountered for 2 steps, or, the maximum size has been reached. The function then determines the optimal size for the stope considering all irregularities. If the stope is too small, the stope is erased from the individual matrix, and the change locations are flagged in the shadow matrix. In all other cases, a border is forced around the optimal stope, and both stope and border areas are flagged as done. For each iteration the value of the shadow grid is checked. If the shadow grid is full, the function will loop back and take the next string.

This process is repeated until all individuals have been corrected.

Figure C.8 from appendix C presents a flowchart of the mutation algorithm.

4.11. Elitism

Elitism in genetic algorithms is a way to ensure a good solution is not accidentally lost or negatively altered. Usually this is done by taking out one or multiple good genetic solutions before the cross-over and mutation operators are applied, and re-inserting them in the next generation unaltered. As the name suggests, usually the best solutions of a generation are taken out to be carried over to the next generation.

The genetic algorithm coded in this research has an inbuilt feature to enable and disable elitism of one or multiple solutions.

Elitism is applied by taking out a copy of the two best solutions of each generation after their individual value has been calculated and the solution sorted. These saved two best solutions are then kept and stored parallel to the selection, cross-over, mutation and constraint operator. After all genetic operators have been active and the generation's results have been documented, one or two of the saved solutions is reinserted or no action is taken (no elitism is applied).

If elitism is applied (saved individuals are re-inserted) they are re-entered in the population matrix replacing the chromosome(s) that were ranked lowest prior to the generation's genetic operations having taken place.

4.12. Input Parameters

The input parameters used for the economic block model creator and optimization algorithm are listed in table 4.3 and table D.4. The parameters are named as they are found in the respective Matlab scripts and are accompanied by a short description. In addition, for each algorithm parameter listed in table 4.3, mention is made of where in the model these parameters are used.

Table 4.3: Explanatory list of input parameters

Parameter model name	Model Assignment	Description
x_model	Parentstring generation, Crossover, Mutation, Constraint Function	Horizontal size of the value model used and stope layout dimensions. Assigned automatically
y_model	Parentstring generation, Crossover, Mutation, Constraint Function	Vertical size of the value model used and stope layout dimensions. Assigned automatically
string_length	Model Initialization, Mutation	Length of the binary chromosome string. Function of x and y dimensions
iter_model_max	Termination Criterion	Maximum number of algorithm iterations allowed
no_change_lim	Termination Criterion	Maximum number of algorithm iterations in which no positive value change was monitored allowed
stope_max	Parentstring generation	Maximum number of stopes allowed in first generation individuals
iter_max	Parentstring generation	Maximum number of stope placement attempts allowed while composing a first generation individual
x_stope_max	Parentstring generation, Mutation, Constraint Function	Maximum horizontal stope size allowed
y_stope_max	Parentstring generation, Mutation, Constraint Function	Maximum vertical stope size allowed
x_stope_min	Parentstring generation, Mutation, Constraint Function	Minimal horizontal stope size allowed
y_stope_min	Parentstring generation, Mutation	Minimal vertical stope size allowed
x_stope_avg	Model Initialization	Theoretic average horizontal stope size
y_stope_min	Model Initialization	Theoretic average vertical stope size
barrier	Parentstring generation, Constraint Function	Pillar size; number of blocks to be left unmined between individual stopes
nrpar	Parentstring generation, Value Calculator, Crossover, Constraint Function	Size of the chromosome pool. This should be an even number.
crossover_probability	Crossover	Probability of crossover being executed during an iteration
p_low_lim	Crossover	The lowest cumulative P-value allowed. Forces a minimal number individuals to be crossed-over
cutting_range	Crossover	The range on which the cut-point can be selected
chance_mut	Mutation	Probability of mutation being executed during an iteration
chance_mut_flipbit	Mutation	Probability of flip-bit mutation being executed during an iteration
nr_mut	Mutation	Number of stopes to place on an individual if it is selected for mutation
perc_mut	Mutation	Percentage of individuals to be selected for mutation if mutation is executed
min_par_to_mut	Mutation	Minimal number of strings to mutate if mutation is executed
selection_pressure	Selection	Determines the probability of the best individual to be selected relative to the average selection probability
nr_sim	Value Calculator	Number of simulations taken into account when running stochastically. For average cases nr_sim = 1

Table 4.4: Input parameters economic

Parameter Model Name	Model Assignment	Description
rock_density	Economic Blockmodel Creation	Average density of a cubic meter of rock within the deposit.
block_size	Economic Blockmodel Creation	Size of a block within the economic block model.
mining_recovery	Economic Blockmodel Creation	Percentage of material recovered after a block is mined.
arsenic_penalty	Economic Blockmodel Creation	Penalty value for the inclusion of high levels of arsenic within a block of ore. Penalty dependent on amount of arsenic present.
processing_cost_zn	Economic Blockmodel Creation	Cost of processing a tonne of ore in the zinc processing plant.
processing_cost_cu	Economic Blockmodel Creation	Cost of processing a tonne of ore in the copper processing plant.
mining_cost_zn	Economic Blockmodel Creation	Cost of mining a block of zinc ore. Derived from the mining and processing cut-off parameter.
mining_cost_cu	Economic Blockmodel Creation	Cost of mining a block of copper ore. Derived from the mining and processing cut-off parameter.
mining_and_processing_cutoff_zn	Economic Blockmodel Creation	Limit cost of mining and processing a tonne of zinc ore combined.
mining_and_processing_cutoff_cu	Economic Blockmodel Creation	Limit cost of mining and processing a tonne of copper ore combined.
processing_recovery_zn	Economic Blockmodel Creation	Percentage of zinc ore recovered after processing.
processing_recovery_cu	Economic Blockmodel Creation	Percentage of copper ore recovered after processing.
metal_price_zn	Economic Blockmodel Creation	Value of a tonne of zinc.
metal_price_cu	Economic Blockmodel Creation	Value of a tonne of copper.
refining_cost_zn	Economic Blockmodel Creation	Cost of refining a tonne of milled/processed zinc ore.
refining_cost_cu	Economic Blockmodel Creation	Cost of refining a tonne of milled/processed copper ore.

Model validation and validation results

5.1. Model validation

5.1.1. Aim of validation

It is important to understand the accuracy and performance of the model. By validating the model its accuracy and performance can be defined. Schlesinger defined model validation as follows: “Substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model” [50].

During the model development stage several validations were done to ensure the theoretical working of the individual model parts. As a model can be tailored to the intended input it can happen that a model works well with one dataset, but fails to work on another. To ensure the working of basic principles different inputs were used during different modelling stages.

5.1.2. Validation procedure

Validation was done in multiple steps during the modelling stage to filter out potential theoretical mistakes. Apart from the random input used to check the working of individual algorithm modules (input-output validation) three datasets were involved with the model validation: A small Excel generated grid, an ASCII data file and the main CSV input files.

After the each of the following milestones were reached a validation was performed:

- Completion of the main genetic algorithm (Excel input)
- Completion of the stochastic algorithm extension (ASCII input)
- Completion of the intended model (CSV input)

In the following sections the individual validations and their procedures will be elaborated.

5.1.2.1. Validation using Excel import

The core genetic algorithm was tested and validated using a number of small excel files to represent a simplistic theoretic orebody. Several small excel grids were made ranging in size from 14-by-19 to 60-by-60 blocks. Theoretical waste zones are represented by negative block values of -10 and ore zones by a value of 10. The ‘ore’ zones are regular square shaped and do not exceed the maximum stope size selected in the model.

First these excel models were used to examine the performance of the genetic algorithm itself. By distinguishing clear ore and waste zones not exceeding the maximum selected stope size, a straightforward optimal solution is presented to the model. As the ore value is fixed throughout the algorithm’s input, the search space is limited as the algorithm does not need to look for optimality within (positive) ore zones. This benefits the runtimes as theoretically less iterations should be required before an optimum is found.

Using small sized test inputs had two main advantages. First of all the small search space of the test case allows for much better model performance as less calculations need to be done per iteration. Also, as the smallest model only consists of three ‘ore’ patches it is very easy to see whether the optimum is reached, and if not, where the non-optimalties are located and of what nature they are.

Aside from using the Excel input to check the theoretical validity of the model these inputs were used to examine the influence of ‘elitism’ in different forms. By comparing the number of iterations required to reach an optimum it could be decided if eliting, and if so, what form of elitism would benefit the model most.

5.1.2.2. Stochastic validation using ASCII import

In addition to standard single layer optimization the basic genetic algorithm was extended to do stochastic optimization. Here, instead of using a single input layer, a number of simulations are each individually taken into account. This, contrary to the common practise in mining to use a single 'smooth' orebody approximation should lead to better optimization results.

In order to evaluate stochastic performance of the algorithm, stochastic runs will be compared to smoothed representations similar to the ones used in mining. The smoothed models are generated by averaging the simulations used in stochastic runs.

For this purpose an ASCII style data file was provided by the department consisting of 200 elevation model simulations. These were the outcome of a previous, non-related research.

Contrary to the Excel test model used in previous validation, the ASCII contained values were not edited prior to running, but instead used as is. This was possible as both positive and negative elevation values were present. As higher elevations result in linearly higher values, the model should optimize its stopes so that they are placed around the elevated (positive) areas and avoid the lower (negative) zones.

In addition to 200 elevation model simulations also the real case scenario was included. Besides enabling comparison of the optimization performance of full stochastic optimization versus smoothed E-type models, it helps placing its performance in perspective.

5.1.2.3. CSV File adaptation

The Neves-Corvo data provided for this research consisted of 3 individual CSV files containing block information of a 3D resource model on copper, zinc, and arsenic. In addition a model for sphalerite content was included, however this was not used in the main research analysis. For the main research the element data had to be combined to transform the element block model into an economic block model. This required several steps to alter the data so that combining data and calculating economic values was possible. During this stage several analyses / validations were performed on the data both to increase understanding of the data and to incorporate it into the genetic algorithm.

Layer data analysis

The main optimization algorithm scripted is capable of optimizing in two dimensions. However, as the provided data contains three dimensional grade data for the various elements a layer needs to be extracted.

The orebody itself is irregular in shape and contains several lenses. To evaluate the algorithm a good layer needed to be selected. In order to do so, 2D horizontal cross-cuts were made and layers were exported individually from the CSV file for evaluation.

Penalty value tailoring

Despite the arsenic grade data being included in the data model few further information was available on arsenic processing and penalty cost factors. As they are a factor to take into consideration when optimizing stope layout a penalty function was incorporated in the economic block model calculation. The function allows realistic penalizing when costs are known and take into consideration worldwide inclusion acceptance rates. To find a suitable penalty costs to use with the Neves-Corvo data economic block models were made with various penalty costs for arsenic inclusion.

Metal cost analysis

The metal prices used in calculating the economic block values were taken from the London Metal Exchange and are representative of the current economic metal market. However, as the block model calculator distinguishes copper ore, zinc ore and waste, fluctuating metal prices might play a non-linear role in the optimization process.

The influence of fluctuating metal prices on the economic block model were evaluated by doing several runs with various exaggerated zinc prices.

5.1.2.4. Performance tweaking

Runtimes

Initial coding and scripting was done with the main focus on algorithm functionality. This had as a result that numerous functions being used were negatively affecting run-times. Using in-built Matlab features a number of run-time profiles were made indicating calculation-time bottle necks, and efforts were made to replace them with alternative functions.

Parameters

The performance of a genetic algorithm heavily depends on the way selection, cross-over and mutation are executed. By using sub-optimal input parameters there is a risk of the algorithm getting stuck in a local optimum, or the algorithm relying too heavily on random search practice. Furthermore the selected parameters influence model run-times.

To find a good balance between performance and run-times small sensitivity analyses were done on the following parameters:

In the sensitivity analyses, a small theoretical Excel orebody representation was used (see section 5.1.2.1). This was done to decrease the run-time required and also to eliminate the solution value as measure of performance. As the optimal value is known, performance can be measured by the number of iterations required to reach the optimum.

5.2. Model validation results

The validation analyses as described in the methodology were mostly performed with a fixed parameter set. The parameter values as used are described in table 5.1 below. Note that the number of mutation operations performed per mutation (nr_mut) is always a variable based on the model and average stope sizes.

Where in the validation analyses different parameter input has been used it is noted specifically.

Table 5.1: Parameter values for model validation

Parameter	Value
iter_model_max	25.000
no_change_lim	1.000
x_stope_max	5
x_stope_min	2
y_stope_max	5
y_stope_min	2
barrier	1
nrpar	50
crossover_probability	0.9
p_low_lim	0.4
cutting_range	0.9
chance_mut	0.2
nr_mut	$round(0.1 * (y_model/y_stope_avg) * (x_model/x_stope_avg));$
perc_mut	0.8
min_par_to_mut	2
nr_constr_loops	1
selection_pressure	1

5.2.1. Validating using Excel import

The aim of this analysis has been to explore the basic stope optimization performance based on a small test orebody and to confirm the main theoretical model workings. The following results are all the result of optimization runs on the smallest of Excel test orebodies with the aforementioned set of parameters.

5.2.1.1. Theoretical optimization performance

The small test-ore body used in this test figure 5.1 as was generated in excel is 14-by-19 blocks in size and contains only two different values. Yellow areas represent ore zones and consist of blocks with an individual value of 10. The blue blocks represent waste and have a value of -10. 39 ore blocks are present bringing the total theoretical value up to 390. This is the optimization goal.

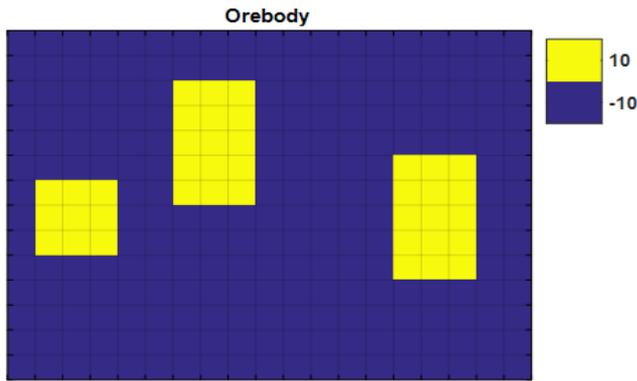


Figure 5.1: Orebody

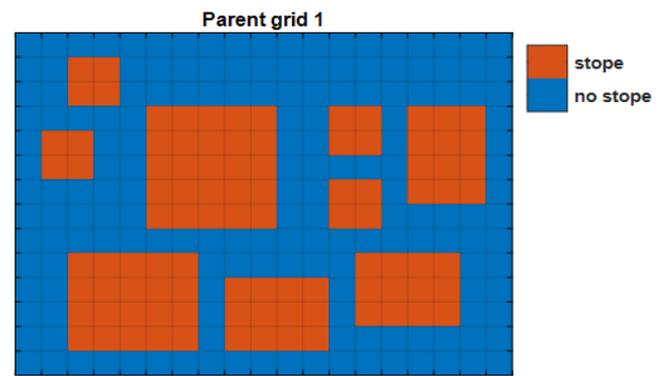


Figure 5.2: parent grid 1

Figure 5.2 shows one of the 50 first-generation parent chromosomes, plotted in 2 Dimensions. Most of the generated stopes within this plot only overlap partially with the 'ore' zones in the orebody grid or do not overlap any at all. The worst of the initial parent chromosomes represented a value of -780.

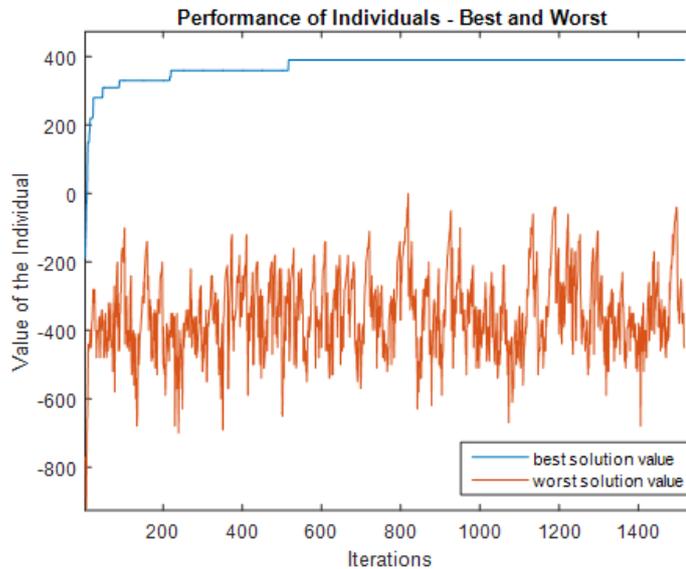


Figure 5.3: Performance of individuals best and worst

Figure 5.3 pictures the optimization process development. Over the course of iterations (generations) the best and worst solution values are plotted for each iteration. The blue line shows a steady and steep incline in the first 80 iterations to then slowly flatten out around 300 iterations. The optimum value of 390 is reached around the 550 iteration mark. However, due to the no change limiter being set at 1000 iterations, the model continues to run until around the 1550 iteration mark.

Over the course of the optimization process the worst individual fluctuates between a value of -700 and 0. This indicates the model remained active after the the optimum solution was reached.

Figure 5.4 and figure 5.5 show the best and worst individual solutions at the end of the optimization run. Figure 5.4, the best solution has the maximum value of 390 as it overlaps the orebody exactly. Figure 5.5 on the other hand has negative value as there is more overlap with negative orebody blocks than there is with positive blocks.



Figure 5.4: Ranked individual 1 (Best)



Figure 5.5: Ranked individual 40 (Worst)

5.2.1.2. Eliting

Using the same test orebody as was used in the theoretical optimization performance validation the effect of enabling and disabling elitism was studied. For this purpose the lines for elitism present in the script were either switched 'on' or 'off'. No input parameters were changed.

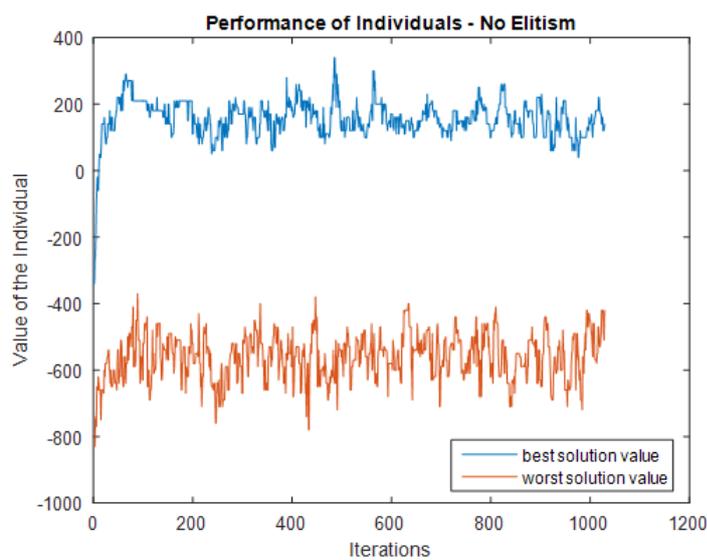


Figure 5.6: Performance of individuals - no elitism

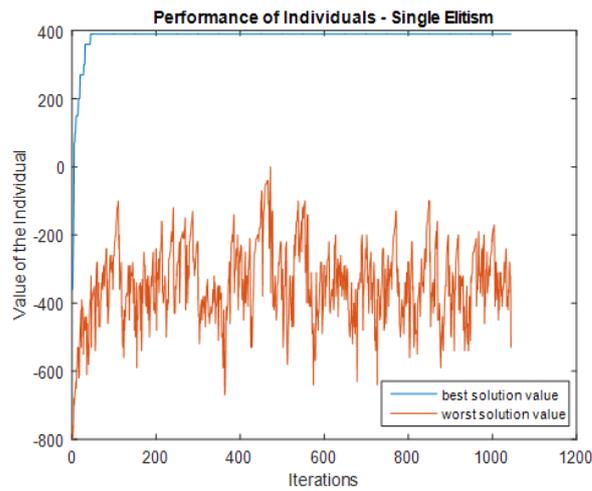


Figure 5.7: Performance of individuals - single elitism

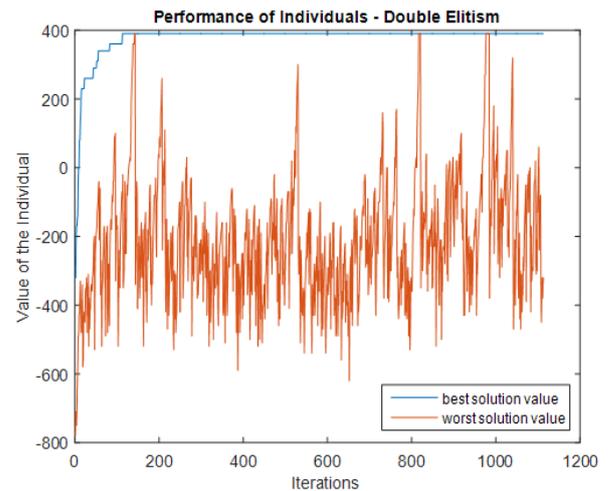


Figure 5.8: Performance of individuals - double elitism

Figures 5.6, 5.7 and 5.8 show the optimization process development for optimization done on identical ore body layers. Figure 5.7, where single elitism is applied shows the same pattern and behaviour as was found for theoretical model testing. The optimum was reached between 50 and 80 iterations. However, with no elitism applied figure 5.6 displays an inability of the model to reach the optimum, with best individual values fluctuating between 300 and 0. Compared to the case where elitism is applied, the worst solution values are also lower. However, fluctuation is less and its value range comparable than that for the best individual.

As the best solution values were frequently going down the model, optimization stopped before the appropriate no change limiter value was reached.

With double elitism enabled (figure 5.8), the model manages to reach an optimum value in under 200 iterations. Worst solution value fluctuations seem more extreme than is the case with single elitism or no elitism. The worst individual equals the best individual on 3 occasions. As individuals are sorted this would mean a temporary unavailability of genetic variation.

For the model it was decided to use single elitism. Where converging was concerned, rates for both single and double elitism were similar and comparably variable. However, double elitism shows a decrease in genetic variability compared to single elitism. Where with large population sizes this should not have a large impact, it can become critical when smaller populations are used.

5.2.2. Validating using ASCII import

For the stochastic model application testing and validation, no alterations were made considering the parameters listed at the beginning of the chapter. However the Input was changed to a set of 200 elevation model simulations. In addition to the simulations a representation of the real scenario was also included. This allowed not only for real case optimization, but also including the real scenario (Figure 5.13) in the stochastic and average case optimizations.

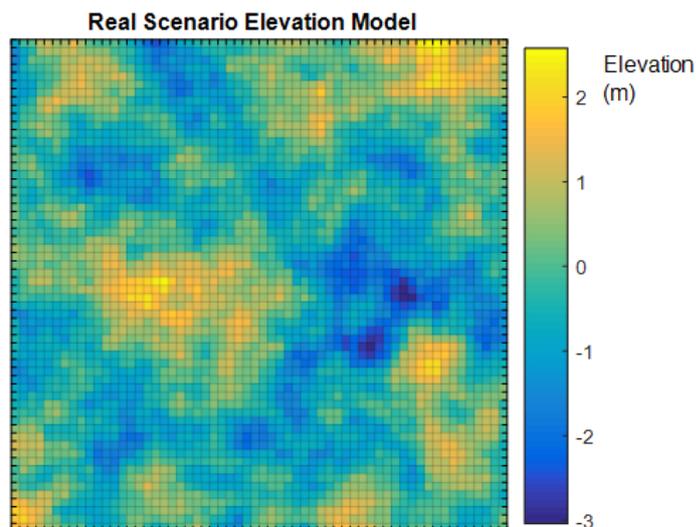


Figure 5.9: Real scenario elevation model

Examples of a provided elevation model simulation and the averaged (E-type) layer used are displayed in Figure 5.10 and Figure 5.11. Notice the change in elevation scale as the elevation range of the E-type model is smaller than that of simulation 128.

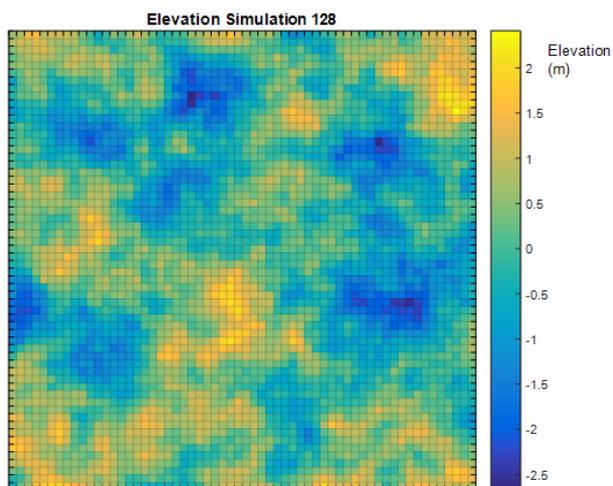


Figure 5.10: Elevation simulation 128

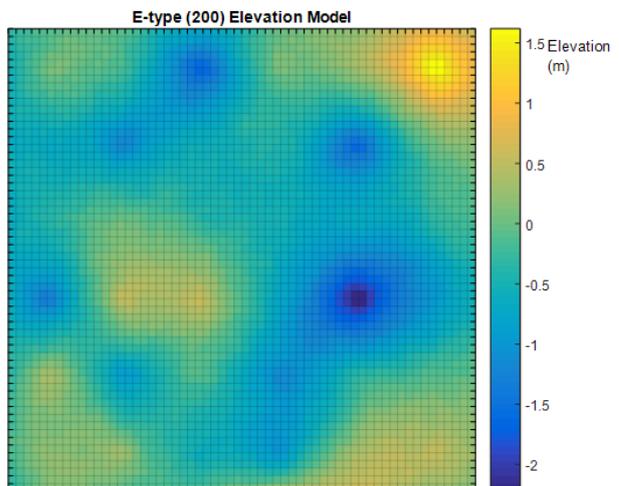


Figure 5.11: E-type (200) elevation model

5.2.2.1. Real elevation model

After preparing the simulation data to make it compliant with the model a number of optimizations were done. First the real elevation model was optimized. Figure 5.12 and Figure 5.13 show the realization over which was optimized, as the resulting best solution. When studied both, it can be noted that the stopes seem to be placed well. All positive elevation zones seem to be covered by a stope, while the low (negative) elevation areas are well avoided. Purely based on the raw elevation data the optimized solution summed up to be 486.1083.

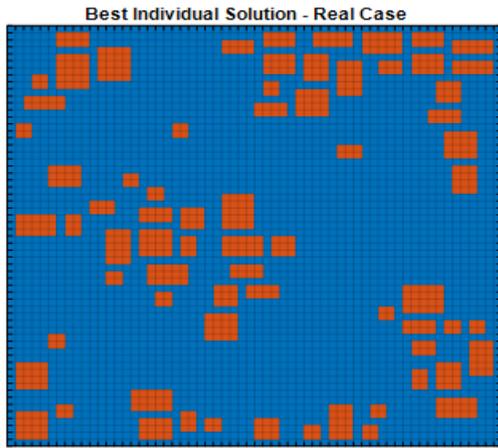


Figure 5.12: Best individual solution - real case

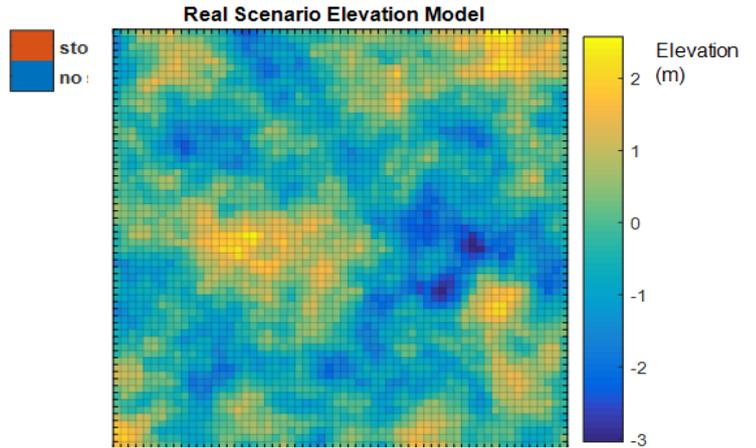


Figure 5.13: Real scenario elevation model

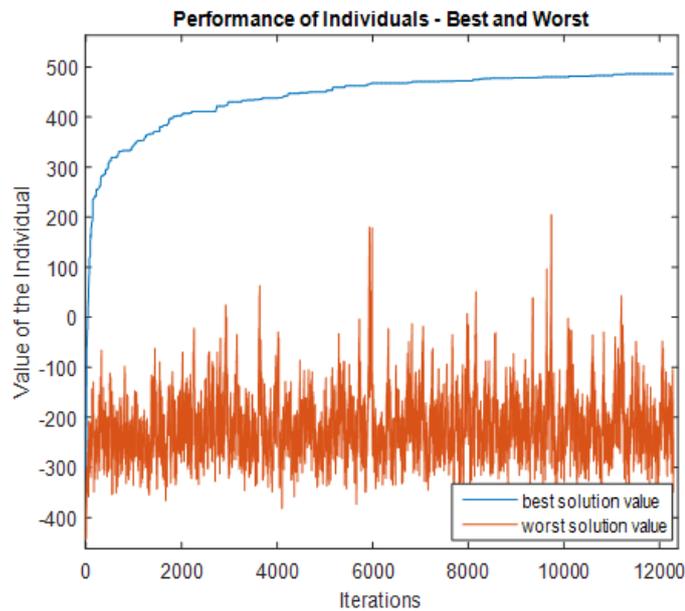


Figure 5.14: Performance of individuals - best and worst

The performance graph (figure 5.14) further showed good progress. After a steep performance increase in the first 500 iterations the optimization progress increasingly slows down until it almost flattens out around 10,000 iterations. The worst individual value mainly fluctuates between a value of -350 and -50. Some spikes are observed.

5.2.2.2. E-type analysis

Following, an analysis was done on the E-type layer of all 200 simulations. Figure 5.15 displays the stope layout of the best individual solution and Figure 5.16 includes the relating performance graphs. One including the performance of the best and worst individual solutions (Figure 5.16a), the other relating the performance of the best individual solution to the real value the stope layout would have if it was used on the real case (Figure 5.16b).

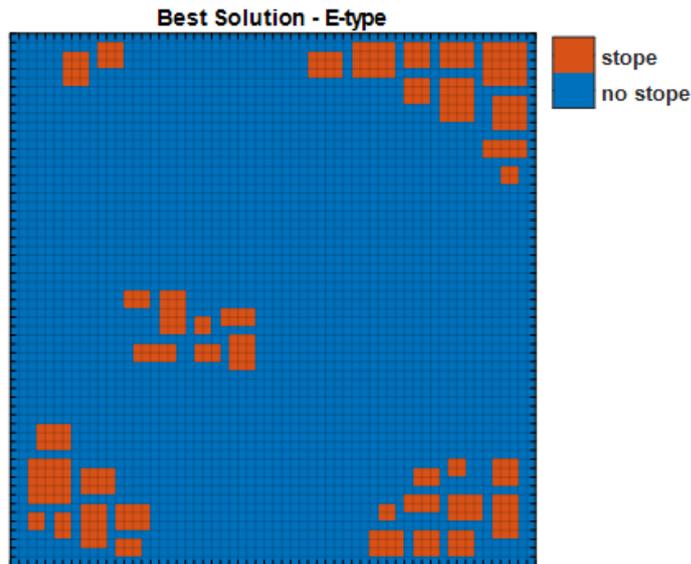
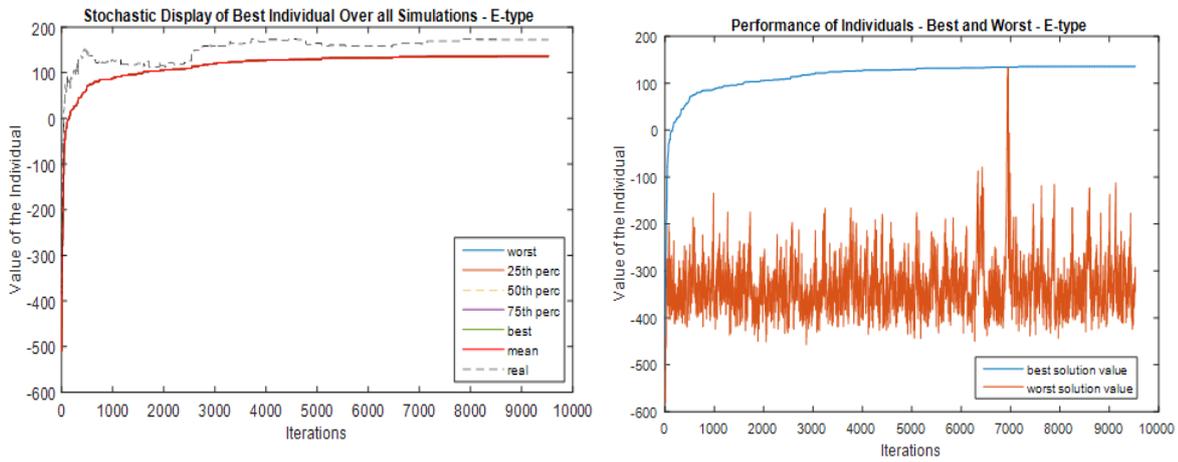


Figure 5.15: Best solution E-type



(a) Stochastic display of best individual for real case scenario over all simulations - E-type

(b) Performance of individuals - E-type

Figure 5.16: Performance graphs

Looking at the performance graphs no unusualities are shown apart from one instance around 7000 iterations, where the worst individual solution equals the best solution. Optimization pace is comparable to the real elevation layer optimization although the final optimized value is much lower.

The stochastic display graph compares the E-type optimization pace with the value the best individual would result in when plotted on the actual elevation model. Where the optimization value (mean) only increases, more variation can be seen in the dashed line representing the real optimization value. Over the course of optimization it remains above the E-type solution indicating better value. It's value is however not as constant as the E-type solution and deteriorates both around the 600 and 4700 iteration point.

The additional labels in the legend (25/50/75 percentile, best and worst) are included as the graph doubles purpose for stochastic runs.

5.2.2.3. Stochastic

Lastly a stochastic analysis was done using the ASCII elevation model dataset. Here all models are taken into consideration as their value is calculated based on the average of all individual values. Apart from the usual

performance and stope placement graphs it allowed monitoring the solution distribution over the optimization process. Figure 5.17 shows the performance graph for the stochastic optimization.

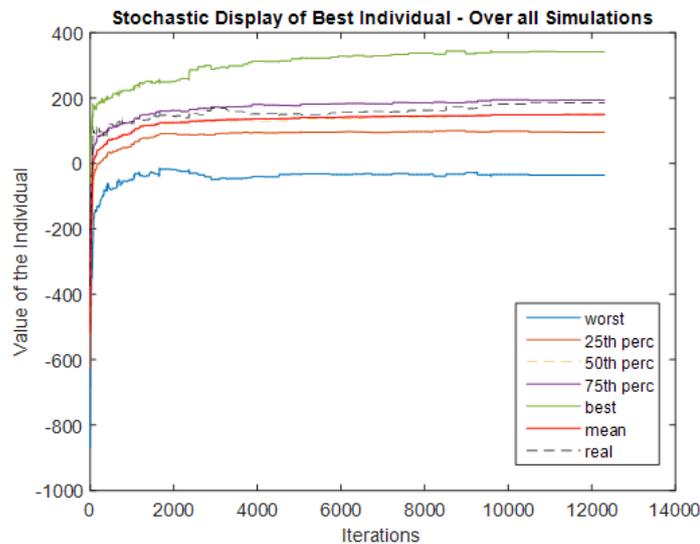


Figure 5.17: Stochastic display of best individual solution projected over all orebody simulations

Where the average value over which is optimized flattens out from 2000 iterations onward, the influence of optimization on the individuals remains active until the 3000 iteration mark. Furthermore, the value of the mean optimized design for the best simulation continues increasing more than that of the mean itself. The value of the mean optimized design over the worst simulation remains constant from 5000 iterations on. The distribution of solution values keeps widening over the model run.

The real solution value for the mean optimized result is higher than the optimizer value as was the case with the E-type optimization.

Another feature of the stochastic optimization module is that, as it calculates the value for all individual simulations, individual performance can be monitored. When the stochastic design is placed on each individual simulation its value can be calculated. In the aforementioned performance graph this was displayed by plotting the percentile values in the graph. In addition this data can be used to plot histogram data. This can be done for either the E-type and Stochastic run results.

For Figure 5.18 stochastic data from 3 E-type runs and 3 Stochastic results have been merged and plotted as histograms. The blue histogram up front shows the count of solutions based on value range. As data from 3 stochastic runs have been added the total number of solution count is 600. Peaks indicate a large number of solutions within a certain value range with the highest peak being for solutions based on the optimized model between 98 and 123. In that range 62 models complied. The shape of the histogram resembles a normal distribution.

Behind the histogram for stochastic data is the histogram for E-type data. Based on the optimized-type model all related individual values were calculated and plotted in a Histogram. As this histogram does not peak as high the distribution is slightly more spread.

Based on the histogram data normal distributions were fitted over the histograms in Figure 5.18. The blue dashed line relates to the stochastic histogram data, the red dashed line to the E-type histogram data.

The normal distribution fits seem to confirm the flatter architecture of the E-type histogram data as it peaks less high. However the curve range is not wider and the surfaces of both curves are not equal.

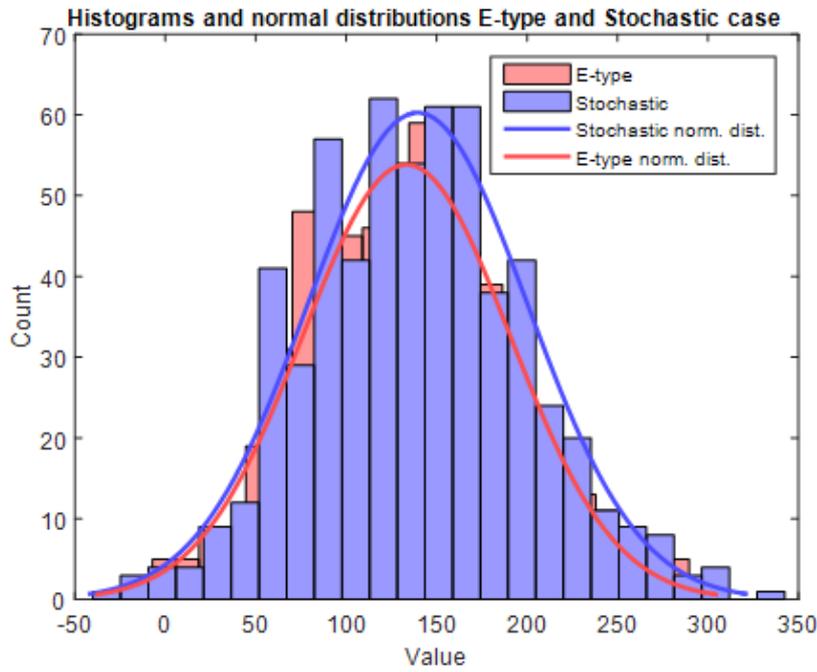


Figure 5.18: Histograms and normal distributions E-type and stochastic case

To equalize the distribution surfaces they were plotted separately in figure 5.19 based on the calculated means, and standard deviations. Two dashed lines were inserted, indicating the mean values of both E-type and stochastic data.

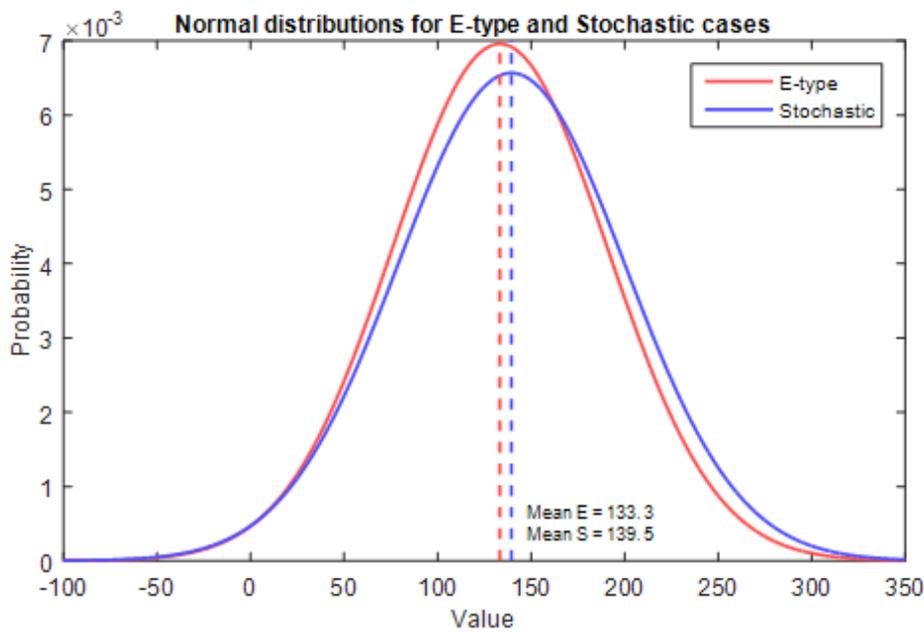


Figure 5.19: Normal distributions E-type and stochastic case

5.2.3. CSV File adaptation

The economic block model is calculated using copper, zinc and arsenic grade data. The three .csv files contained data for the full Neves orebody. The size of the individual files (over 2 GB each) meant they could not be imported as is, and thus had to be adjusted. From previous work done on smaller files it had become clear the layers of economic interest ranged from the 600 to the 800 layer.

After extracting the necessary data layers they were imported, analysed and converted to economic block model layers, to find a representative single depth layer to use in the stochastic analyses.

5.2.3.1. Parameters used

The parameters used for calculating the economic block models during validation are listed in table 5.2. An explanatory list of input and output parameters can be found in appendix D.

Table 5.2: Parameter values for calculating the economic block models

Parameter	Value
rock_density	3.3
block_size	4x4x4
mining_recovery	0.95
arsenic_penalty	2500
processing_cost_zn	-12.79
mining_and_processing_cutoff_zn	-39.9
processing_recovery_zn	0.6498
metal_price_zn	2584
processing_cost_cu	-7.27
mining_and_processing_cutoff_cu	-42.8
processing_recovery_cu	0.8694
metal_price_cu	5422

Where, in any of the csv analyses, different parameters were used it will be specified.

5.2.3.2. Layer data analysis

To explore the dataset a large number of independent data-layers were imported into Matlab and analysed. Figure 5.20. Shows a number of copper ore layers in the range of 600 – 800 (796) with an interval of 40 meters.

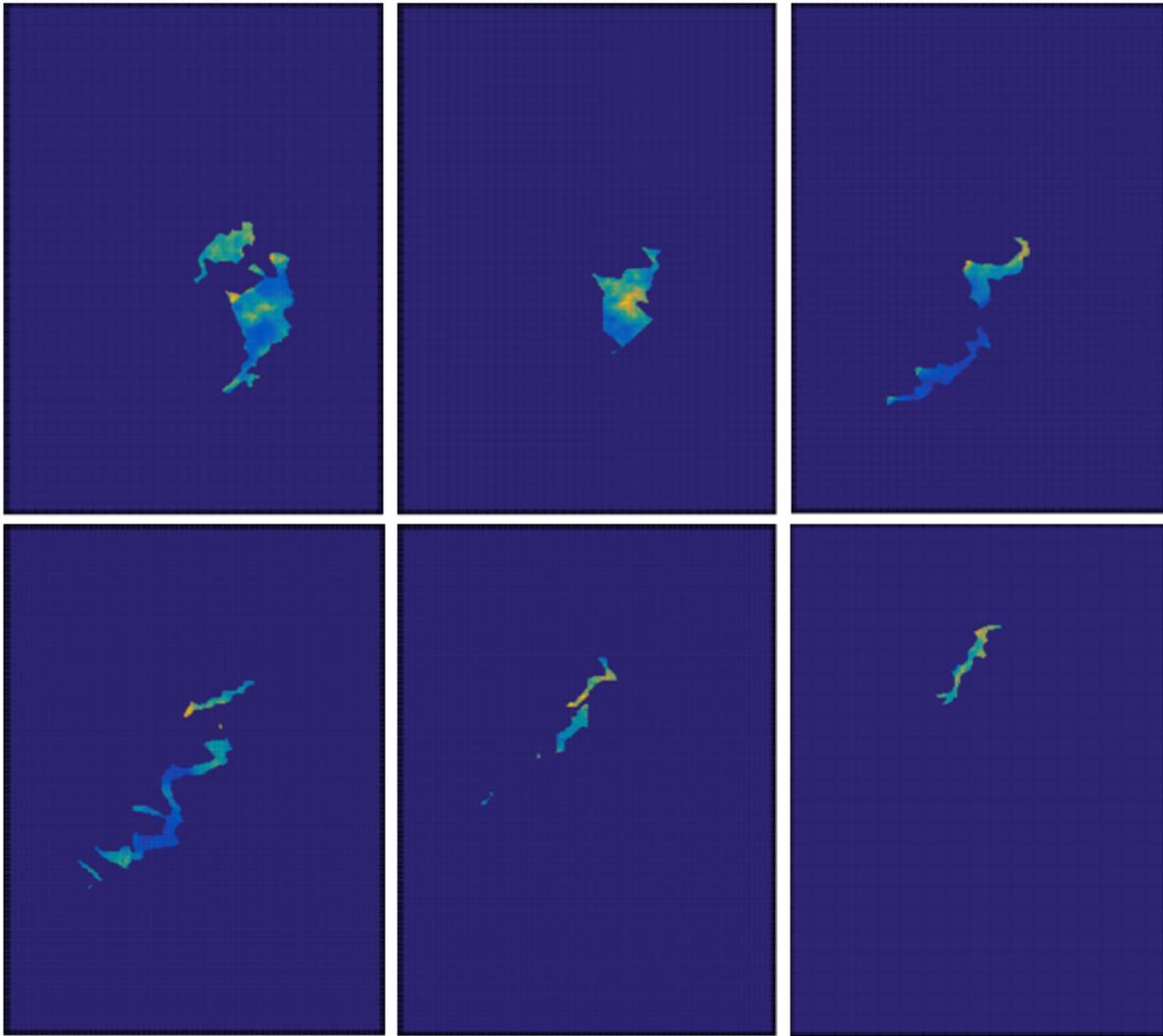


Figure 5.20: Independent data-layers

Using the different element layers economic block model layers and ore-waste layers were plotted next. Figure 5.21 shows the different element maps a simulation of the 720 layer, together with the resulting economic block model and ore-waste map.

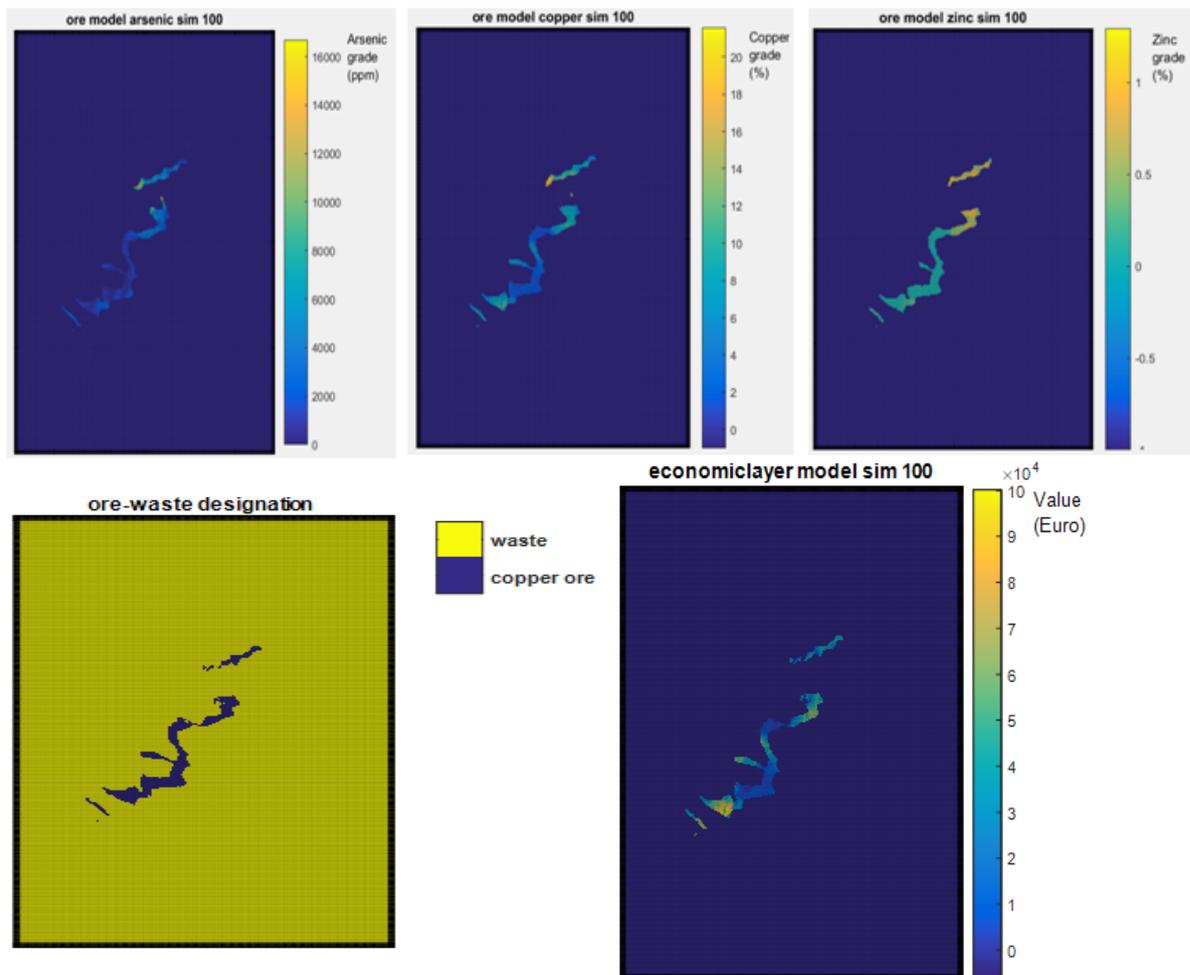


Figure 5.21: economic block model

Arsenic grades are given in parts per million (ppm). In later calculations the arsenic grades were converted to percentage grades. Zinc grades are very low barely exceeding 1%, where copper grades on the other hand range up to 20% in high grade zones. As a result of low zinc grades no zinc ore has been defined in the ore waste designation map. Arsenic high grade zones appear where copper grades are high, however, no clear relation can be seen as not in all higher grade copper ore zones high arsenic grades are monitored.

In the economic block model it the influence of arsenic on block value can be seen as the top zones with extreme copper content are not as valuable as the high copper grade zones in the lower part.

5.2.3.3. Penalty value tailoring

As ore from multiple orebodies is blended to keep arsenic contamination below penalty limits, Actual penalty values for arsenic content were not defined by the Neves-Corvo mine. However, as in this research only data from the Neves orebody was used a penalty function was implemented. The penalty value as stated in table 5.3 was used for further analyses, however, tests with various penalty values were run. Between 0.2 and 0.5% arsenic content was penalized based on amount of arsenic included. Above 0.5% the ore is automatically defined as waste. The penalty values tested for are listed in table 5.3.

Table 5.3: Penalty values

Penalty value	Penalty value (€) is applied linear per 0.01 % over the penalty threshold of 0.2%, up to the penalty limit of 0.5%.
No penalty	
10	
500	
1000	
2500	
3500	
5000	

Figure 5.22 and 5.23 show the influence of penalizing for arsenic content on the economic ore / waste distribution map. Areas with high arsenic content have been assigned to waste completely. Waste is showcased in yellow. Zones with 'over-waste' economic potential are shown in blue.

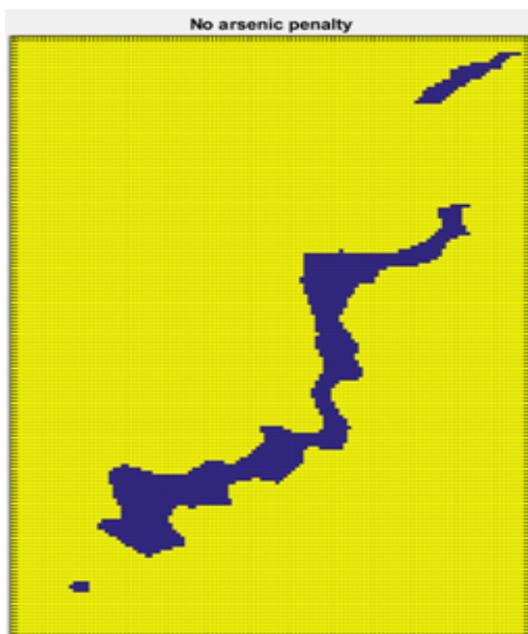


Figure 5.22: No arsenic penalty

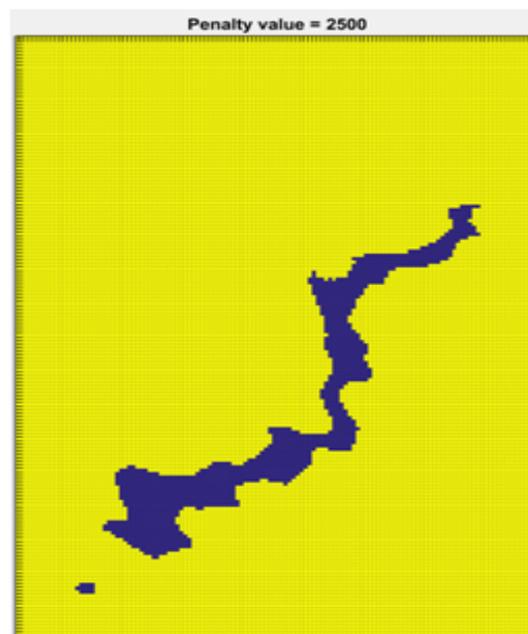


Figure 5.23: Penalty value 2500

5.2.4. Zinc cost analysis

For the dataset, measurements of both copper and zinc were included. However, after quick analysis and economic block model valuation, zinc grades appeared too minor to contribute. Testing the economic block model creator function required assessing the impact of zinc on the economic block models. Hence a number of small tests with various altered zinc prices were done.

Figure 5.24 and figure 5.25 show the ore / waste distribution maps for zinc grades 10 times and 100 times higher than currently (December 2016) listed on the LME. The distribution maps are for simulation 100.

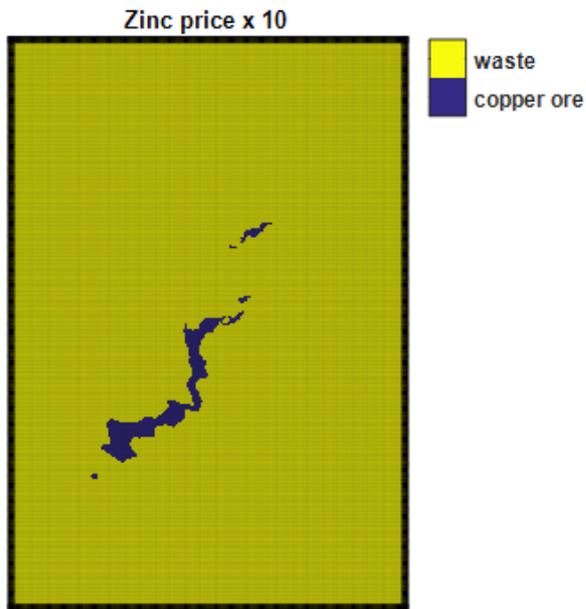


Figure 5.24: Zinc price x 10

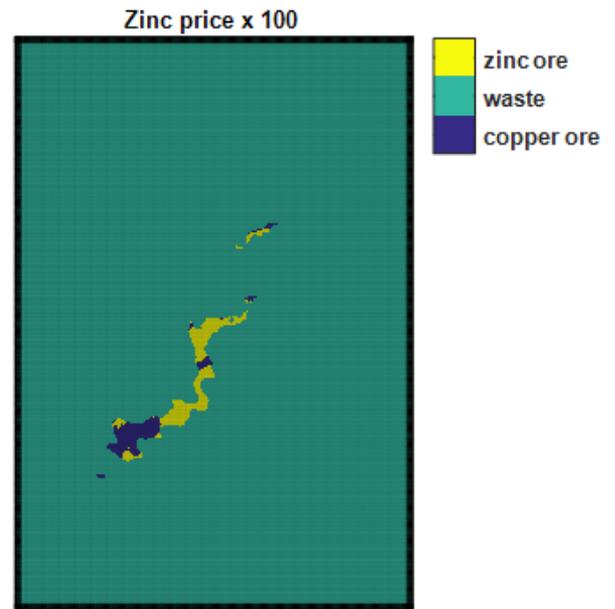


Figure 5.25: Zinc price x 100

Where standard zinc prices have been multiplied 10 times, no zinc ore is present. Zinc grades are too low to contribute to the copper ore / waste mapping. However, when the zinc price is further upped, despite zinc ore being of low grades it manages to contribute. Only in zones with zinc grades nearing zero copper ore is mined.

6

Methodology

6.1. Introduction

To explore the possibilities of stochastic techniques in underground mine design, and to review the benefit of using these techniques in comparison to traditional methods, several different methods for analysis were selected. Some of these analyses were influenced by previous research and adapted for use with the genetic algorithm model, taking into account the resources available and model limitations.

6.2. Analysis scope

In order to better understand the influence of grade uncertainty on underground mine design, and to see if the genetic algorithm scripted in this research is capable of dealing with it three independent analyses have been drawn up. With each individual analysis a different approach to grade uncertainty is taken, as well as each individual analyses explores a different potential of stochastic optimization techniques applied to underground mine optimization.

The first analysis, the 'Maximum upside potential / Minimum downside risk' approach, is used to explore the influence of grade uncertainty on mine design value and to see to what extent it differs from a traditional design approach. Furthermore it aims to select a final design that is most capable of dealing with metal price fluctuation.

Next, the influence of grade uncertainty towards other underground mining performance parameters is explored. The average grade, arsenic content, ore tonnage and metal content are reviewed for different equally probable scenarios in order to map the extent and potential influence of the present uncertainty. Furthermore the approach will be used to create a map of stope inclusion probabilities.

In the last analysis, instead of using, and optimizing for, individual stochastically generated orebody simulations, a single stochastic mine design optimization is done taking into account all simulations at once. This is done to map the added value of automated stochastic optimization when compared to a more traditional approach where individual simulations are averaged to create a smoothed orebody model.

In performing these analyses, several key points are followed :

- For the different analyses only the parameters that are obtainable from the Neves grade model are used.
- For all analyses a single horizontal layer from the Neves orebody model will be used
- All 100 simulations available are taken into account. Where less simulations are required simulations will be randomly drawn from the simulations pool.

6.3. KPPI's

Several performance indicators are used to investigate the potential of stochastic optimization. The indicators, or KPPI's (Key Project Performance Indicators) represent the most important consideration parameters for investors. The indicators used in this research are all obtainable from the current grade model and economic block model by direct calculations. An overview is given in table 6.1.

Table 6.1: Key Project Performance Indicators

KPPI	Analysis
Economic Value [€]	Maximum upside potential / Minimal downside risk analysis, Stochastic risk analysis, Stochastic design optimization
Arsenic Penalty Value [€]	Maximum upside potential / Minimal downside risk analysis
Metal Price (Copper & Zinc) [€]	Maximum upside potential / Minimal downside risk analysis
Average Grade (Copper) [%]	Stochastic risk analysis
Average Grade (Arsenic) [%]	Stochastic risk analysis
Ore Tonnage [Tonnes]	Stochastic risk analysis
Waste Tonnage [Tonnes]	Stochastic risk analysis

6.4. Maximum upside potential / Minimal downside risk optimization

The 'Maximum upside potential / Minimal downside risk' method employs the stochastically simulated orebody models from the Neves-Deposit to quantify grade uncertainty. Based on an approach proposed by Ramazan and Dimitrakopoulos [36] it aims to capture the deposit's upside potential, whilst minimizing the downside risk for the deposit's value. In contrary to traditional methods, which use estimated and smoothed orebody models in combination with complex, input dependent algorithms, it should give an indication of the bandwidth of grade uncertainty. The method should also result in a mine design that, unlike the design obtained through non-stochastic optimization, is both robust towards price metal price fluctuation and, as it incorporates grade uncertainty, is a solution better fit to the actual situation.

The individual steps are as follows:

1. From the Geovariances dataset, 20 simulations are selected randomly and from these simulations the copper, zinc and arsenic grades for a certain depth are exported.
2. For each block within this depth layer its economic value is calculated
3. Based on economic value, for each of the 20 simulations stope geometry optimization is performed using the genetic algorithm optimization model.
4. The 20 stope envelopes are each individually cross-referenced with all simulations. For each design a risk profile is created based on economic value.
5. The three best performing stope geometries are selected for further evaluation
6. The final three stope geometries are tested for varying metal prices.
7. The best performing design is selected based on how robust it is in regard to metal price variation

6.5. Stochastic risk analysis

In the 'Stochastic risk analysis' the concept from the first analysis are extended. However, Instead of searching for a single robust design capturing the grade uncertainty towards economic value several other KPPI's are analysed. For each simulation used here again an individual design is generated. These designs are then used to determine stope inclusion probabilities. Based on the orebody input some zones might feature stoping in all of the generated mine designs while others are specific to a certain simulation and are therefore less likely to be beneficial to the optimum design. Using the various stope inclusion probabilities a risk quantification is done based on metal content, ore tonnage, arsenic content and average ore grade. Using these outcomes risk profiles can be made for the individual KPPI's.

The individual steps are as follows:

1. From the Geovariances dataset, 20 simulations are randomly selected and from these simulations the copper, zinc and arsenic grades for a certain depth are exported
2. For each of the 20 simulations stope geometry optimization is performed based on economic value

3. The different solutions (stope geometry outlines) are layered to create a map of stope probabilities (5% - 100%)
4. For different inclusion probabilities (40%/60%/80%/100%) the KPPI values are calculated for each of the 20 simulations
5. Using this data risk profiles are generated for each KPPI

6.6. Stochastic design optimization

The final analysis, more than the other analyses, aims to find the specific value of stochastic optimization when compared to a more traditional approach. Aside from studying the effect of grade uncertainty on underground stope optimization is also tests the full capabilities of the genetic algorithm. Using all unconditional simulation models from Geovariances a stochastic design optimization is done whereby a single design is optimized based on all simulations at the same time. While running, for each generation all solutions are validated through all designs. Based on their resulting average values (average value for each individual over all simulations) the best individuals are selected for further optimization. In theory this should lead to a better final design as no data is ignored. This in contrast to optimizing using a smoothed E-type orebody model. The results of the stochastic design optimization are compared to 3 E-type models which stem from different amounts of simulations. Furthermore a fictional real case scenario is used to indicate the effect of grade uncertainty on mine design optimization.

The individual steps are as follows:

1. Stope geometry is optimized by stochastically optimizing economic value over all simulations at the same time
2. 3 E-type models are created based on 20, 50 and 100 (all) simulations.
3. The 3 E-type models and the 'real case' model are used for stope geometry optimization based on economic value
4. Resulting values are used to create normal distributions for economic value.
5. The normal distribution properties are compared.

7

Results

7.1. Stochastic analyses

For the data analyses a different set of parameters were used. Based on previous experiences some parameters were changed:

- Due to the larger layer size of the CSV Neves orebody models higher termination criterium values were chosen.
- The stope sizes were changed to a range of between 3-by-3 and 8-by-8 blocks
- The barrier size was kept to 1 block
- The cutting range was increased to 0.95

The full input parameter sets are listed in table 7.1 and table 7.2. Where, during the analyses, parameters were altered it will be specifically mentioned.

Table 7.1: Optimizer model

Parameter	Value
iter_model_max	25.000
no_change_lim	10.000
x_stope_max	8
x_stope_min	3
y_stope_max	8
y_stope_min	3
barrier	1
nrpar	50
crossover_probability	0.90
p_low_lim	0.4 0
cutting_range	0.95
chance_mut	0.20
nr_mut	$round(0.1 * (y_modell/y_stope_avg) * (x_modell/x_stope_avg));$
perc_mut	0.80
min_par_to_mut	2
nr_constr_loops	1
selection_pressure	1

Table 7.2: Economic block model creator

Parameter	Value	Units
depth	700	Meters
nr_sim	100	-
rock density	3.3	Tons/m3
block size	4 * 4 * 4	meters
mining_recovery	0.95	%
arsenic_penalty	2500	€ / 0.1 ppm over limit
processing_cost_zn	-12.79	€ / tonne
processing_cost_cu	-7.27	€ / tonne
mining_and_processing_cutoff_zn	-39.9	€ / tonne
mining_and_processing_cutoff_cu	-42.8	€ / tonne
processing_recovery_zn	0.6498	%
processing_recovery_cu	0.8694	%
metal_price_zn	2584	€/tonne
metal_price_cu	5422	€/tonne

7.1.1. Maximum Upside / Minimum Downside Analysis

The aim of this analysis was to explore stochastic optimization under economic variability. For that purpose 20 individual design optimizations were run on 20 different random unconditional simulations. Each design was then tested on each of the 20 simulations resulting in a total of 400 individual values (20 for each simulation). In figure 7.1 the performance of each individual stope design (stope envelope) over the different simulations is displayed.

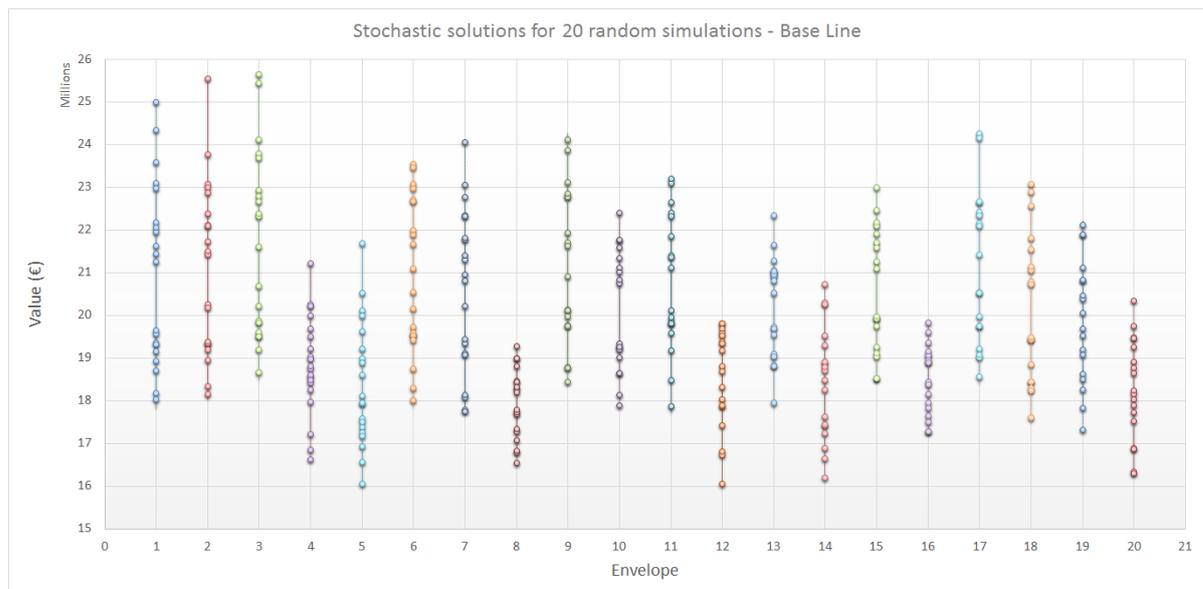


Figure 7.1: Stochastic solutions for 20 random simulations

7.1.2. Base-case

In the graph solution values can be seen ranging from 16 to roughly 26 million euros. The first three designs seem to have larger data ranges relative to the other ranges, whereas most of the other 17 solutions display similar, smaller, solution ranges. Exceptions are solution envelopes 8 and 16.

From these 20 potential stope envelopes the 5 best performing were selected to do a cost analysis with the aim to find the most robust model. The 5 designs selected; envelopes 2, 3, 9, 15 and 17, were selected mainly on their downside risk. Apart from having good baseline values they also have relatively better upside potential than the average upside potential. Envelope 3 is a clear standout as it has the lowest downside risk and highest upside potential.

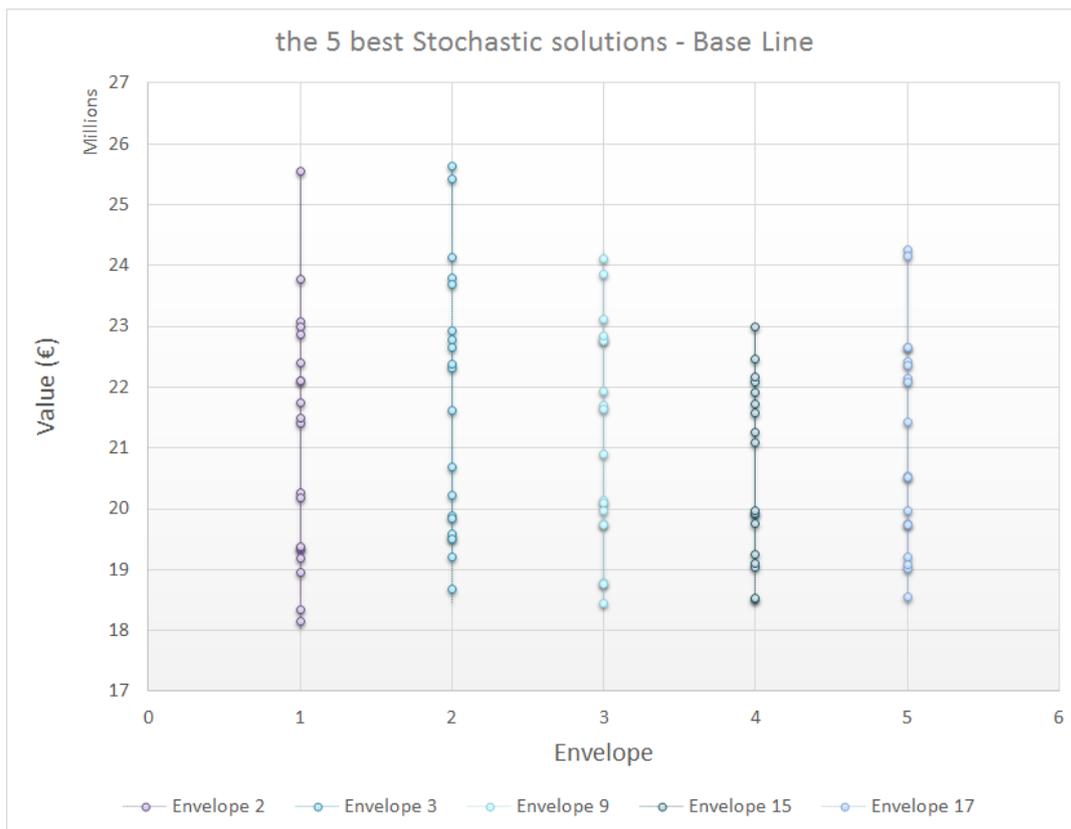


Figure 7.2: Stochastic solutions for 5 best random simulations

Figure 7.2 shows the solution ranges for the 5 selected stope envelopes only. This figure indicates the benchmark case for the selected solutions to which, through the sensitivity analyses, the various scenarios will be compared.

7.1.3. Price-sensitivity analysis

The five selected best performing stope designs were then tested for price fluctuation. Both copper and Zinc prices were increased and decreased with respectively 25 and 50 percent. Figure ?? and 7.4 show scenarios where the metal price was increased, Figure ?? and Figure 7.6 represent decreased metal values. The changed parameters are shown in table 7.3.

Table 7.3: Changed parameters

Change factor	-50%	-25%	No Change	+25%	+50%
Copper price (€)	2711	4066,5	5422	6777,5	8133
Zinc price (€)	1292	1938	2584	3230	3876
Change Factor	No Change	x2	x5		
Arsenic penalty (€/0.1 ppm o.l.)	2500	5000	12500		

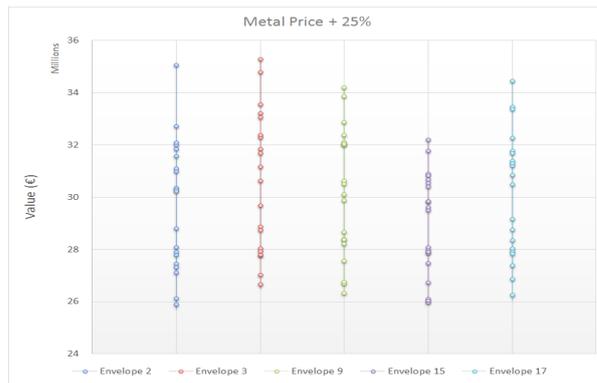


Figure 7.3: Metal price +25%

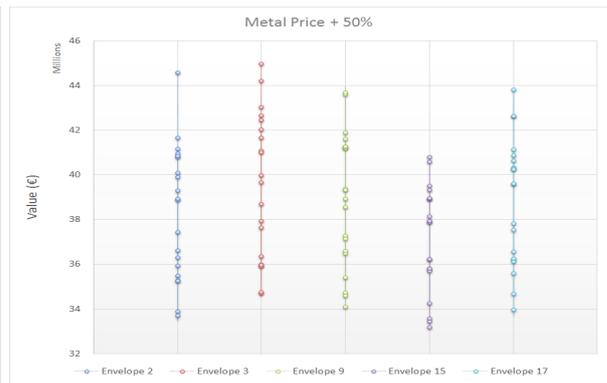


Figure 7.4: Metal price +50%

With the metal prices increased the possible solution values have gone up as well. Where metal prices were increased with 25% the values range between 26 and 35 million roughly, and where a 50% increase was considered potential values are between 33 and 45 million. For the 25% metal price increase this means the lowest possible solution for all envelopes is now higher than their own maximum upside potential with current - benchmark- metal prices.

Envelope 3 outperforms all other selected envelopes both in downside risk and upside potential.



Figure 7.5: Metal price -25%

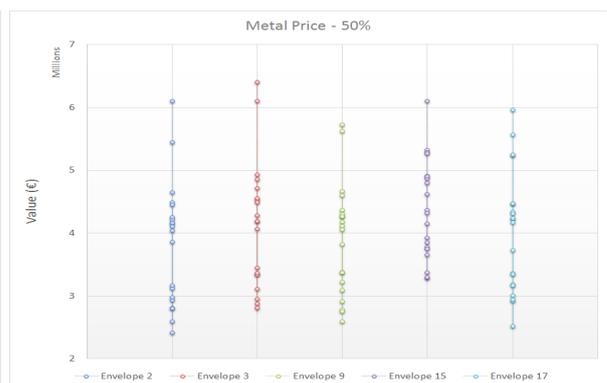


Figure 7.6: Metal price -50%

Decreasing the metal price by 25% has the same effect as a 25% increase in the sense that the highest values recorded for this scenario are lower than the lowest recorded values for the baseline case. Observed is also that envelope 2, 3, and 17 perform very similar, where previously more deviations were seen between 2 and 3 on one side and 17 on the other. Deviation partly returns when the metal prices are further reduced (-50%). Where envelope 15 has underperformed in upside potential in all other cases, with the lowest upside potential recorded, it now clearly outperforms envelope 9 and 17 on both upside potential and downside risk. It has by a margin the lowest downside risk of all selected envelopes. In the worst metal price scenario, value ranges between 2.5 and 6.5 million euros. Individual solution deviation remains similar to the other metal price scenarios with exception of envelope 15. Where with metal prices decreased 25% stope layout 15 was already showing relative improvement, it improved further. Not only does it have by far the best downside risk scenario, in upside risk it now outperforms envelope 9 and 17 where previously this was not the case.

7.1.4. Penalty value impact analysis

In addition to monitoring the effect of metal price fluctuation on individual stochastic design solutions the effect of fluctuations in arsenic penalty were studied. Figures 7.7 and 7.9 show penalty cost factor increases of 2 and 5.

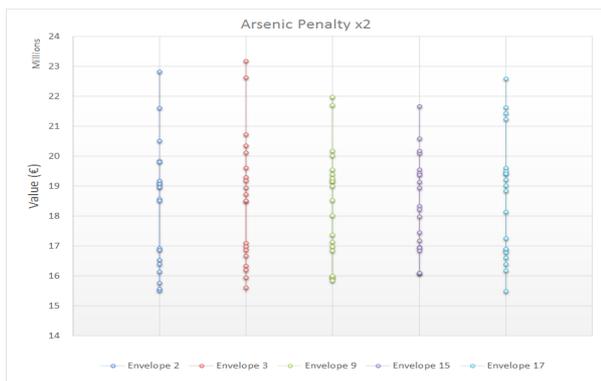


Figure 7.7: Arsenic penalty x2

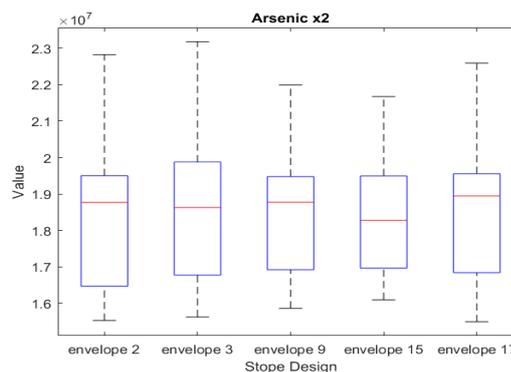


Figure 7.8: Boxplot arsenic penalty x2

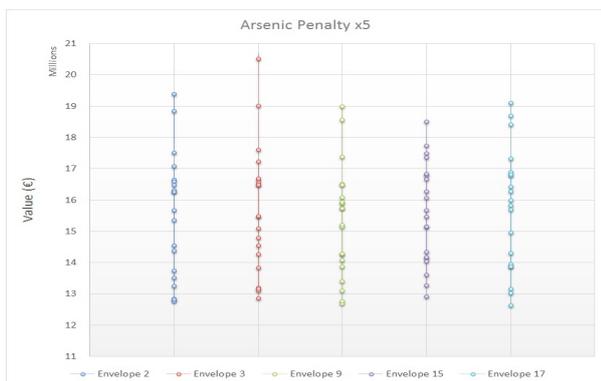


Figure 7.9: Arsenic penalty x5

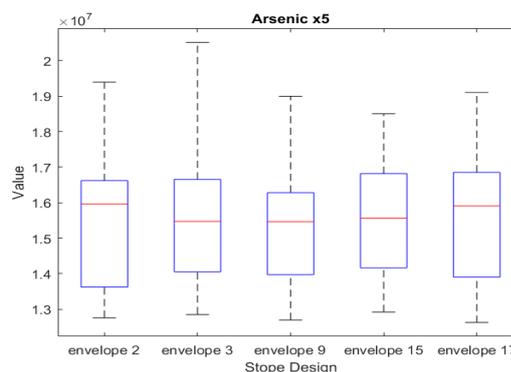


Figure 7.10: Boxplot arsenic penalty x5

When the arsenic penalty is doubled the economic range of solutions is between 15 and 24 million euros. This is a minor decrease (+/- 2M) when compared to the base-line solutions. As in previous analyses, envelope 3 comes out as the best solution on upside potential. On downside risk however, it is envelope 15 that shows best performance. The worst solution is envelope 2 as it has most downside risk and most bad solutions situated in the lower ranges. Envelope 15 has the lowest upside potential. Apart from envelope 4 all solutions remain similar in range.

If the arsenic penalty is further increased few notable changes are monitored in both the distribution range of the envelope solutions and the relation between the different envelopes. Economic potential decreases and values range between approx. 12 and 21 million. An average loss of 3 million per solution compared to the base-line case.

When looking at the boxplots (Figure 7.8 and Figure 7.10) corresponding to the graphs 7.7 and 7.9, we can have a better look at the inter quartile ranges (IQR). The IQR is the range between the 1st and 3rd quartile and corresponds to a certainty of 50%.

Where the lowest non outlier whiskers shows that with higher arsenic penalty values, the variation between the different designs becomes less, the IQR boxes also show smoothening. Envelope 15, having lowest downside risk, surprisingly also shows a relative increase in upside potential. This is not visible in the normal graphs, but when looking at the boxes only it shows highest IQR upside potential from all scenarios.

7.2. Stochastic risk analysis

The stochastic risk analysis is a continuation of the minimum downside/maximum upside potential analysis. The 20 random unconditional simulations and stope envelopes used in the first analysis were also used for the stochastic risk analysis.

Using the different stope envelopes a maps for stope inclusion probabilities are created. Based on the different probabilistic stope inclusion filters risk profiles are created for several Key Project Performance Indicators (KPPI's). The KPPI's studied are: average copper grade, average arsenic grade, ore tonnage, waste

tonnage and economic value.

7.2.1. Stope inclusion probabilities

Figure 7.11 contains a stope inclusion probability map. The inclusion probabilities are given in percentages. Higher percentages indicate blocks or stopes mined in a large portion of the stope envelopes. Small stope occurrence probabilities indicate stopes were placed in that location only in few envelopes.

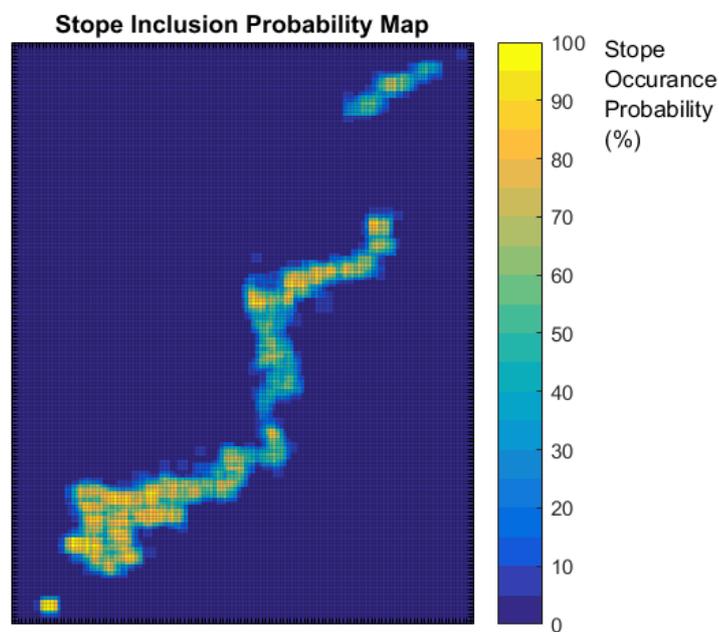


Figure 7.11: Stope inclusion probability map

In general stopes have been distributed well over the positive economic zones. Map locations associated with high arsenic contents have been largely avoided with either stope probabilities of zero, or small percentages. Most low occurrence probability zones appear on the edges of the ore zones. On its own these blocks might not be profitable but are remnants of larger economically positive zones. few stopes seem to have been placed outside of profitable zones at all. In most cases these have probabilities below 10 percent. Some overlap between these outliers occurs.

Figure 7.12 depicts different stope inclusion probability filters. They indicate stope inclusions above a certain inclusion percentage only. Maps of 20, 40, 60 and 80 percent stope inclusion percentages are shown.

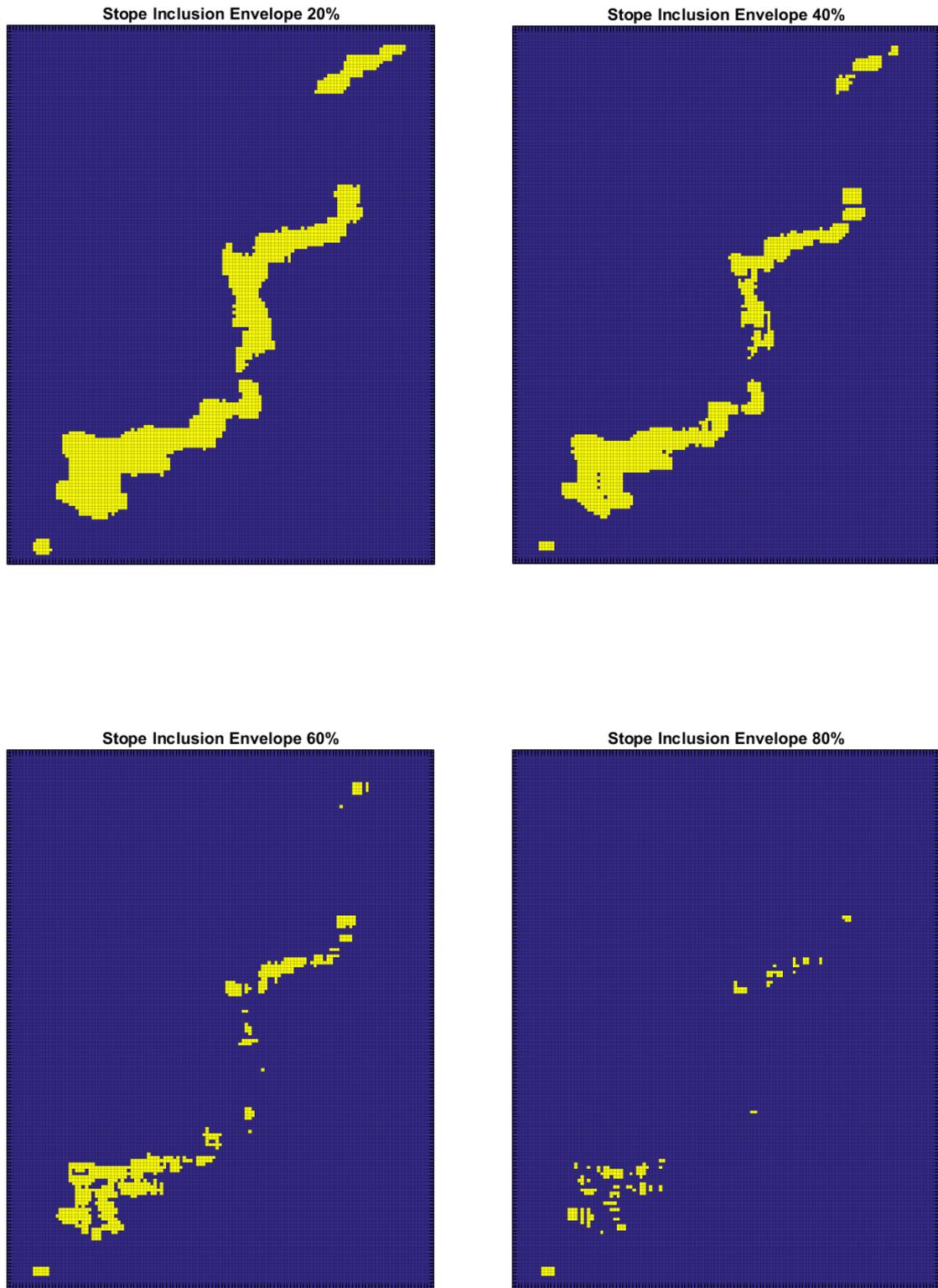


Figure 7.12: Stope inclusion probability filters

These maps show clearly the decreasing amount of stopes for higher inclusion percentages. Where the 20% inclusion map shows nearly all economic zones being mined. Where the contours of the copper grade model are not followed in most cases copper grades are low. Furthermore with a 20% inclusion probability already all outliers have disappeared.

With 40% inclusion probability more zones of low economic potential appear un-stoped. The main orebody zones (bottom, middle and upper isolated zones) are all still covered with one or more stopes.

For the 60% stope inclusion filter the amount of stopes has further declined. The stopes left are all in high grade zones. All three orebody zones are still covered with one or more stopes.

The 80% filter map shows a limited amount of stopes. The top isolated orebody zone is no longer stoped. The stopping areas left are remnants of stopes that have been more extensive for the lower inclusion probabilities.

7.2.2. Risk Profile analyses

7.2.2.1. Economic value

Figure 7.13 shows the resulting economic value for the different stope inclusion probability filters. The economic value for each filter was calculated based on the economic block models for all 20 unconditional simulations.

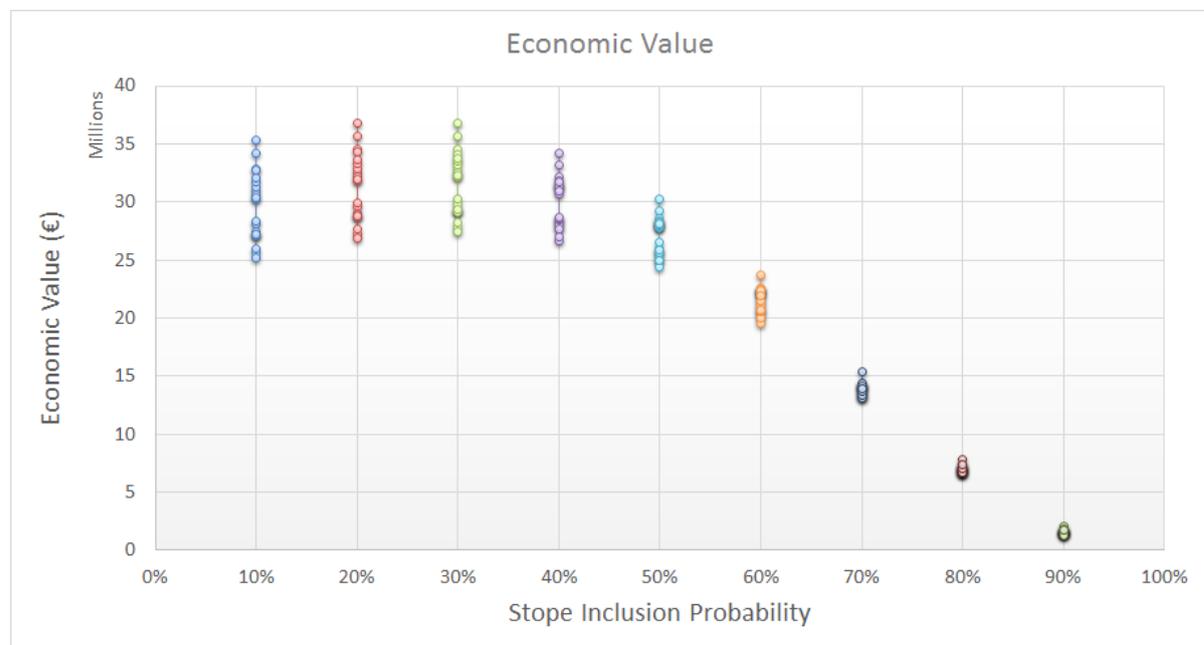


Figure 7.13: Economic value

The different stope inclusion probability scenarios form a nice curve downwards as the inclusion probability increases. The first three scenarios (10, 20 and 30 percent) however appear to show an upward trend in the same range of solution values with spread not declining noticeably. From 30% probability on, the lower the probability, the smaller the solution range. This trend continues all through the graph. For the lower probability designs profits range between 25 and 37 million euros. In the most conservative scenario profits drop to less than a million euros.

7.2.2.2. Ore tonnage

Figure 7.14 shows the ore tonnages recoverable for the different stope inclusion probability filters.

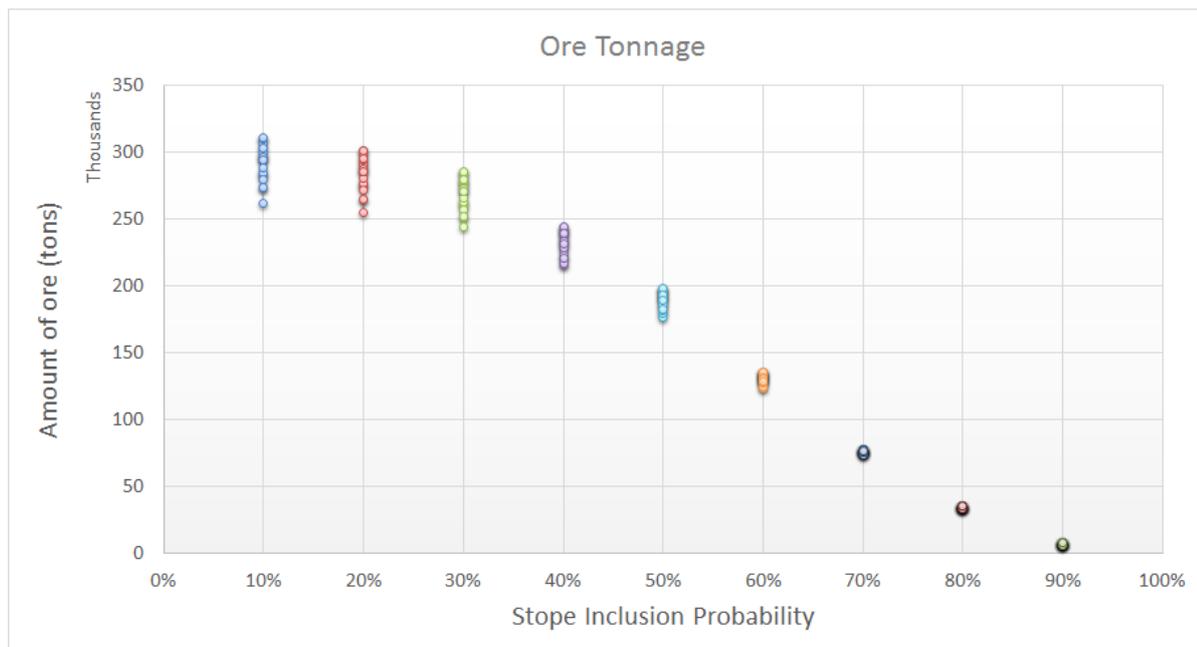


Figure 7.14: Ore tonnage

Like in the economic value graph, the ore tonnage curve shows a downward trend with decreasing value range as more conservative scenarios are calculated. Unlike the Economic value graph however this also includes the low inclusion probability scenarios. In the least conservative scenario prices range around 280.000 tonnes. For high probability models from 70% on, tonnages drop below 25.000 tonnes of ore to be recovered.

7.2.2.3. Waste tonnage

Figure 7.15 shows the waste tonnages recovered for the different stope inclusion probability maps.

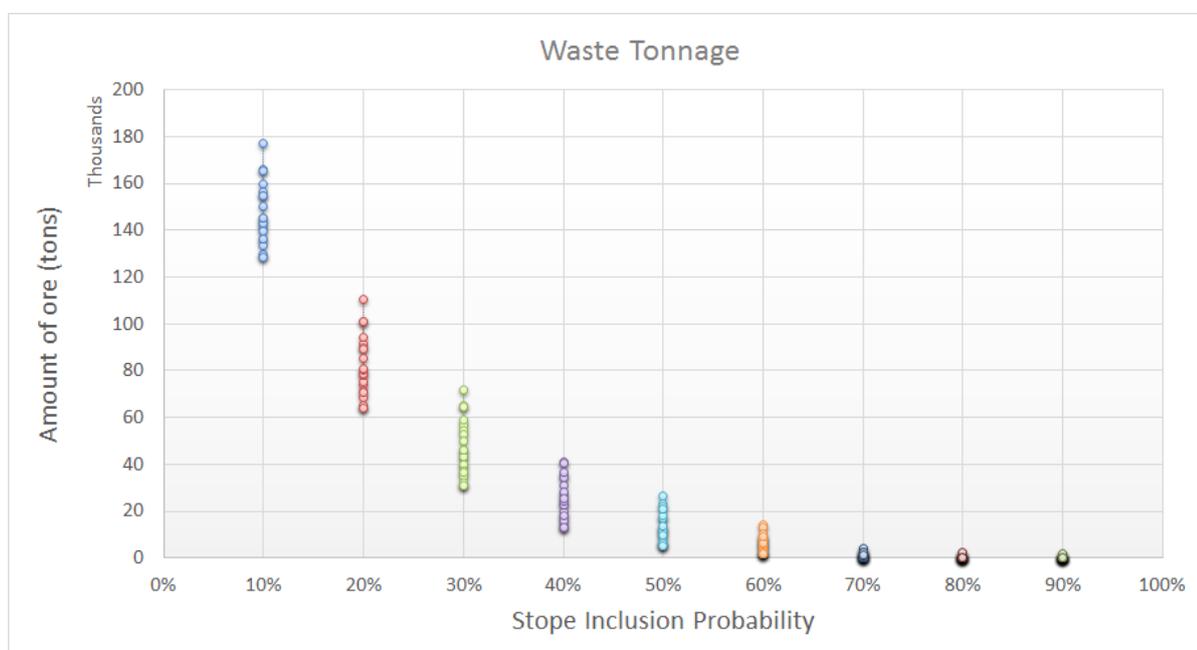


Figure 7.15: Waste tonnage

The figure shows a convex curve downwards. Waste tonnages range between 180.000 and 130.000 tonnes in the lowest probability scenario and curve towards 0 in the high probability scenarios. Waste-less solutions

are present from the 60% probability envelope on with decreasing upside waste as the probabilities increase. The worst case for 20% stope inclusion includes less waste than the most positive scenario if the 10% inclusion probability scenario

7.2.2.4. Average ore grade

Figure 7.16 shows the resulting average ore grade for the different stope inclusion probability filters. For this the copper grades have been taken into account as no profitable zinc ore is present on the 700 meter level.

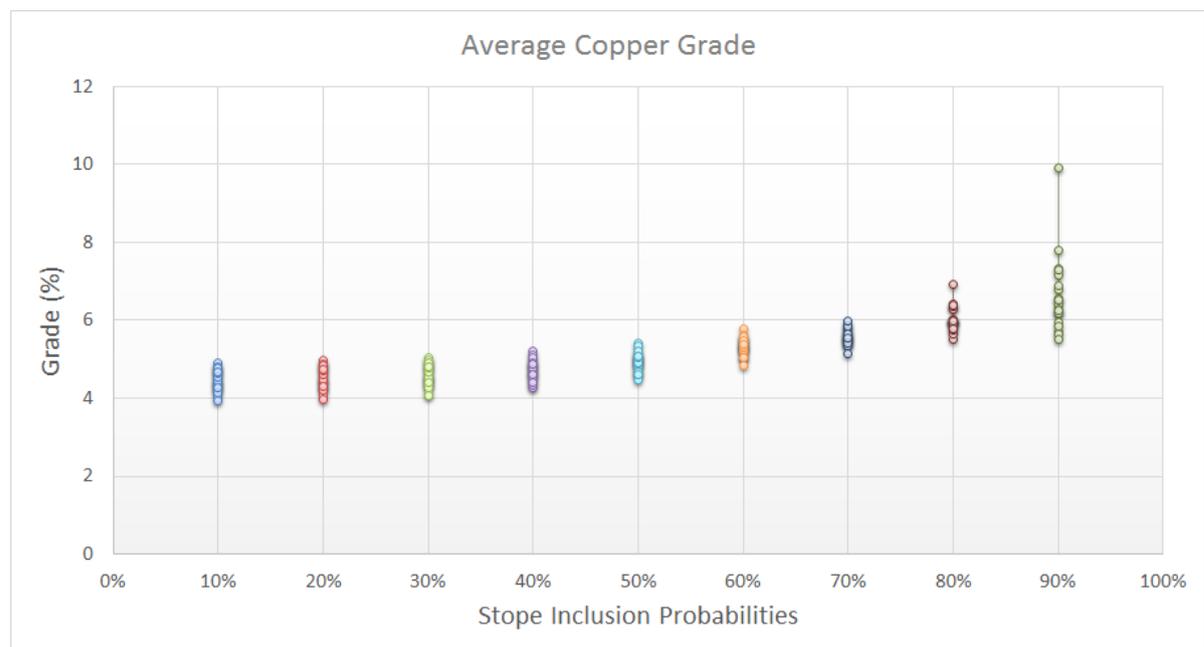


Figure 7.16: Average copper grade

Average ore grades are lowest with low inclusion probability. They range between 4% and 5% until inclusion probabilities of 40%, around 5% for inclusion probabilities of 50% and 60%. After 70% range increases dramatically upward with average grades up to 10% found for the 90% inclusion filter. In these scenarios however it seems freak cases are included.

7.2.2.5. Average arsenic content

Figure 7.17 shows the resulting average arsenic grade for the different stope inclusion probability filters.

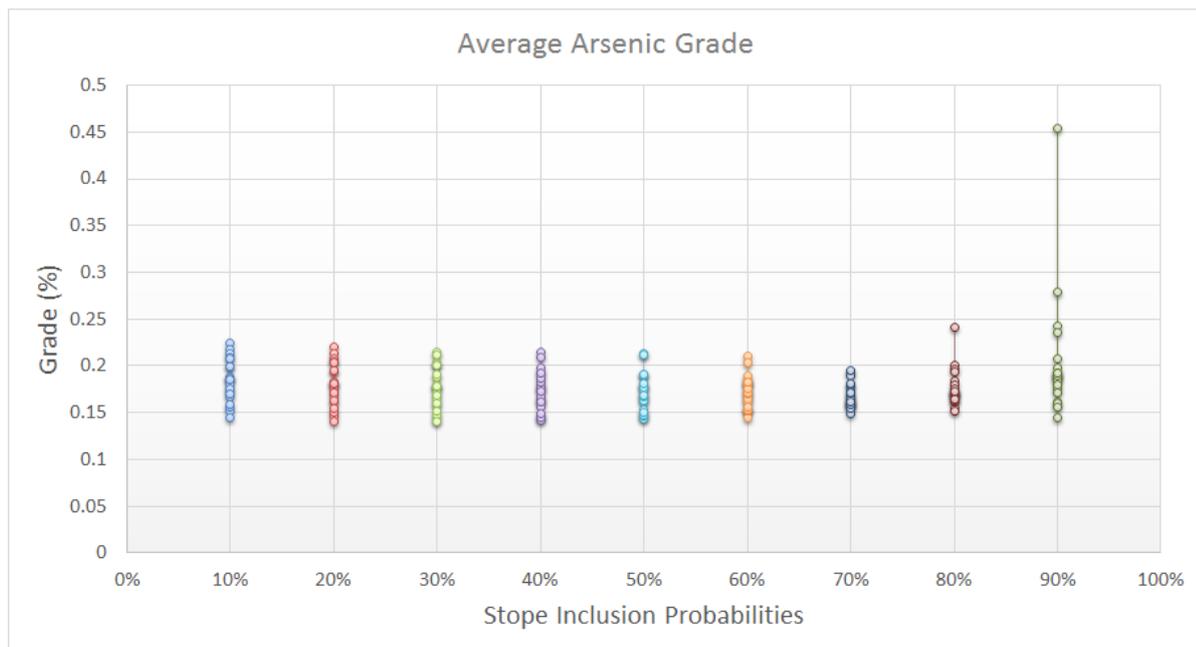


Figure 7.17: Average arsenic grade

Average arsenic grades appear to be constant all stope inclusion probability scenarios. Average grades range between 0.13 and 0.24 percent with for the last scenarios a strong increase in range. As observed in the average copper grade graph, the high probability scenario's (70 – 90%) each include one model design having mayor influence over average grade range.

7.3. Stochastic Design Optimization

For the stochastic design optimization long optimization runs were done using both the stochastic model function, where all 100 simulations are taken into account while running, and an E-type model. The E-type model was created by averaging 100 the random simulations. In the case of averaged layer, based on the optimal solution individual solutions are calculated for all separate economic simulation models. With this data several histograms and normal distributions were plotted.

In an attempt to increase the model accuracy and ensure good optimization performance the termination criterion was changed by increasing maximum number of iterations from 25.000 to 50.000.

Figure 7.18 includes histograms for the four individual cases run. On these histograms normal distribution curves have been fit. As their shape is influenced by the size of the histograms on which they are fit, they can only be used as spread indicator.

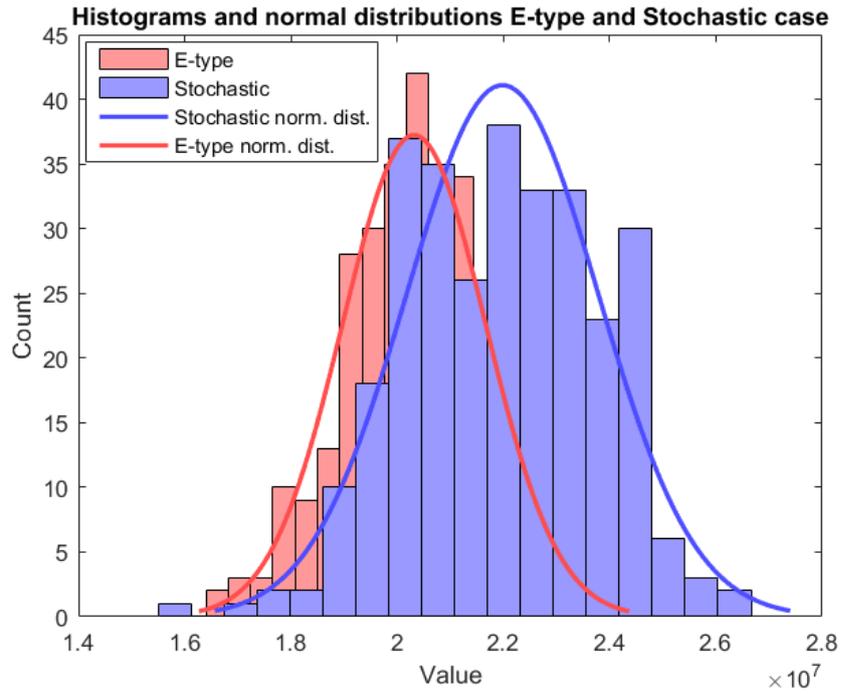


Figure 7.18: Histograms and normal distribution for all cases

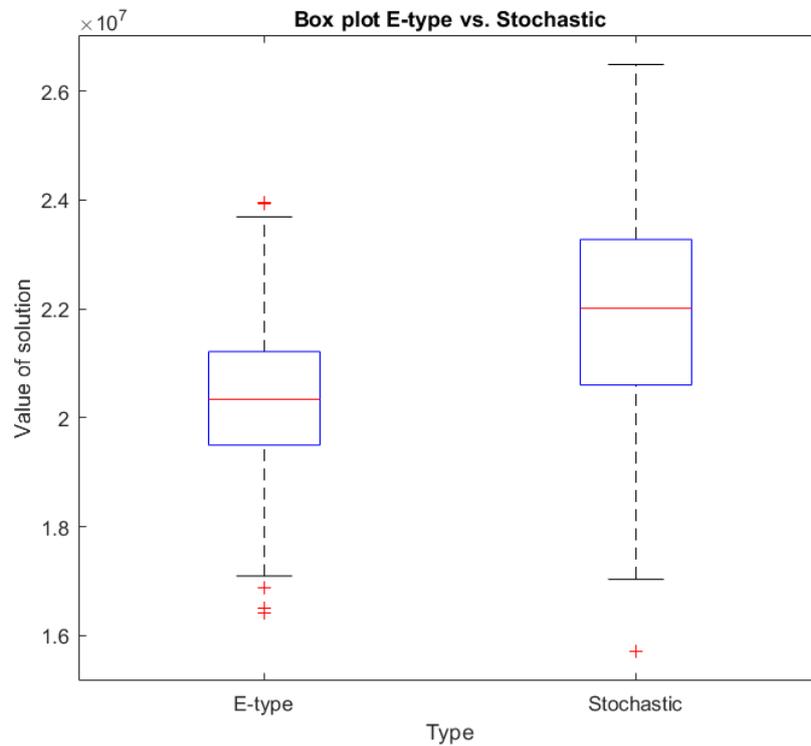


Figure 7.19: Boxplot analysis 3

The two histograms show noticeable differences. It can be clearly seen that the histogram of Stochastic data shows less solutions in the lower value range and its worst results are of higher value than those of the E-type data histogram. The stochastic histogram had a wider base as can be seen well from figure 7.19, show-

ing a box plot of the same data.

The box plot shows that the 25th percentile of the stochastic scenario depicts a higher value than the median of the e-type case. Where downside risk, the lower solutions, are equal, the stochastic results show a much higher upside benefit (2.4E7 vs. almost 2.7E7), and over all better solutions.

In Figure 7.20 the normal distributions for the two model solutions are displayed. Instead of deriving them from the histogram curves they were plotted from the calculated mean and standard deviation of the individual solutions. This results in their shapes no longer being dependent of the histogram spread, but instead being normalized.

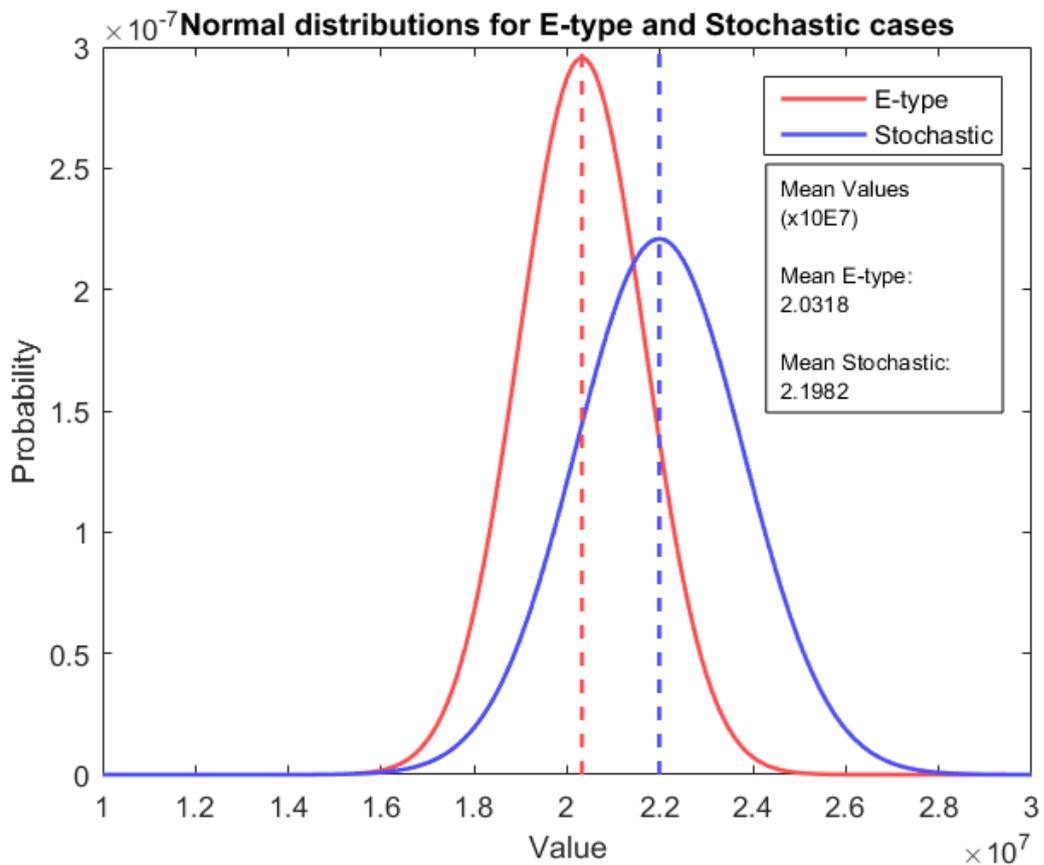


Figure 7.20: Normal distributions

Unlike the boxplot, the normal distribution is centralized, with equal spacing between the 25th and 75th percentile and the mean. As already noted in the boxplots, it can be seen here that the E-type distribution has a narrower body indicating a smaller spread of solution values. The mean value of the stochastic distribution is higher than that of the E-type distribution. The mean of the stochastic distribution being 2.2E7, the mean of the E-type solution 2.0E7.



Discussion of results

8.1. Stochastic analyses discussion

8.1.1. Minimum Downside Risk / Maximum Upside Potential Analysis

The 'Maximum upside potential / Minimal downside risk' method employs the stochastically simulated ore-body models from the Neves-Deposit to quantify grade uncertainty. Based on an approach proposed by Ramazan and Dimitrakopoulos [36] it was executed in an adjusted form with the aim to capture the deposit's upside potential, whilst minimizing the downside risk for the deposit's value.

In selecting the top 5 solutions from the 20 stope designs generated through the genetic algorithm optimizer, only downside risk was taken into account. Partially upside potential was discarded as models with low downside risk in general had good upside potential. Also, if upside potential would be taken into account this would include averaging between downside risk and upside potential. In such a scenario models with larger solution variation would benefit disproportionately as their downside risks are diluted with good upside potential. Stope envelope 2 could serve as an example as it has the highest upside potential from all models. However, this is only one solution where other equally probable solution scenarios are substantially lower in value.

From a business perspective it is more important to reduce risk than to have a large, but risky upside potential. Especially as all models in the current scenarios are of positive value this risk would not be acceptable.

In the analysis 20 simulations were taken into account out of the hundred available. This was deemed enough as the simulations used were generated unconditionally. Generating more stope layout solutions was not expected to deliver substantially better solutions. Especially as the solution ranges of the modelled individual stope envelopes are relatively close, increasing the pool size would most probably increase the computational time more than it would benefit the final solution.

8.1.1.1. Metal price sensitivity analysis

In the Metal price sensitivity analysis the upside potential and downside risk of the five selected stope envelopes was studied. Price fluctuations taken into account were: 25 and 50 percent metal price increase and decrease and multiplying the penalty for arsenic with a factor of two or 5.

When looking at metal price decrease, the range of solutions decrease as a whole for all stope envelopes. However, where, with metal price increase, the individual ranking of solutions is not changed, for lower metal prices the order of solutions is changed significantly.

If we take into account envelopes 15 and 17 it can be seen that solution envelope 17 performs better than envelope 15 for both the base-case and metal price increase scenarios for both upside potential and downside risk. However, for a metal price reduction of 25% the upside potential and mean for envelope 17 might still be higher, but on downside risk envelope 15 now performs better. With a metal price decrease of 50% envelope 17 performs worse than envelope 15 as a whole. Both on upside potential than on downside risk.

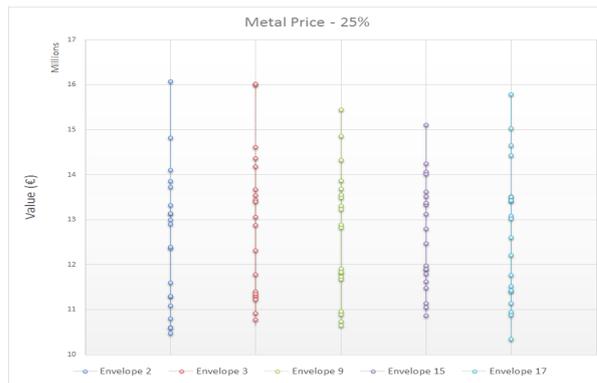


Figure 8.1: Metal price –25%

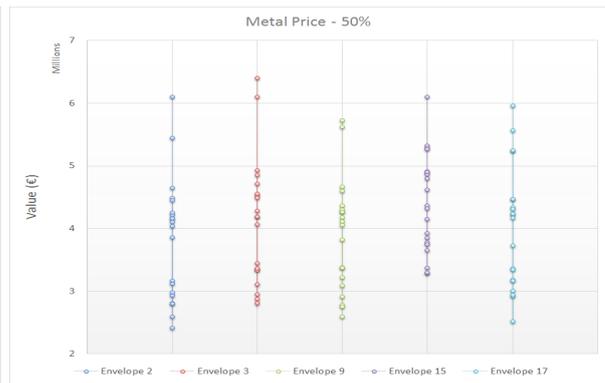


Figure 8.2: Metal price –50%

The different behaviour of the selected stope envelopes towards price decrease can be explained through the constant cost of waste mining. Where, when metal prices go up, the orebody as a whole becomes more profitable linearly, when prices decrease, zones with low metal content will become waste instead of decreasing in value more. An explanation for the un-linear value decrease of envelope 15 compared to the other stope envelopes could be because it has relatively more stopes places in high grade zones or zones with lower arsenic content. In that case envelope 15 has a lower ability to decrease in value linearly before the fixed value of waste prevents further loss and leads to un-linear behaviour.

In all metal price scenarios however it is stope design 3 that performs the best on upside potential, and, when looking at the 25th percentile, also on downside risk. Compared to the other models it behaves stable and its position relative to the other models does not change in any of the price scenario's. When looking at the range between the 25th and 75th percentile it displays most upside potential and lowest downside risk in all cases.

8.1.1.2. Arsenic penalty sensitivity analysis

The increase of arsenic does have a negative influence on the solution values, with all profits decreasing. However, unlike in metal price decrease scenario's it does not affect the relative order of solutions, nor the relative spread in solutions. This would indicate the increase of arsenic penalty has little effect on the appearance of waste in high grade ore zones. Where ore grades are relatively low it would result in the transition of economic zones to waste. However, as in good economic scenarios these contributed little to the total solution value, the transition of low profit zones into waste has no visible impact on the solution upside potential and downside risk ranking, as no change in solution range or relative envelope can be seen.

8.1.1.3. Concusive statements

The aim of this method has been to select the most robust scenario, capable of dealing with market fluctuations. For doing so, four characteristics have to be taken into account: downside risk, upside potential, the median solution and the range of individual solutions. When selecting a stope layout based on upside potential envelope 3 is most suitable. Through the different scenarios it shows the most stable mean value, on average the lowest downside risk and has good upside potential. However, the envelope has a wide range of solutions in all scenarios. This indicates a large amount of uncertainty on the final output of the mine. If uncertainty is to be mitigated envelope 15 would be the most suitable design. Even though it does not provide a large upside potential, it has a small solution range. Where it does not benefit as much from increased metal prices as the other models, it is best in mitigating downside risk when metal prices drop or arsenic penalties increase. When trying to mitigate risk this would therefore be the best solution.

8.1.2. Stochastic Risk Analysis

The stochastic risk analysis can be seen as an extension to the minimum risk / maximum upside potential analysis as the same stope envelopes, and simulations on which they were based, have been used. The stope envelopes were layered to create a map of probability of stope occurrence. The influence of different stope inclusion probabilities has been examined for a number of Key Project Performance Indicators.

Looking at the stope inclusion probability map few low probability stope areas can be seen that lie outside

the orebody. As they do not contribute to the final value, they should theoretically not be there. It indicates that in some cases the optimization process was aborted before an optimum was reached. It indicates that the iteration limit value used was too low. As these 'out of orebody' stopes have minimum probabilities (5%) we can assume they are not regular occurrences.

Another explanation for some low probability stopes lying on the very edge of the orebody model is that even though most of the stope is out of the orebody contributing waste, the small portion of the stope that overlies the orebody captures such a high zone that the benefit of including these blocks overshadows the negative contribution of the waste blocks.

When considering the individual probability layers it can be seen that the most uncertain stopes (those included only in low probability models) are all situated in the low grade orebody zones. This would indicate first of all that optimization has indeed succeeded and contradicts random search practices. Also this indicates a higher optimization stability for higher grade orebody zones than for low grade zones.

A rapid decrease in stope inclusion is noted. Comparing the 60% stope probability map with that of 20% inclusion probability shows that in the latter twice as much blocks are mined. When stope inclusion probability is further increased, few blocks are left to be mined.

Where other researches have shown that for higher probabilities the number of stopes included rapidly decreases, [37] these results show a more extreme decline. The reason for this is that in the optimized stope layouts pillar barriers have been taken into account. These pillars are required to surround the plotted stopes completely. As depending on the variation in the orebody simulations, stopes are not always placed in exactly the same location, this amplifies the effect of pillars on the stope inclusion probabilities. The inclusion of pillars in the optimization algorithm therefore 'dilutes' the high probability stope inclusion zones. What is now indicated to have a 60 or 40% inclusion probability would most likely have a substantially higher probability when no pillars are taken into account.

8.1.2.1. Copper grade risk

Instead of evaluating average ore grades it was decided to take into account the copper grades only. Even though the zinc grades were taken into account whilst building the economic block model, the zinc grades were too low to produce positive economic block values. As a result the economic block values were purely copper driven.

Evaluating the average copper grade risk profile, a smooth upward trend can be seen indicating that as less probable stopes are taken out, higher grade zones are mined. This as they are more certain to be of economic interest in the varying simulated scenarios.

The estimated average copper grades at the 80 and 90% probabilities showcase great spread where spread for other probabilities was constant. Their range is stretched upwards by one model each time indicating a simulated orebody scenario that differs much from the other simulations.

8.1.2.2. Arsenic risk

Based on the graph for average arsenic grades it can be concluded that arsenic, although of influence on overall profitability, does not play a large role in the stope optimization process. Although arsenic grades are far from uniform over the orebody model, they are usually connected to the copper grades. As the mineral containing copper also contains arsenic, zones with copper mineral abundance usually also show higher arsenic values. However, the lack of a significant upward trend as was seen with the average copper grades seems to indicate there is no direct linear relation between the two.

The outliers in the high probability models that could suggest some upward trend give too much of a distorted image to prove a relationship between average copper grades and average arsenic grades for this case.

8.1.2.3. Tonnage risk

The ore tonnage curve shows a fluent decrease in material mined as less probable stopes are left out. This is logical, as with more certain models only high grade material is mined and therefore less stopes are present. The flattening out of the curves in the very low and very high probability regions (10/20% & 80/90%) is the result of there being less of a difference in how much stopes are mined. For both the 10 and 20% probability models almost all ore is mined, for the 80 and 90% models very few stopes are mined.

8.1.2.4. Waste Tonnage risk

The ore tonnage curve shows a convex decrease in waste mined as less probable stopes are left out. This is logical, as with more certain models only high grade material is mined and therefore less stopes are present. Furthermore the stopes left are mainly present in the high grade zones. Waste however remains present in small quantities also for the higher stope inclusion probability envelopes. This is due to some high grade zones bordering waste zones directly. In order to profit from those economic zones whilst still placing valid stopes, some stopes overlap waste slightly. The flattening out of the curves in the very high probability regions (70/80/90%) is the result of there being less of a difference in how much stopes are mined. For both the 80 and 90% probability models very few stopes are mined.

8.1.2.5. Economic value

The economic value probability model clearly indicates a decrease in value for the high probability stope layouts. Where mining high grades gives the certainty of making a profit, it does not assure the highest possible profit as that depends on the metal price fluctuations. In the lower probability regions we can see that the stope inclusion model with the lowest probability is potentially less profitable than the 20% stope inclusion probability model. One of the things contributing to this is that, as mentioned prior, in the low probability zones several outlier stopes are placed outside the orebody. Another reason is that, where low probability stope models can work well with some of the simulated economic block models, in less favourable scenarios they are more influenced by the negative impact as these areas are not avoided in low probability models.

8.1.2.6. Conclusive Statements

If, based on these stochastic KPPI risk analyses a mine design is to be chosen it would most likely be the 30, 40 or 50% inclusion probability model. The choice for either of these two would be based purely on economic value as this is the main driver of the operation. Moving down the stope inclusion probabilities from high to low, we can see that up to the 60% model, each model of higher inclusion probability shows lower upside potential than the next model's downside risk. Choosing a probability of over 60% would therefore be not feasible from a value perspective. The 30% model shows more upside potential and less downside potential than the 40 and 50% models, however here the choice is not as easy anymore as the 30% probability model is not better in all cases than the 40% model. The goes for the 40% model compared to the 50% model although the difference between models is bigger. Comparing the 30% and 40% model there are few large differences in upside potential and downside risk, whereas the 10% and 20% inclusion probability models clearly shows more downside risk for economic value.

The other KPPI's do not have an impact on profitability itself but do influence the processing scheme required. If the processing plant does not deal with the output of the mine efficiently this can be a source of economic loss. The decision whether to pick the 30, 40 or 50% stope inclusion probability model can therefore be made considering processing preferences and plant requirements.

The average arsenic content is stable through all inclusion probability scenarios and is therefore not decisive in picking either of the three models. The range indicated however can be used in designing the optimal arsenic removal process.

A model with higher average ore grades is preferred when blending is not considered as it requires less concentrating to get the right concentrate grade. If blending is part of the process, the model with either the lower or higher average grades can be picked. The range of average grades will help as blending can be modelled more accurately.

The present mill capacity also plays a role in selecting the optimal model. If large capacity is present the 30% probability model should be chosen as it has slightly more upside potential. In capacity is less and could be critical, choosing the 40 or 50% would be better depending on the processing

8.1.2.7. Final Remark

In the risk maps pillars are not included as all models were layered. To come to a feasible solution, optimal pillars should be added to the stope inclusion probability models. Without them present the solution values are not accurate enough for production scheduling.

8.1.3. Stochastic Design Optimization

The results from the stochastic runs show significant differences between optimization results. The differences are certainly large enough to state that the stochastic model is able to achieve better optimization results. As both the stochastic model and the E-type models average results are likely to be similar in linear cases. It is only in non-linear cases that stochastic optimization leads to better results due to averaging being performed in a later stage of the optimization, where during individual optimization all variance is taken into account.

The active capping of block values in the case of waste, and the non-allowance of negative value blocks as ore has contributed greatly to the non-linearity of the model.

Non-linearity was also shown in the minimal risk / maximal upside potential analysis. With current metal prices and economic parameters therefore it can be expected greater stochastic benefit is to be found. This as it will increase the un-linearity of the model.

Where in the model validation stage small differences between stochastic and E-type runs were found, these are more likely to be the result of optimization differences and can be expected to disappear completely if more cases are analysed, or longer run-times are allowed.

If, through adding additional non-linear constraints, un-linearity was further increased, the model should produce more decisive results in the favour of stochastic optimizing. As the stochastic optimization module has been built in the model effectively, this would only require minor alterations in the constraint module.

The difference in mean value (€1 700 000) found in the stochastic analysis might not seem a massive value increase overall. However, considering that in this analysis only one orebody layer was taken into account stochastic optimization over the whole orebody would most likely result in a value increase of €100 million (ballpark).

Even though stochastic modelling will be more time consuming the traditional – average based – optimization, its potential value gain would far outweigh the extra effort.

8.1.3.1. Conclusive Statements

The optimized stope layouts generated for the stochastic design optimization were of much better quality than those used in the minimal risk / maximal upside potential analysis and stochastic risk analysis. No stopes were found to be outside the ore bearing zones. This strengthens the assumption that the no change limiter value used in the optimization runs for these analyses was too low to reach the optimal stope layout.

8.2. Comparing the Min / Max analysis with the Stochastic analysis

In the previous sections both the minimal downside risk / maximum upside potential analysis and the stochastic design analysis were discussed. Where both methods aim to be beneficial over traditional mine optimization, they both have their up and downsides. Furthermore, where the min / max analysis strives to mitigate downside risk, it has not yet been compared to the mine design algorithm developed for this research.

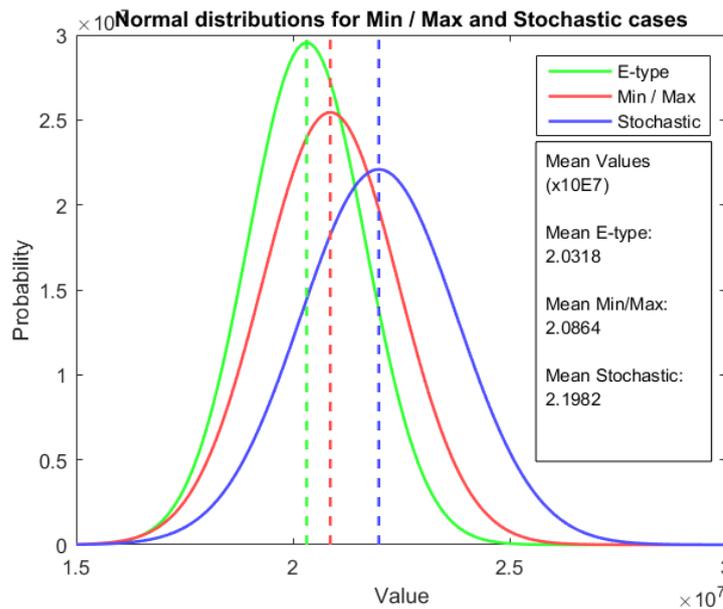


Figure 8.3: Comparison Min/Max analysis with E-type analysis and stochastic analysis

Figure 8.3 compares the results for the different optimization scenario's. The green distribution shows the range and probability of possible outcomes when optimization is done based on a smoothed orebody model. The red line is a distribution based on the hand-picked best-case scenario from the minimal downside risk / maximum upside potential analysis. The higher mean value and the shift of the distribution to the right shows that downside risk has been limited and upside potential has increased.

The blue distribution is resulting from the stochastic mine design optimization using the stope optimizer. Its mean is higher than that of both other displayed scenarios and it shows lowest overall downside risk and upside potential.

In table 8.1 the key statistical values describing the three distributions are listed.

Table 8.1: comparing mean, standard deviation and percentile

	E-type	Min / Max	Stochastic
Mean	$2.0318 * 10^7$	$2.0864 * 10^7$	$2.1982 * 10^7$
Standard Deviation	$1.3499 * 10^6$	$1.5673 * 10^6$	$1.8057 * 10^6$
5th percentile value	$1.7618 * 10^7$	$1.7729 * 10^7$	$1.8371 * 10^7$
95th percentile value	$2.3018 * 10^7$	$2.3998 * 10^7$	$2.5593 * 10^7$
SEM (standard error of the mean)	$1.3499 * 10^6$	$3.5046 * 10^5$	$1.8057 * 10^5$

An increase in standard deviation can be seen for the different scenarios as the mean solution value increases. Where the E-type distribution is represented by a standard deviation of 1.35, the better performing stochastic solution has a standard deviation of 1.81.

This is not the result from the optimization itself, as this would decrease the standard error, as can be seen, but not the standard deviation per se [26]. The increase of standard deviation however is likely to result from the variation in different solutions being different for the three cases. Where an average model was used, standard deviation is smallest. However, in the stochastic solution 100 individual simulations were used increasing the variation in outcome. Even though the solution results are more accurate, as is indicated by the standard error of the mean (SEM), this would explain the increase in standard deviation.

The lesser performance of the minimal downside risk / maximum upside potential analysis when compared to the stochastic design optimization analysis is mainly a result of different objective aims. Where the stochastic analysis takes into account one optimization scenario with fixed parameters, the min / max analysis aims to find the most robust model for changes in metal prices and penalty costs, in addition to the economic

value. In this analysis a solution with a higher mean in the standard scenario can be rejected based on the fact that it shows unfavourable behaviour in other scenarios. The selected stope design (envelope 15) did prove to be most robust in less favourable economic scenarios, it does not have the highest mean and upside potential in the standard case.

Another explanation for the lower optimization performance of the min / max analysis compared to the stochastic design optimization analysis is that the stochastic runs were allowed more iterations. Although all optimization curves indicated that the optimum was approached, in most cases the no change limited termination criterion was not reached.

9

Conclusion

The aim of this research, as formulated in Section 1.4 was:

“to introduce and evaluate and optimization approach for underground stope mining, based on genetic algorithms, taking into account grade uncertainty and to quantify grade uncertainty risk by stochastic analysis.”

This aim was achieved by successfully completing the achievements set out for this research:

- A 2D stope layout optimizer was developed in Matlab based on a binary genetic algorithm.
- The algorithm was converted to allow stochastic input in the form of unconditional orebody simulations or unconditional economic block models.
- A module was created to convert the available unconditional ore-grade simulations into an equal number of unconditional economic block models
- Using this genetic algorithm stope layout optimizer the orebody was evaluated stochastically and the potential benefits of stochastic optimization were investigated by comparing stochastic optimization to optimization of an average case model.

In section 1.5 the research hypothesis was stated:

“Simulation based stochastic optimization of underground mine design will lead to better results than optimization based on customary average grade models”

To answer this hypothesis two research questions were considered:

- How can genetic algorithms be used in underground stope geometry optimization?
- How can stochastic optimization of stope geometries, using genetic algorithms, be used to manage and eliminate uncertainty in underground mine design?

The answer to the first question can be found in the developed model itself. An optimizer was developed based on the basic principles of genetic algorithms in order to find the optimum and most economical stope geometries and locations. Model validation has shown that the algorithm developed is capable of optimizing stope locations by finding the true optimum. The model was extended to allow a stochastic input so that it could be used to answer the second question.

Using unconditional orebody simulations the model was used to perform three different stochastic analyses in order to investigate the capabilities of stochastic optimization.

The Maximum upside potential / Minimum downside risk showed the large variety of solution values when considering the individual unconditional simulations. However by referencing each individual design to 20 different orebody simulations, the range of solution values for each stope layout was visualized and both upside potential and downside risk could be determined. Changing the economic parameters showed non-linear behaviour of the different design solution ranges. The most robust and favourable design could then be selected in order to minimize the potential downside risk and maximize design potential for different economic scenarios.

A stochastic risk analysis was done to show the probability of stopes being mined when considering the individual optimized mine designs. Maps for different stope inclusion probabilities were made and analysed for a number of KPPI's. In general larger variability was seen for the lower probability models indicating a larger amount of uncertainty.

By investigating each solution's upside potential and downside risk, the most economic stope location zones could be identified and grade and penalty element ranges determined. Lastly, The optimizer was used to optimize stope layout while taking into account all unconditional economic block models at once. The results were compared to similarly optimized stope layouts based on commonly used E-type models. This was done to see if better results could be obtained while taking into account grade uncertainty.

The results showed a mean value increase of almost 10% for the stochastic optimization result. Furthermore, Comparing the distribution of possible solutions of the E-type model to the distribution for the stochastic case showed that the stochastic solutions downside risk and upside potential were increased even more.

When the optimization results for the E-type case were compared to the same results for both the minimal downside risk / maximum upside potential analysis and the stochastic design analysis, the latter two both showed improved results over the base case. However, as the min / max analysis aims to find a more robust solution for different economic scenarios, its average optimized value showed less improvement over the base case than the stochastic design analysis. Improvement was hampered by the lack of focus on mean performance and the emphasis on downside risk instead of upside potential.

Based on these stochastic analyses results, which have shown the potential and benefits of stochastic optimization in underground mining the hypotheses stated can be accepted.

Simulation based stochastic optimization of underground mine design will lead to better results than optimization based on customary average grade models.

Recommendations and further work

The model as was developed for this research is not ready to be implemented as is. Further work and research are necessary to utilize the capacity of the model. Several recommendations for further research are listed below.

- The optimizer model used in this research was scripted in Matlab. This due to the versatility of the language, extensive documentation and large database of functions. However, Matlab is not the most efficient and streamlined programming language. Effort could be made to script the model in a different programming language such as Python or C as these have proven to be faster in calculating. Another advantage of re-programming the model to C would be that this language allows for generating an executive file, therefore an installed version of Matlab would no longer be needed.
- The Genetic Algorithm scripted does not separate the two different generations during the cross-over and mutation steps. This means that after individuals have been altered, they overwrite the originals. As good individuals have a higher chance to be selected for cross-over this increases the risk of losing good individuals and thus decreases optimization potential. By adding a parallel generation and merging the two after the alteration steps, faster optimization can be expected.
- The model makes use of horizontal orebody layers. Due to the size of the orebody file it was not possible to load in the full orebody dataset, and using multiple layers would slow down optimization significantly. As the model is now, adding a third dimension would be relatively easy and would be recommended in further research.
- The stochastic risk analysis showed that the pillars included in the model diluted the probability of stopes being included. It would be interesting to compare the analysis with the pillars included (as in this research) with a case where the pillars are not taken into account.
- In order to find out the true benefit of stochastic optimization over conventional mine design optimization a new case could be examined. By comparing the optimization results of that case to a mine design created conventionally, its true benefit can be determined.
- Apart from mine design optimization alone, effort could be made on scheduling the optimized layout to obtain the Net Present Value. Further research could include adding a stope scheduling algorithm to the existing model script.
- Flexible pillar sizes were not taken into account in this optimization algorithm. This generalization makes the model less realistic as for larger stopes in equal geological conditions, thicker pillars would be required to keep them stable. The effect of this generalization on realism was mainly featured in the stochastic risk analyses, where it had a profound effect on the stope inclusion probability envelopes. In future attempts or further work it would be recommended to incorporate flexible pillar sizing in all individual model aspects.
- Development costs were not taken into account in this stope location optimizer. Realistically isolated stopes that are profitable in the current optimizer results, could in fact not be profitable when large development works are required to reach them. A recommendation would be to incorporate a fixed starting location and discount or penalize based on the distance between the start point and potential stope location. This would have to be done in the string evaluation module.

- For the minimal downside risk / maximum upside potential analysis different economic scenarios were investigated and its influence on the stope design performance analysed. This most likely had an effect on the amount of improvement over the E-type optimization. It would be interesting to see if the benefits of stochastic design optimization would also be hampered by taking into account multiple economic scenarios, and how big this influence would be.

Bibliography

- [1] Divide and conquer algorithm. https://en.wikipedia.org/wiki/Divide_and_conquer_algorithm. Accessed:.
- [2] Overhand cut-and-fill mining. <https://www.britannica.com/topic/overhand-cut-and-fill-mining>. Accessed:.
- [3] Horizon 2020 project real-time mining started. <http://www.citg.tudelft.nl/nl/actueel/laatste-nieuws/artikel/detail/horizon-2020-project-real-time-mining-started/>. Accessed:.
- [4] Kriging and cokriging. http://petrowiki.org/Kriging_and_cokriging. Accessed:.
- [5] Mathematical programming. http://glossary.computing.society.informs.org/ver2/mpgwiki/index.php?title=Extra:Mathematical_programming, . Accessed:.
- [6] Mathematical optimization. https://en.wikipedia.org/wiki/Mathematical_optimization, . Accessed:.
- [7] Metaheuristics. <https://classes.soe.ucsc.edu/ism206/Fall10/Lecture12.pdf>, . Accessed:.
- [8] Metaheuristics classification. https://commons.wikimedia.org/wiki/File:Metaheuristics_classification_fr.svg, . Accessed:.
- [9] Mining software history. <http://www.miningsoftware.org/portfolio-items/surpac/>. Accessed: February 2016.
- [10] Vms volcanogenic massive sulphide ore deposits and mineralization. <https://www.911metallurgist.com/blog/vms-volcanogenic-massive-sulphide-deposits-ore-mineralization>. Accessed:.
- [11] South deep gold mine. https://www.goldfields.co.za/reports/rr_2009/tech_south.php. Accessed:.
- [12] Lme copper. <https://www.lme.com/en-gb/metals/non-ferrous/copper/>. Accessed: November 2016.
- [13] Fitness landscape analysis for computational finance. <http://www.turingfinance.com/fitness-landscape-analysis-for-computational-finance/>. Accessed:.
- [14] Genetic algorithms - mutation. https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_mutation.htm. Accessed:.
- [15] Fitness proportionate selection. https://en.wikipedia.org/wiki/Fitness_proportionate_selection. Accessed:.
- [16] <https://www.hindawi.com/journals/ijdmb/2012/716780/fig3/>. Accessed:.
- [17] Model smelter schedule. <http://costs.infomine.com/costdatacenter/smeltingcosts.aspx>. Accessed: November 2016.
- [18] https://www.researchgate.net/figure/254428732_fig4_Fig-4-Example-of-a-two-point-crossover. Accessed:.
- [19] Lme zinc. <https://www.lme.com/metals/non-ferrous/zinc/>. Accessed: November 2016.

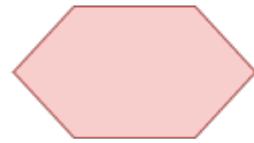
- [20] C. Alford. Optimisation in underground mine design. *Applications of computers and operations research in the minerals industries. Conference, Brisbane, 1995*, pages 213–218, 1995. URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0029481407&partnerID=40&md5=8a422571faed3d37d307f23dd8c8e176>.
- [21] Majid Ataee-Pour. A heuristic algorithm to optimise stope boundaries. 2000.
- [22] Majid Ataee-Pour. Optimisation of stope limits using a heuristic approach. *Mining Technology*, 113(2): 123–128, 2004. ISSN 1474-9009. doi: 10.1179/037178404225004959. URL <http://dx.doi.org/10.1179/037178404225004959>.
- [23] NA Barricelli. Experiments in bionumeric evolution executed by the electronic computer at princeton, nj. *Archives of the Institute for Advanced Study, Princeton, NJ*, 1953.
- [24] J Benndorf and R Dimitrakopoulos. Stochastic long-term production scheduling of iron ore deposits: Integrating joint multi-element geological uncertainty. *Journal of Mining Science*, 49(1):68–81, 2013. ISSN 1062-7391. doi: 10.1134/S1062739149010097.
- [25] O Bertoli, C Bessin, and C Mariz. Virtual asset model of neves-corvo test case - h2020: Real time mining - t8358. Consultation report, Geovariances, March 2016.
- [26] David J Biau. In brief: Standard deviation and standard error. *Clinical Orthopaedics and Related Research*, 469(9):2661–2664, 2011.
- [27] Tobias Blickle and Lothar Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394, 1996.
- [28] Lashon Booker. Improving search in genetic algorithms. *Genetic algorithms and simulated annealing*, pages 61–73, 1987.
- [29] Ginette Bouchard and Marcel Vallée. Mineral development statistics, a mine of information. *CIM Bulletin*, 93(1042):78–83, 2000.
- [30] S. Carpentier, M. Gamache, and R. Dimitrakopoulos. Underground long-term mine production scheduling with integrated geological risk management. *Mining Technology*, 125(2):93–102, 2016. ISSN 1474-9009 1743-2863. doi: 10.1179/1743286315y.0000000026.
- [31] Jenna Carr. An introduction to genetic algorithms. *Senior Project*, pages 1–40, 2014.
- [32] I Carwse. Multiple pass floating stope process. In *Proceedings of the Fourth Biennial Conference: Strategic Mine Planning*. Australasian Institute of Mining and Metallurgy Melbourne, 2001.
- [33] Noel Cressie. The origins of kriging. *Mathematical geology*, 22(3):239–252, 1990.
- [34] Clayton V Deutsch, Andre G Journel, et al. Geostatistical software library and user’s guide. *New York*, 119:147, 1992.
- [35] R Dimitrakopoulos and S Ramazan. Uncertainty-based production scheduling in open pit mining. *TRANSACTIONS*, 316, 2004.
- [36] R Dimitrakopoulos, L Martinez, and S Ramazan. A maximum upside/minimum downside approach to the traditional optimization of open pit mine design. *Journal of Mining Science*, 43(1):73–82, 2007. ISSN 1062-7391.
- [37] Roussos Dimitrakopoulos and Nikki Grieco. Stope design and geological uncertainty: quantification of risk in conventional designs and a probabilistic alternative. *Journal of mining science*, 45(2):152–163, 2009. ISSN 1062-7391.
- [38] Alex S Fraser. Simulation of genetic systems by automatic digital computers ii. effects of linkage on rates of advance under selection. *Australian Journal of Biological Sciences*, 10(4):492–500, 1957.
- [39] Alex S Fraser. Simulation of genetic systems by automatic digital computers i. introduction. *Australian Journal of Biological Sciences*, 10(4):484–491, 1957.

- [40] Norman Hanson. Lerchs grossman-contribution to reliable reserve estimates in surface mining. *AusIMM Bulletin*, 5:34–36, 2013.
- [41] John H Holland. Outline for a logical theory of adaptive systems. *Journal of the ACM (JACM)*, 9(3):297–314, 1962.
- [42] John H Holland. *Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence*. Ann Arbor, MI: University of Michigan Press, 1975.
- [43] William A Hustrulid, Mark Kuchta, and Randall K Martin. *Open Pit Mine Planning and Design, Two Volume Set & CD-ROM Pack*. CRC Press, 2013.
- [44] Wardell Armstrong International. Technical report on the neves-corvo mine, southern portugal. Wai 61-0465 technical report, Wardell Armstrong International, October 2007 2007.
- [45] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.
- [46] A Leite and R Dimitrakopoulos. Stochastic optimisation model for open pit mine planning: application and risk analysis at copper deposit. *Mining Technology*, 116(3):109–118, 2007. ISSN 1474-9009.
- [47] Aristotelis Mantoglou and John L Wilson. The turning bands method for simulation of random fields using line generation by a spectral method. *Water Resources Research*, 18(5):1379–1394, 1982.
- [48] J Ovanic and DS Young. Economic optimisation of stope geometry using separable programming with special branch and bound techniques. In *Third Canadian Conference on Computer Applications in the Mineral Industry*, pages 129–135.
- [49] Don Suneth Sameera Sandanayake, Erkan Topal, and Mohammad Waqar Ali Asad. A heuristic approach to optimal design of an underground mine stope layout. *Applied Soft Computing*, 30:595–603, 2015. ISSN 15684946. doi: 10.1016/j.asoc.2015.01.060.
- [50] Robert G Sargent. Verification and validation of simulation models. In *Proceedings of the 37th conference on Winter simulation*, pages 130–143. winter simulation conference, 2005.
- [51] J Sens and E Topal. A new algorithm for stope boundary optimisation. In *Proceedings of the The AusIMM New Leaders' Conference*. Australasian Institute of Mining and Metallurgy Melbourne.
- [52] SN Sivanandam and SN Deepa. *Introduction to genetic algorithms*. Springer Science & Business Media, 2007.
- [53] William M Spears and Kenneth A De Jong. An analysis of multi-point crossover. Technical report, DTIC Document, 1990.
- [54] M. L. Wardell Armstrong International (Owen and L.) Meyer. Lundin mining; ni 43-101 technical report for neves-corvo mine and semblana deposit, portugal. Ni 43-101 technical report, Wardell Armstrong International, 18 January 2013 2013.
- [55] Thomas Weise. *Global optimization algorithms-theory and application*. Self-published, 2, 2009.

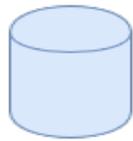
A

Flowchart Legend

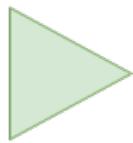
Model Schematics - Legend



Parameter



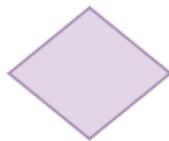
Core
Data
File



Go/NoGo
Decision
variable



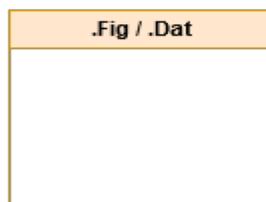
Task
Execution



UI-Selectable
Model Path
Selectable



Loop
Operator



Graphic
Output /
Additional
Log Data



Parameter
Flowpath



Model
Flowpath

Figure A.1: Flowchart legend

B

Flowcharts block model creator

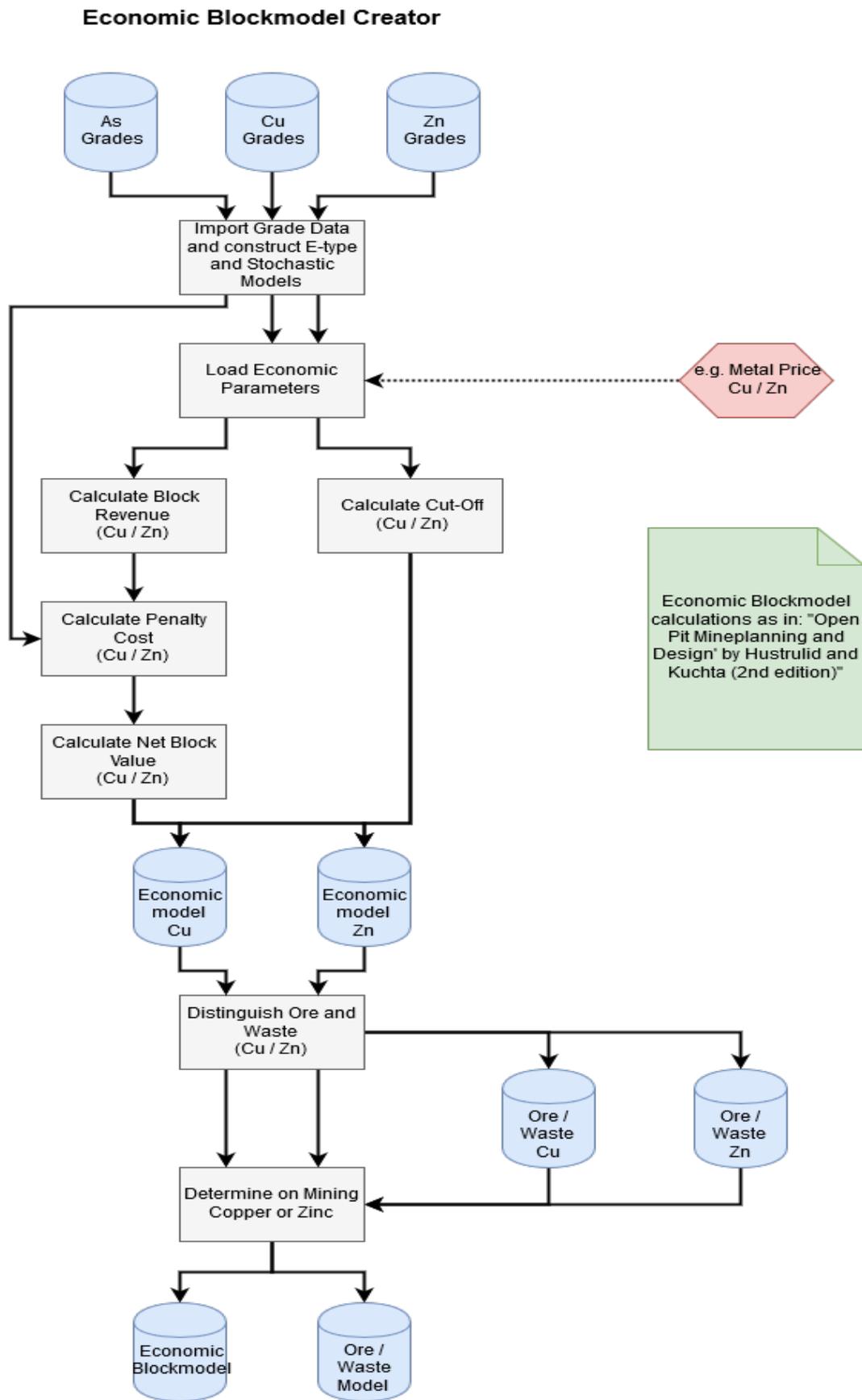


Figure B.1: Flowcharts block model creator

C

Flowcharts optimizer model

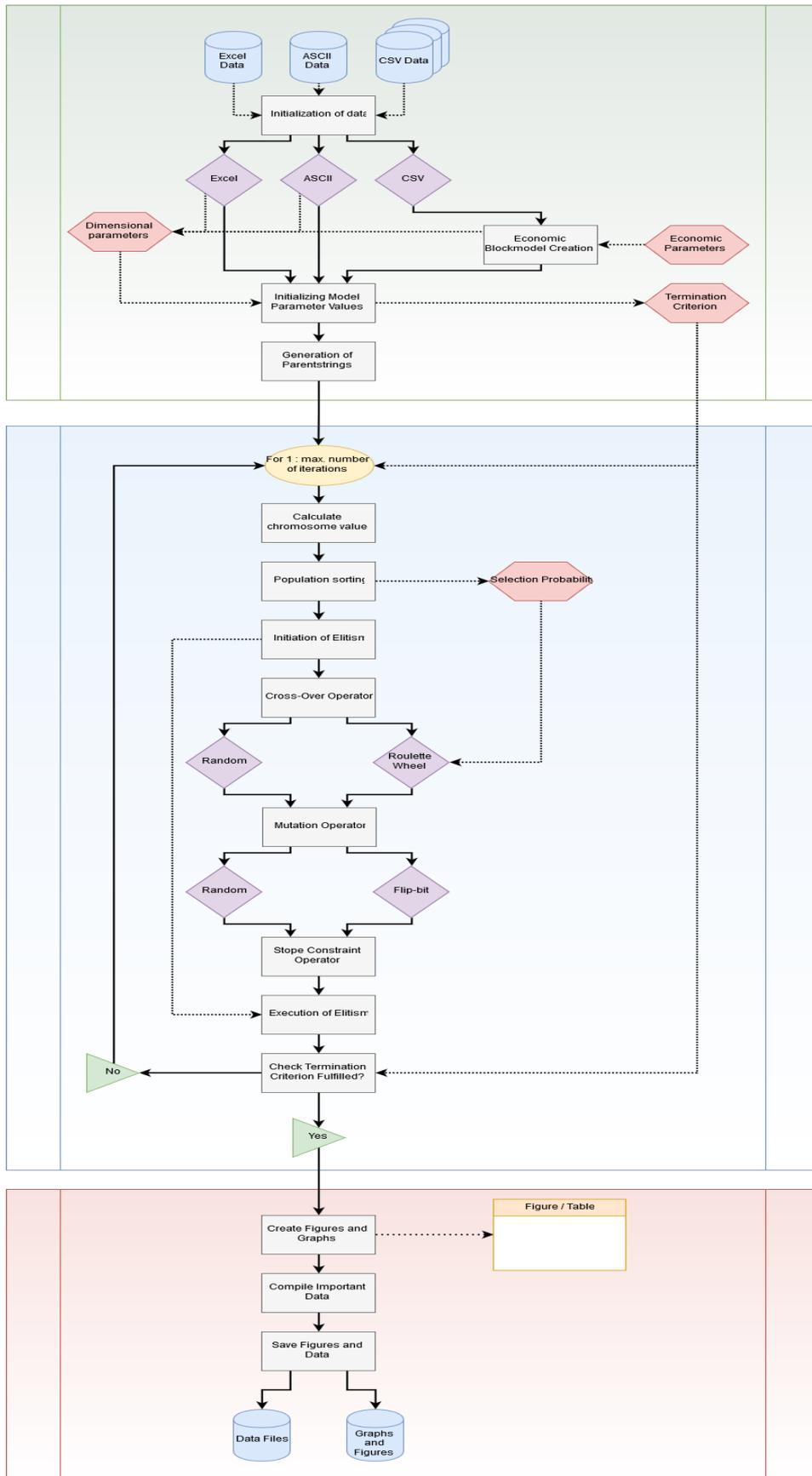


Figure C.1: Model overview

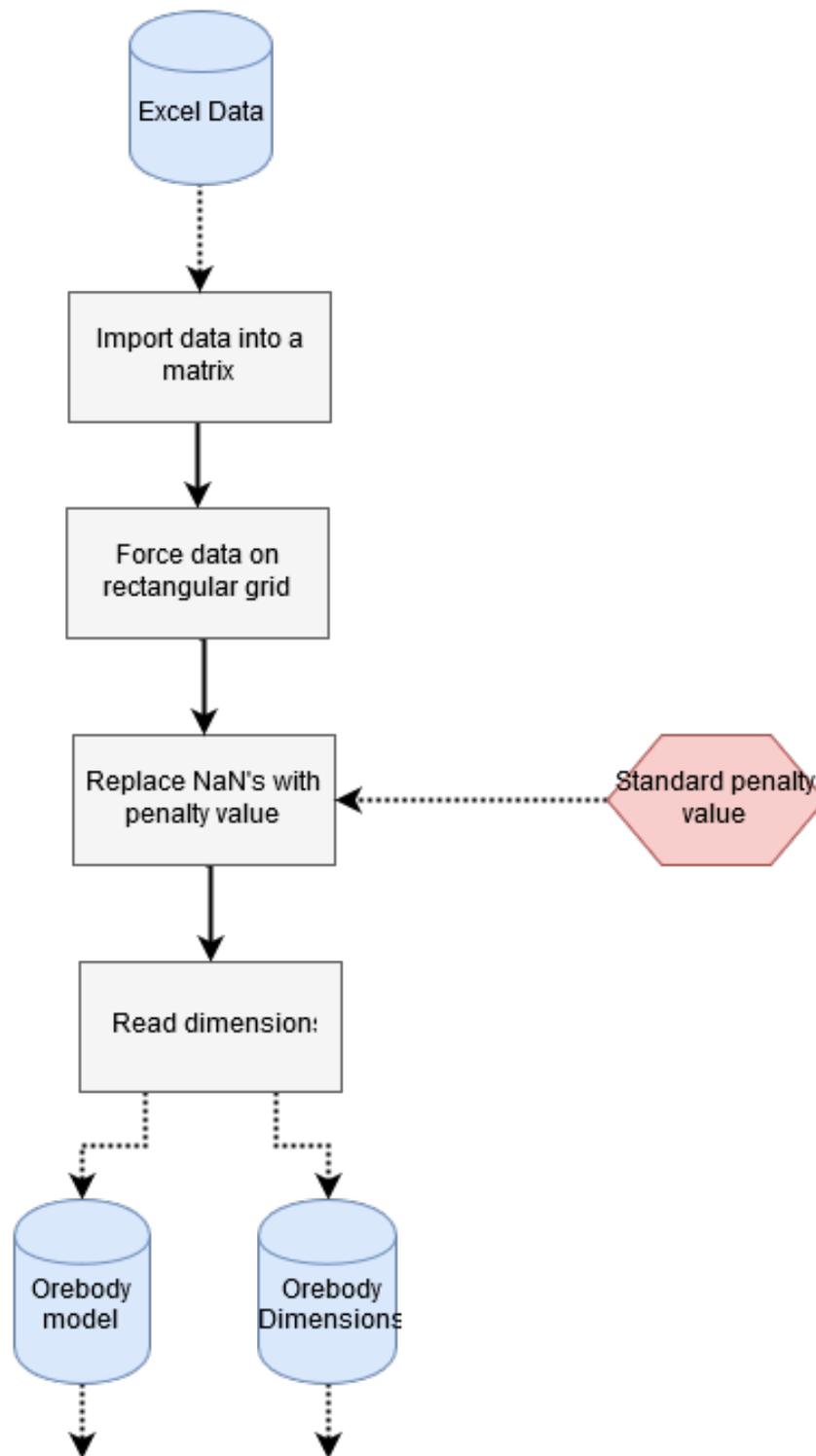


Figure C.2: Excel Import

ASCII Import Operator

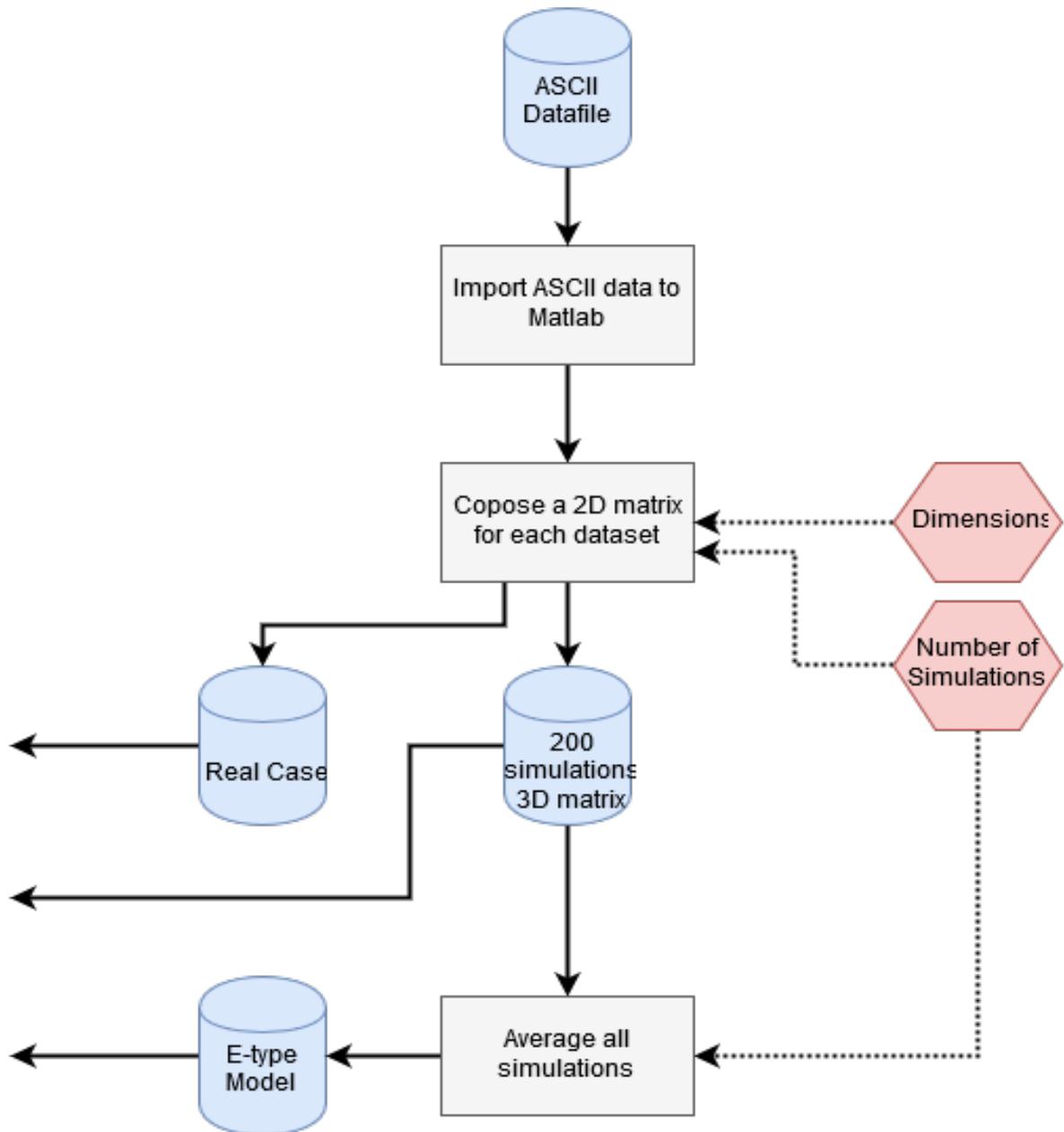


Figure C.3: ASCII Import

CSV Import Operator

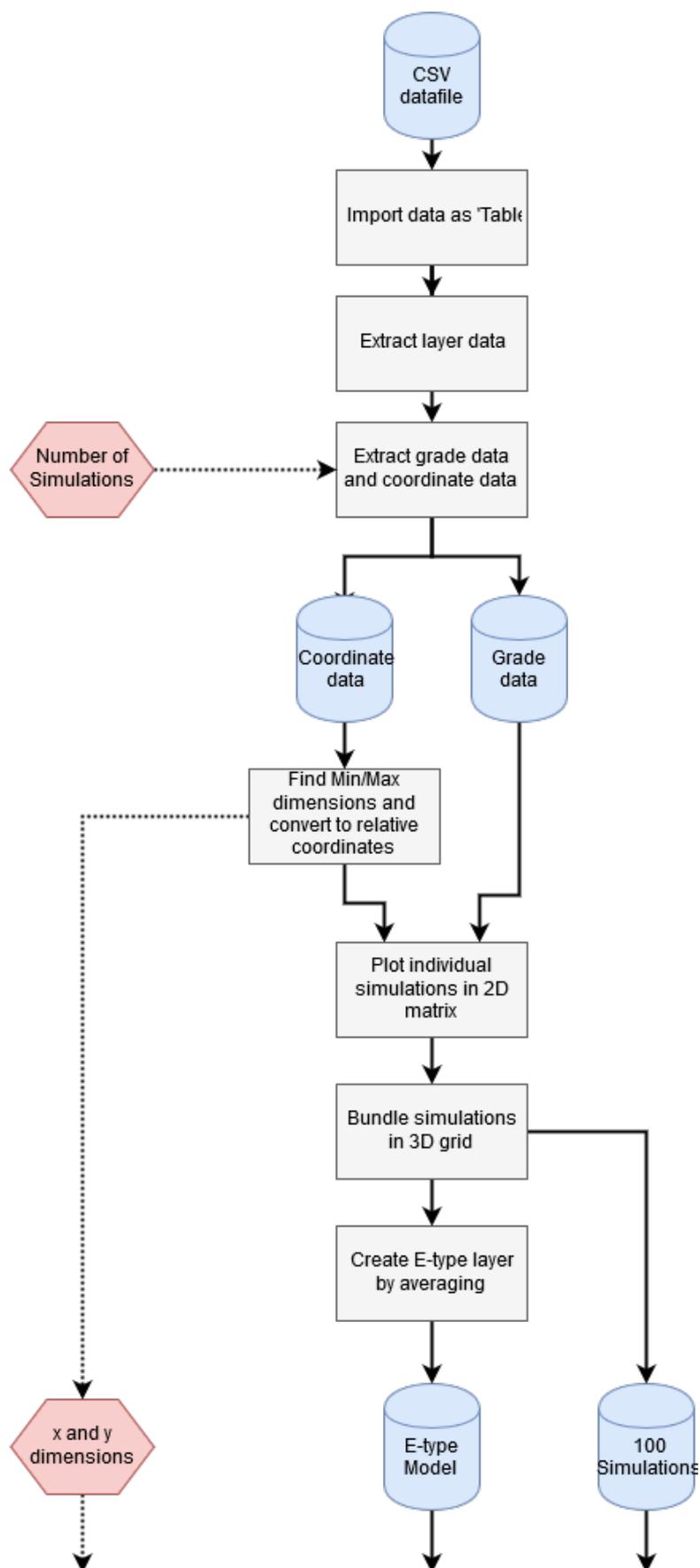


Figure C.4: CSV Import

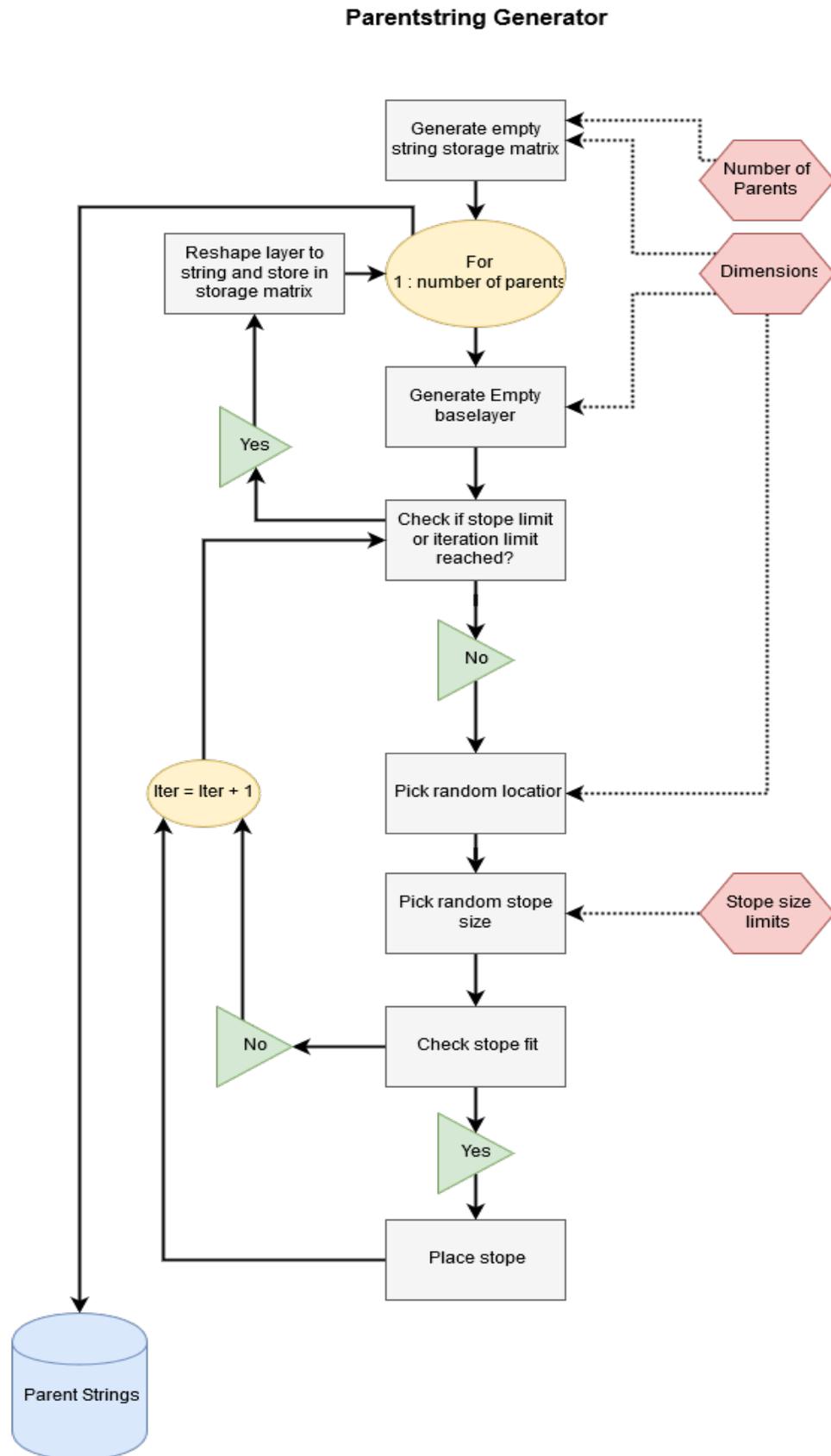


Figure C.5: Parentstring Generator

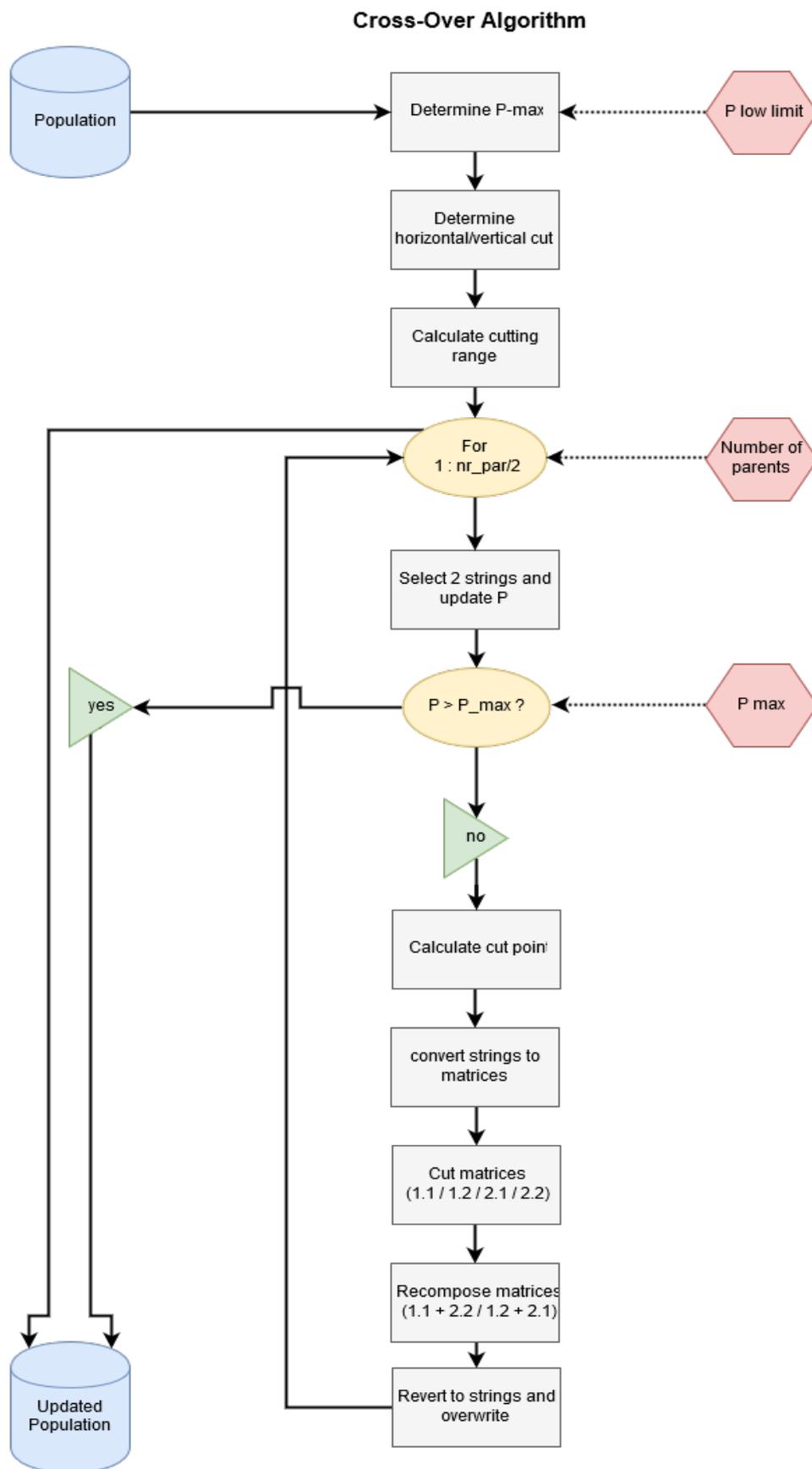


Figure C.6: Cross-Over Algorithm

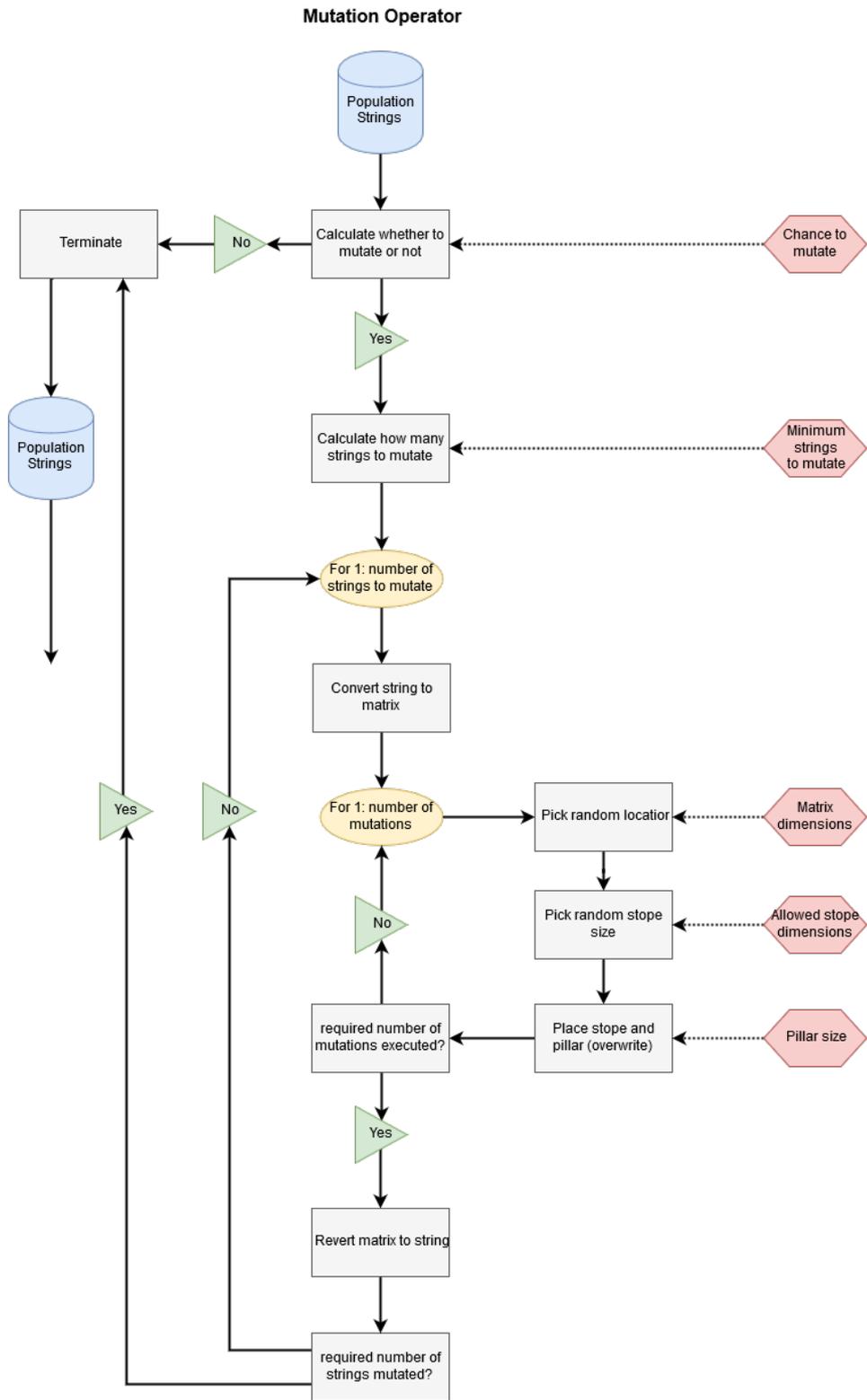


Figure C.7: Mutation

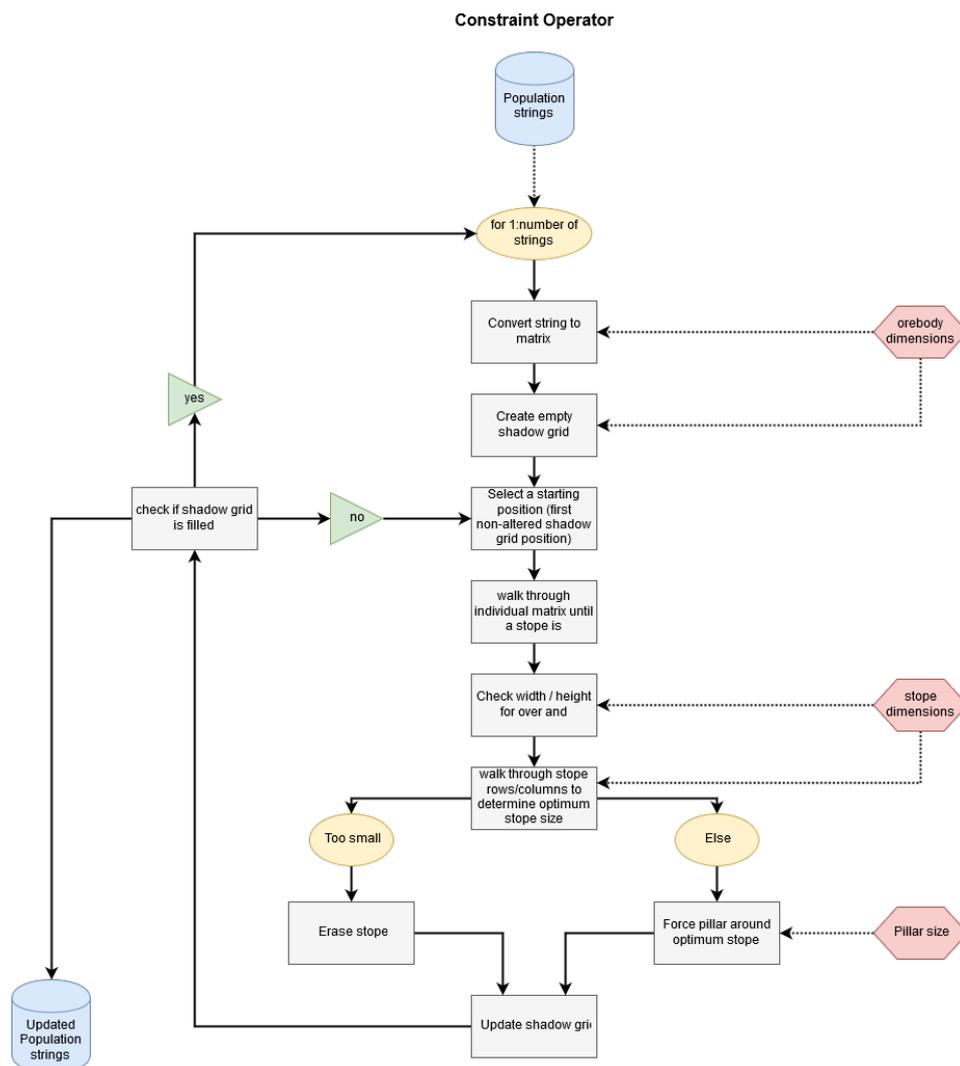


Figure C.8: Constraint Function

D

Model parameters and variables

Table D.1: Explanatory list of input parameters

Parameter	Model Assignment	Description
x_model	Parentstring generation, Crossover, Mutation, Constraint Function	Horizontal size of the value model used and stope layout dimensions. Assigned automatically
y_model	Parentstring generation, Crossover, Mutation, Constraint Function	Vertical size of the value model used and stope layout dimensions. Assigned automatically
string_length	Model Initialization, Mutation	Length of the binary chromosome string. Function of x and y dimensions
iter_model_max	Termination Criterion	Maximum number of algorithm iterations allowed
no_change_lim	Termination Criterion	Maximum number of algorithm iterations in which no positive value change was monitored allowed
stope_max	Parentstring generation	Maximum number of stopes allowed in first generation individuals
iter_max	Parentstring generation	Maximum number of stope placement attempts allowed while composing a first generation individual
x_stope_max	Parentstring generation, Mutation, Constraint Function	Maximum horizontal stope size allowed
y_stope_max	Parentstring generation, Mutation, Constraint Function	Maximum vertical stope size allowed
x_stope_min	Parentstring generation, Mutation, Constraint Function	Minimal horizontal stope size allowed
y_stope_min	Parentstring generation, Mutation	Minimal vertical stope size allowed
x_stope_avg	Model Initialization	Theoretic average horizontal stope size
y_stope_avg	Model Initialization	Theoretic average vertical stope size
barrier	Parentstring generation, Constraint Function	Pillar size; number of blocks to be left unmined between individual stopes
nrpar	Parentstring generation, Value Calculator, Crossover, Constraint Function	Size of the chromosome pool. This should be an even number.
crossover_probability	Crossover	Probability of crossover being executed during an iteration
p_low_lim	Crossover	The lowest cumulative P-value allowed. Forces a minimal number individuals to be crossed-over
cutting_range	Crossover	The range on which the cut-point can be selected
chance_mut	Mutation	Probability of mutation being executed during an iteration

chance_mut_flipbit	Mutation	Probability of flip-bit mutation being executed during an iteration
nr_mut	Mutation	Number of stopes to place on an individual if it is selected for mutation
perc_mut	Mutation	Percentage of individuals to be selected for mutation if mutation is executed
min_par_to_mut	Mutation	Minimal number of strings to mutate if mutation is executed
selection_pressure	Selection	Determines the probability of the best individual to be selected relative to the average selection probability
nr_sim	Value Calculator	Number of simulations taken into account when running stochastically. For average cases nr_sim = 1

Table D.2: Explanatory list of model output variables

Variable	Model Assignment	Description
orebodies	Data Import, Value Calculation,	3D matrix containing economic block models for all individual simulations
parents	Parentstring Generation, Initialization	Matrix containing binary chromosome strings for all individuals
iter	Main	Algorithm Current iteration
value	Value Calculation, Population sorting	Economic value of a certain stope layout / economic block model combination
value_mat_combined	Value Calculation	Matrix containing best, worst, mean, 25th and 75th solution values for all stope layouts, when referenced to all orebody simulations
value_mat_complete	Value Calculation	Matrix containing solution values of all stope layouts for all simulations
sorted_stochastic_results	Population Sorting	Value sorted value_mat_combined based on the referenced result
sorted_stochastic_results_complete	Population Sorting	Value sorted 'value_mat_complete' based on the referenced result
best_parent	Selection/Elitism	The best performing stope layout
best_parent2	Selection/Elitism	The second best performing stope layout
P	Selection	Relative selection probability value
flipbit_val_old	Mutation	Value of individual prior to mutation
flipbit_val_new	Mutation	Value of individual after mutation
solution_best_mat	Post Algorithm Processing	Matrix containing the best solution optimization value for each iterations
solution_worst_mat	Post Algorithm Processing	Matrix containing the worst solution optimization value for each iterations
no_change_count	Termination Criterion	Records the number of iterations in which the best solution value remains unchanged
beta	Selection	Selection pressure, used in calculating P
population	Value Calculation, Value Sorting, Crossover, Mutation, Constraint Function, Eliting	Matrix containing stope layout chromosomes for all individuals in a generation

Table D.3: Explanatory list of model output values

Output	Model Location	Description
runtime	Main Algorithm	Time needed for run to complete (in seconds)
w	Termination Criterion	Waitbar in which number of iterations is displayed live
w2	Termination Criterion	Waitbar in which number of iterations without a change is displayed live
img	Termination Criterion	Image used to terminate the model (when clicked upon figure)
abort	Termination Criterion	Indicates whether run was aborted early (1) or not (0)
sol_stoch_worst	Post Algorithm Processing	Lowest stochastic solution value for best stope layout
sol_stoch_25	Post Algorithm Processing	25th percentile stochastic solution value for best stope layout
sol_stoch_50	Post Algorithm Processing	50th percentile stochastic solution value for best stope layout
sol_stoch_75	Post Algorithm Processing	75th percentile stochastic solution value for best stope layout
sol_stoch_best	Post Algorithm Processing	Highest stochastic solution value for best stope layout
sol_stoch_mean	Post Algorithm Processing	Mean stochastic solution value for best stope layout
value_real	Post Algorithm Processing	Value of best individual stope layout when combined with the 'real' orebody model
solution_best_mat	Post Algorithm Processing	Matrix containing the best solution optimization value for each iterations
solution_worst_mat	Post Algorithm Processing	Matrix containing the worst solution optimization value for each iterations
flag	Termination Criterion	Flag is set to one when no change limit is reached, after which genetic algorithm is terminated
iter_check	Termination Criterion	Registers the final number of iterations

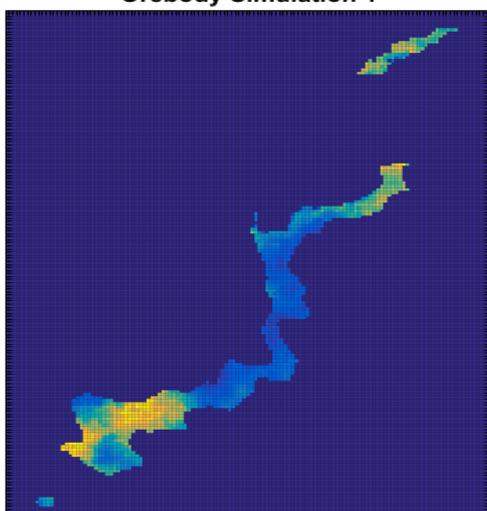
Table D.4: Input parameters economic

Parameter Model Name	Model Assignment	Description
rock_density	Economic Blockmodel Creation	Average density of a cubic meter of rock within the deposit.
block_size	Economic Blockmodel Creation	Size of a block within the economic block model.
mining_recovery	Economic Blockmodel Creation	Percentage of material recovered after a block is mined.
arsenic_penalty	Economic Blockmodel Creation	Penalty value for the inclusion of high levels of arsenic within a block of ore. Penalty dependent on amount of arsenic present.
processing_cost_zn	Economic Blockmodel Creation	Cost of processing a tonne of ore in the zinc processing plant.
processing_cost_cu	Economic Blockmodel Creation	Cost of processing a tonne of ore in the copper processing plant.
mining_cost_zn	Economic Blockmodel Creation	Cost of mining a block of zinc ore. Derived from the mining and processing cut-off parameter.
mining_cost_cu	Economic Blockmodel Creation	Cost of mining a block of copper ore. Derived from the mining and processing cut-off parameter.
mining_and_processing_cutoff_zn	Economic Blockmodel Creation	Limit cost of mining and processing a tonne of zinc ore combined.
mining_and_processing_cutoff_cu	Economic Blockmodel Creation	Limit cost of mining and processing a tonne of copper ore combined.
processing_recovery_zn	Economic Blockmodel Creation	Percentage of zinc ore recovered after processing.
processing_recovery_cu	Economic Blockmodel Creation	Percentage of copper ore recovered after processing.
metal_price_zn	Economic Blockmodel Creation	Value of a tonne of zinc.
metal_price_cu	Economic Blockmodel Creation	Value of a tonne of copper.
refining_cost_zn	Economic Blockmodel Creation	Cost of refining a tonne of milled/processed zinc ore.
refining_cost_cu	Economic Blockmodel Creation	Cost of refining a tonne of milled/processed copper ore.

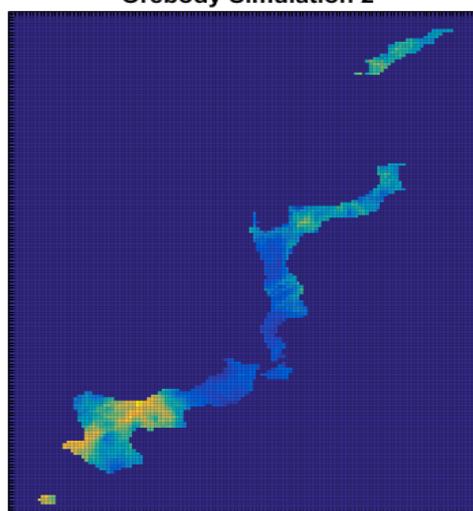
E

Economic baseline blockmodels analysis 1 and 2

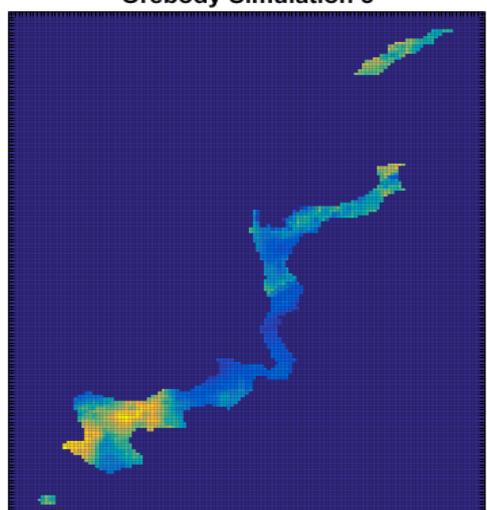
Orebody Simulation 1



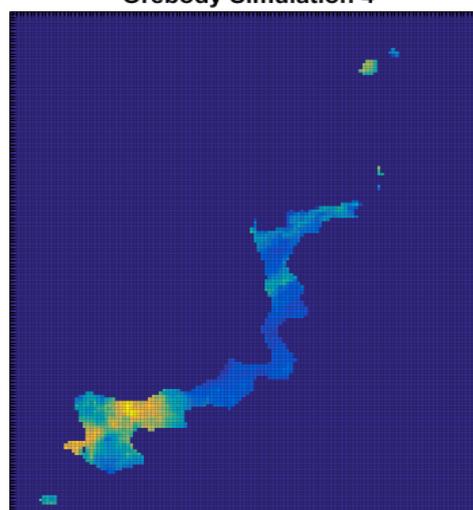
Orebody Simulation 2



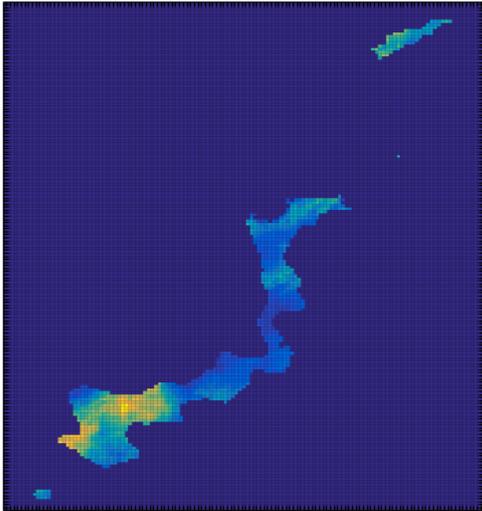
Orebody Simulation 3



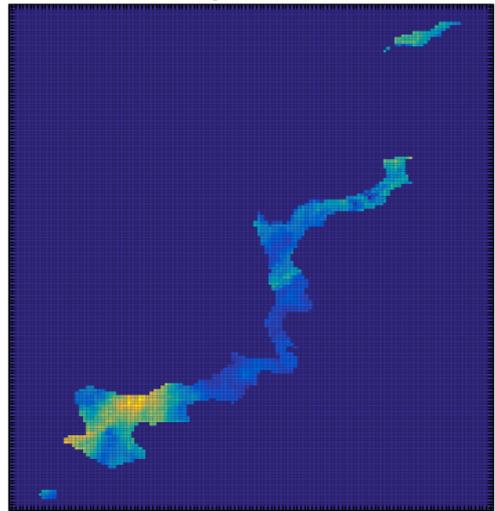
Orebody Simulation 4



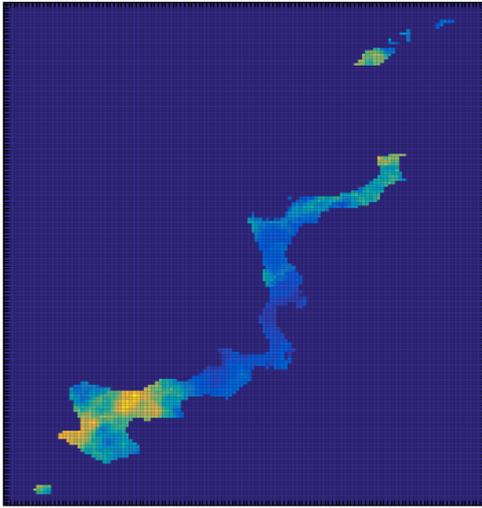
Orebody Simulation 5



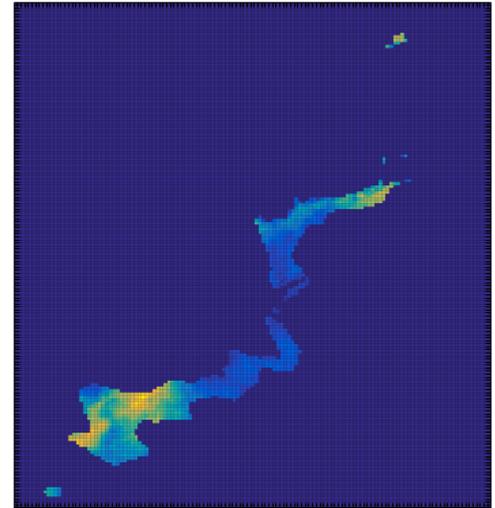
Orebody Simulation 6



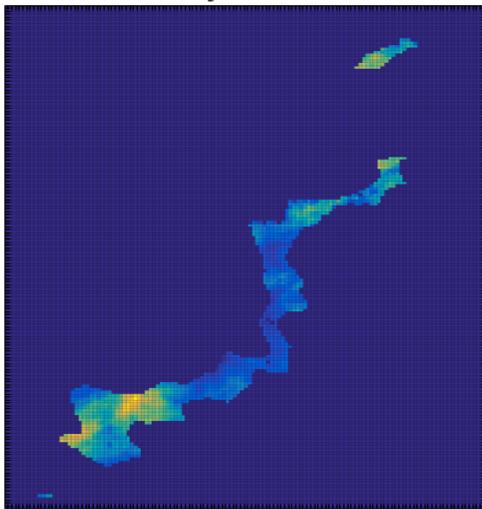
Orebody Simulation 7



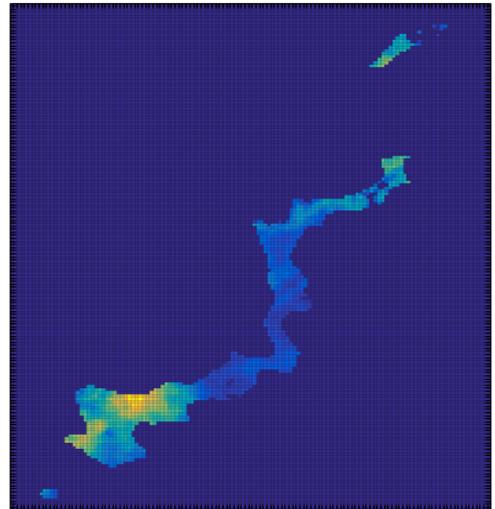
Orebody Simulation 8

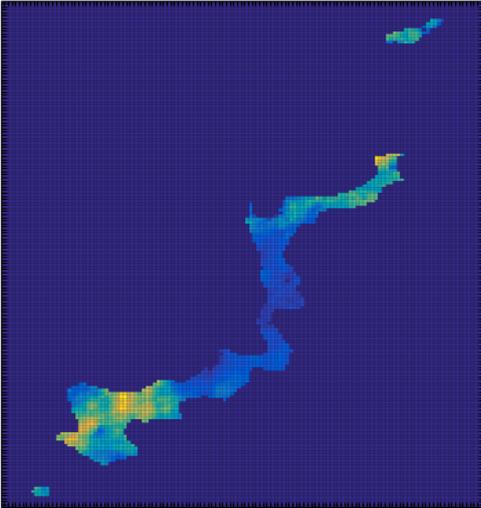
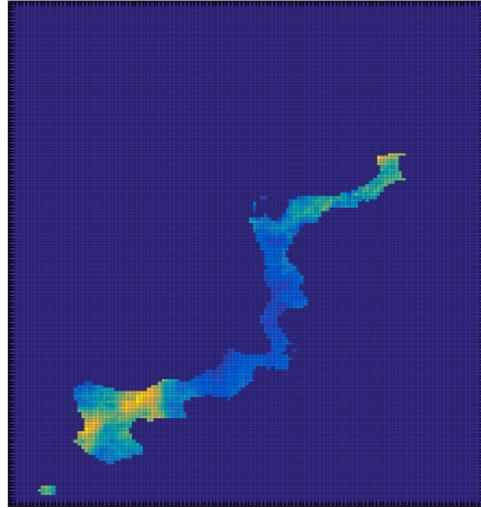
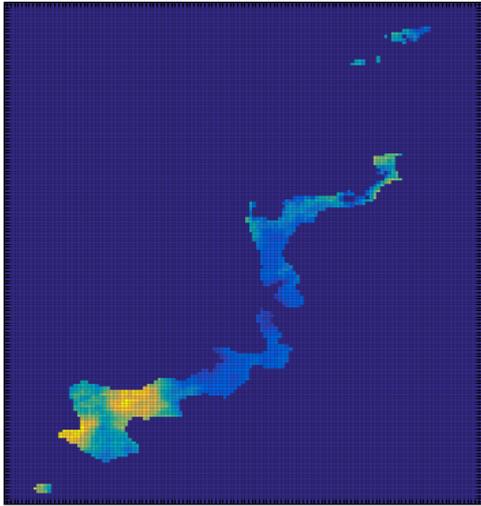
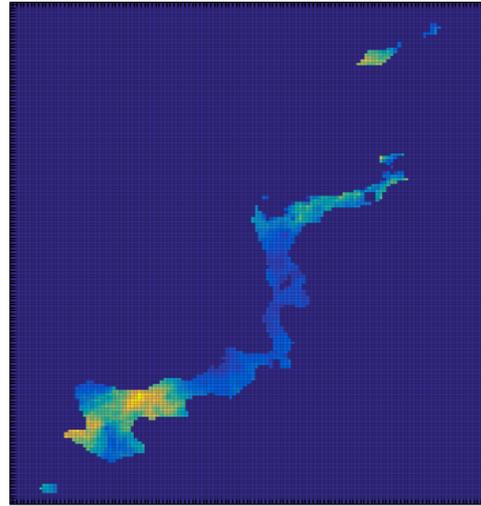
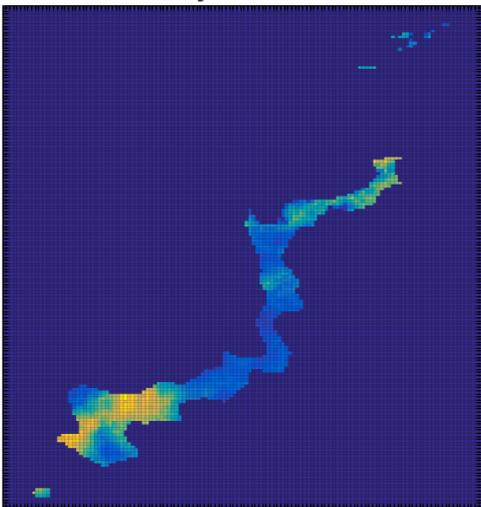
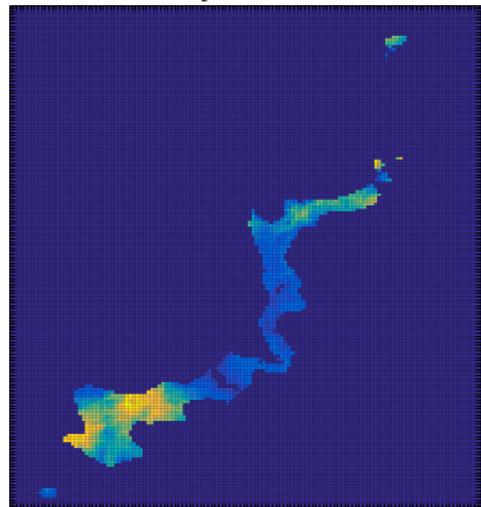


Orebody Simulation 9

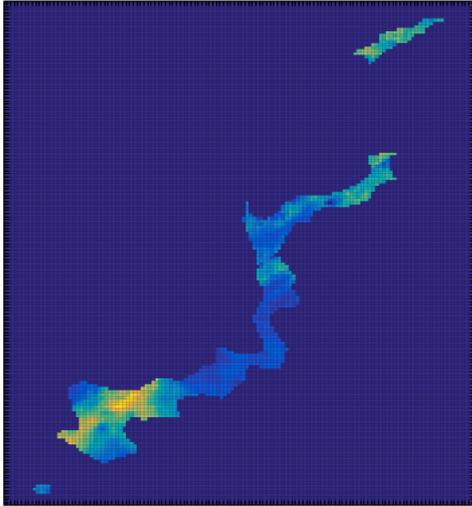


Orebody Simulation 10

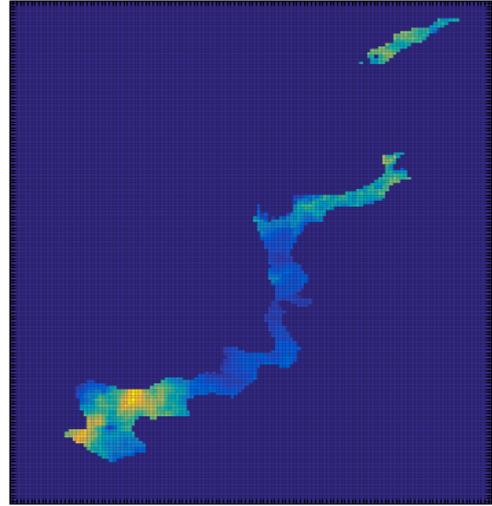


Orebody Simulation 11**Orebody Simulation 12****Orebody Simulation 13****Orebody Simulation 14****Orebody Simulation 15****Orebody Simulation 16**

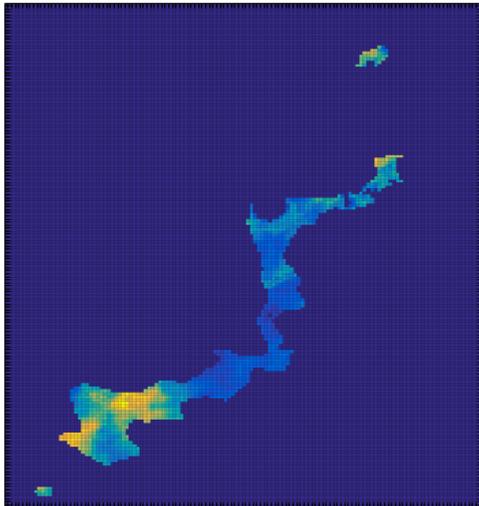
Orebody Simulation 17



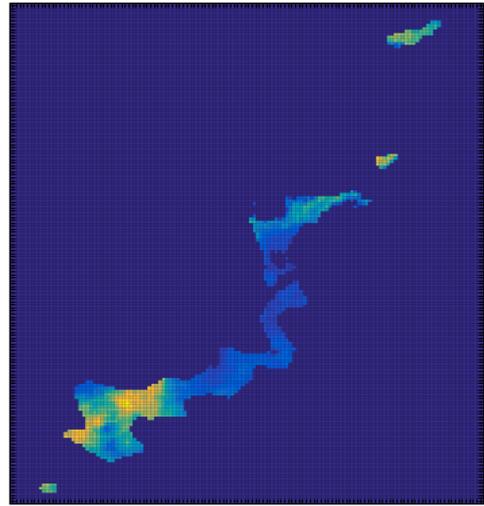
Orebody Simulation 18



Orebody Simulation 19



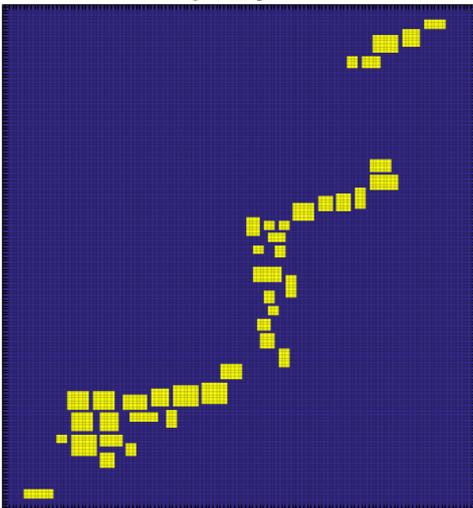
Orebody Simulation 20



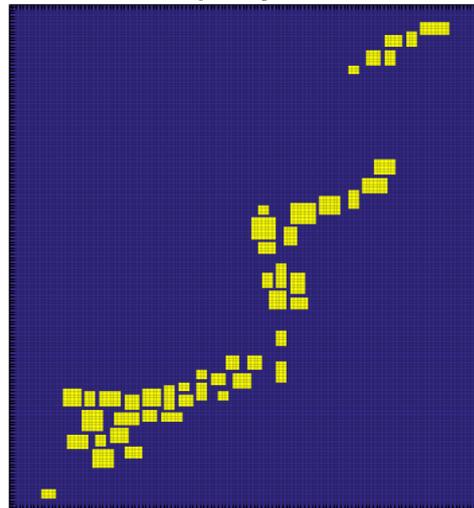
F

Stope layout analysis 1 and 2

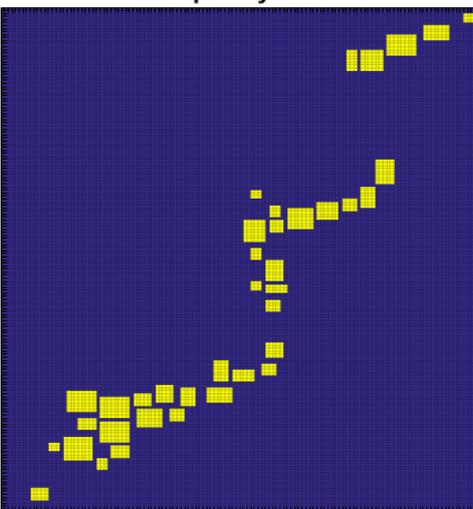
Stope Layout 1



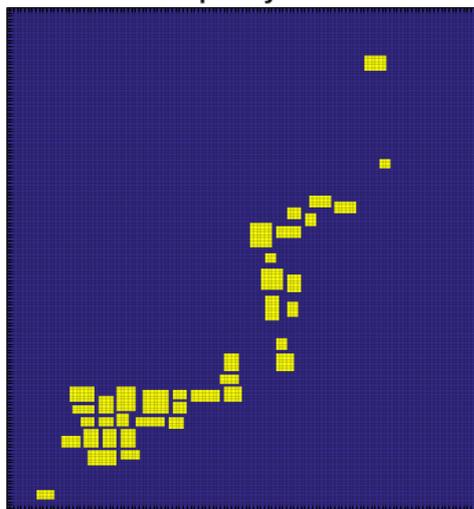
Stope Layout 2



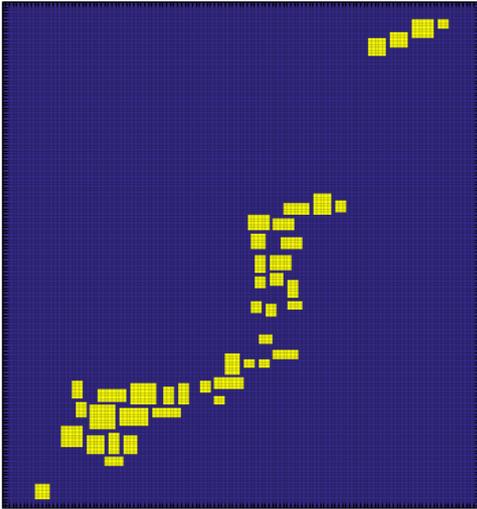
Stope Layout 3



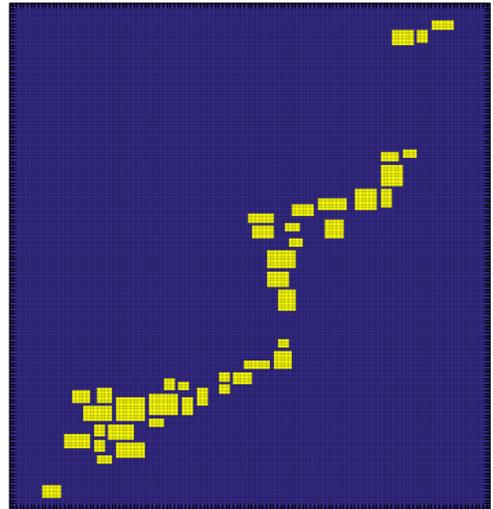
Stope Layout 4



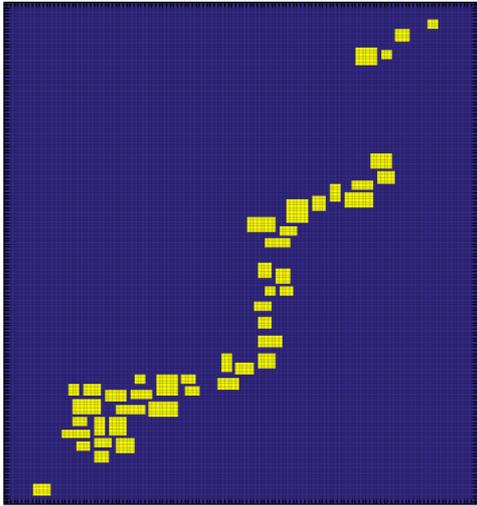
Stope Layout 5



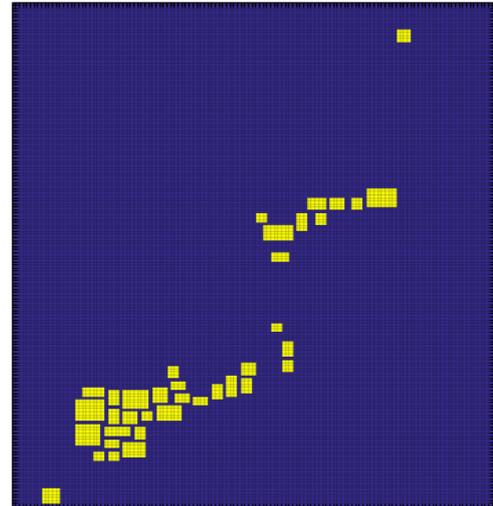
Stope Layout 6



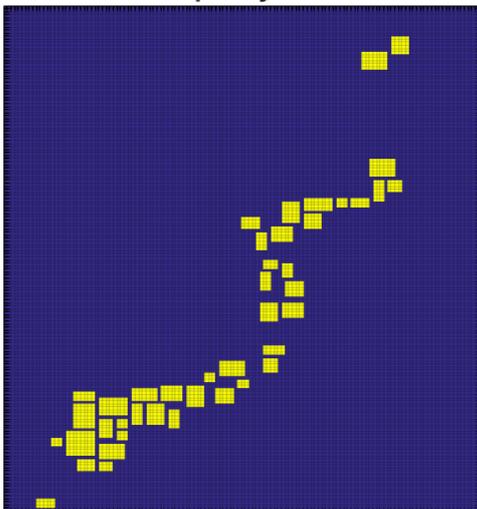
Stope Layout 7



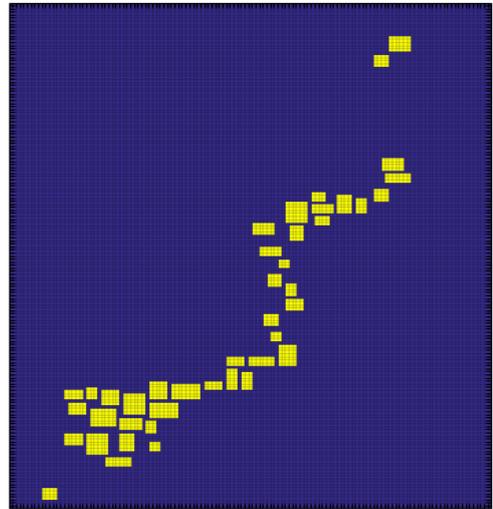
Stope Layout 8



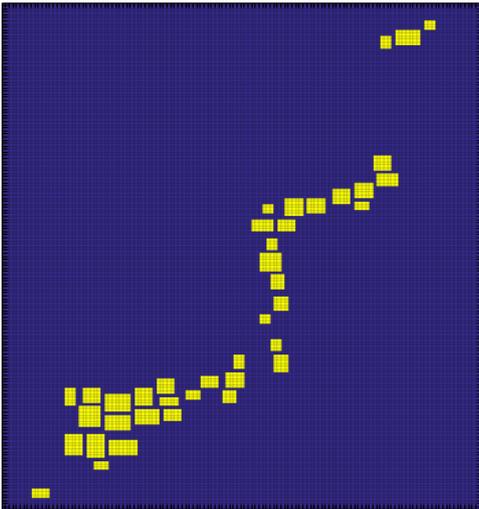
Stope Layout 9



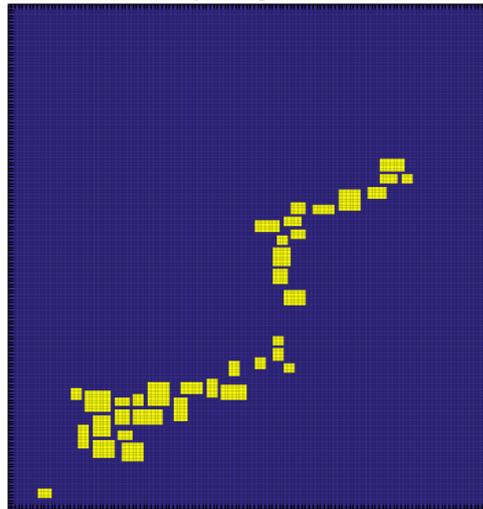
Stope Layout 10



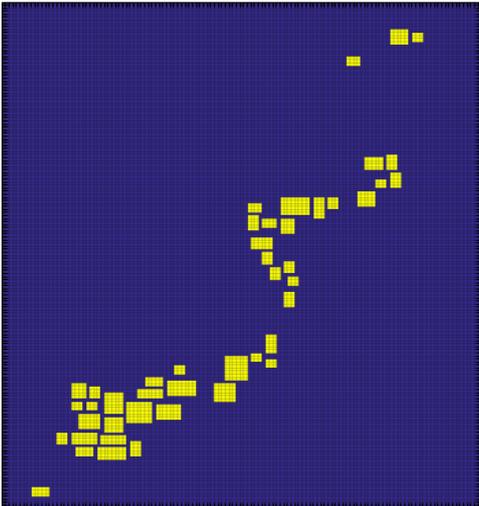
Stope Layout 11



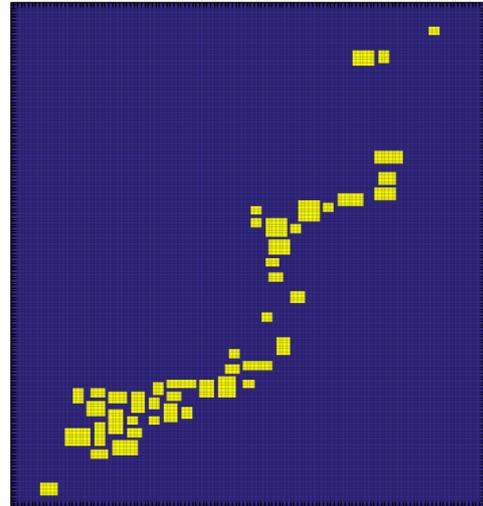
Stope Layout 12



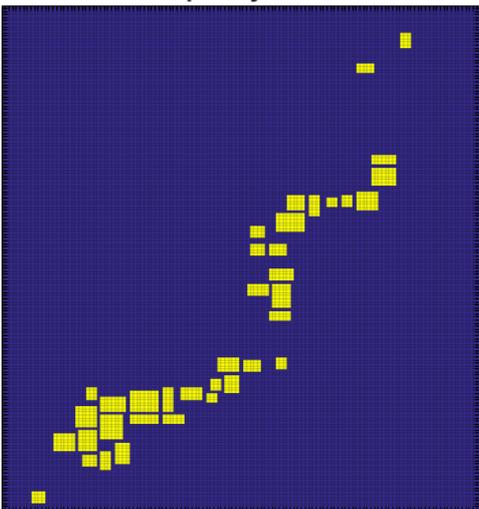
Stope Layout 13



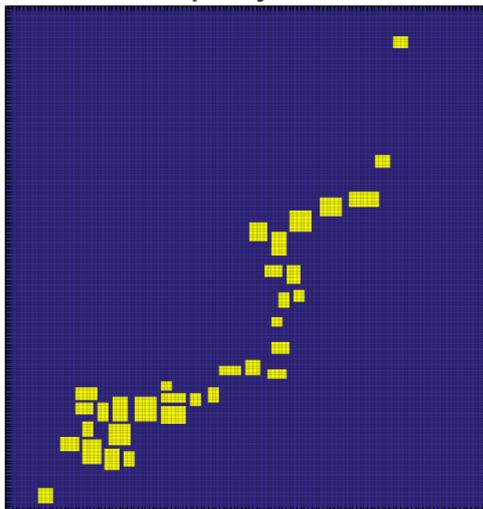
Stope Layout 14



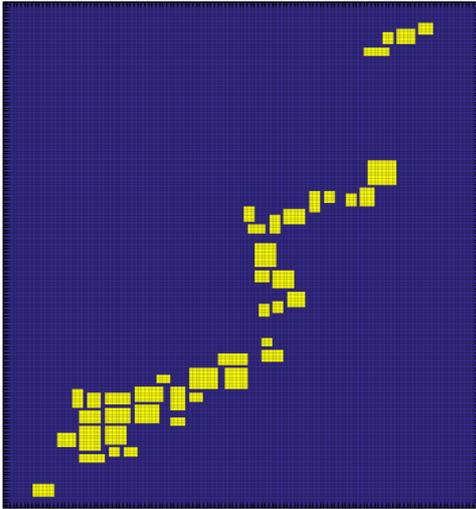
Stope Layout 15



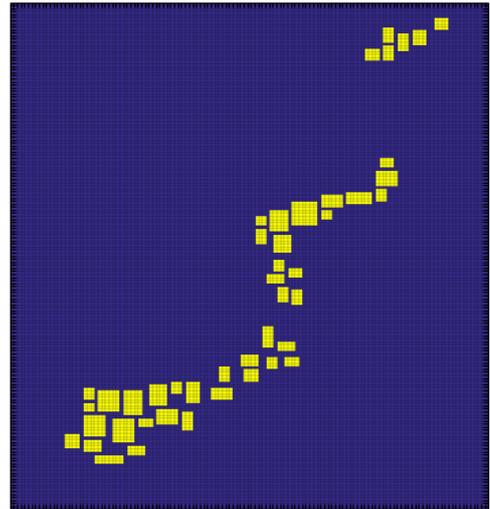
Stope Layout 16



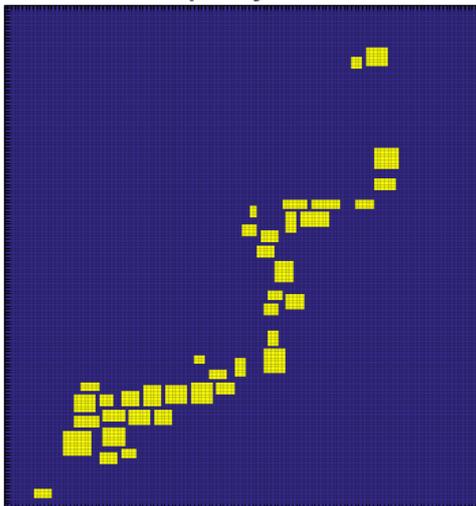
Stope Layout 17



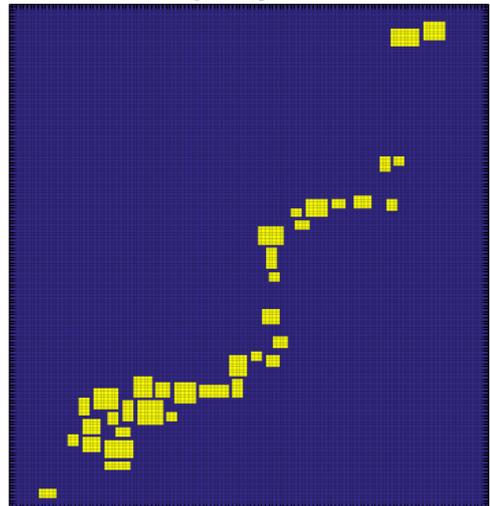
Stope Layout 18



Stope Layout 19



Stope Layout 20





Excel Files

17,737,460.96	17,521,452.78	17,601,825.76	14,441,797.11	14,328,435.03	18,380,279.84	16,891,745.06	15,071,113.75	17,389,102.89	17,317,640.40	Orebody 1
18,971,334.04	19,393,053.19	19,021,749.09	16,116,747.10	15,868,536.39	18,015,859.33	18,353,805.76	15,593,764.29	18,562,244.04	17,299,883.00	Orebody 2
19,498,709.69	18,851,705.19	20,516,022.95	16,350,952.82	15,843,302.68	17,865,030.77	18,584,899.72	15,249,512.38	18,998,757.47	17,347,608.06	Orebody 3
12,657,966.09	13,245,623.20	13,098,408.60	14,342,982.78	12,428,663.59	13,192,796.21	12,565,211.38	12,734,269.21	13,406,808.15	13,282,270.34	Orebody 4
16,483,905.52	16,645,792.60	16,626,985.72	15,592,961.62	15,874,778.31	16,111,675.94	14,615,373.44	14,611,896.47	15,892,609.05	15,318,209.32	Orebody 5
16,246,645.65	16,594,880.56	16,542,122.78	15,037,691.01	14,694,756.96	16,660,288.29	15,357,351.14	13,989,396.46	15,746,764.06	16,279,437.37	Orebody 6
15,097,469.86	15,359,458.54	15,467,513.21	14,435,749.55	12,936,002.07	15,269,387.16	15,856,098.37	13,932,069.34	15,142,525.28	15,442,524.43	Orebody 7
12,971,445.56	12,837,715.40	14,784,859.60	13,061,479.00	12,255,935.82	13,932,074.27	13,137,420.68	14,168,683.58	13,101,153.04	14,046,645.05	Orebody 8
15,554,038.66	15,665,943.01	15,100,620.98	15,244,646.85	13,127,838.77	14,477,724.91	15,706,274.84	13,394,729.80	16,095,557.38	15,723,397.91	Orebody 9
15,492,861.87	16,477,953.39	16,685,279.77	15,743,091.88	15,119,327.59	16,014,741.48	15,754,430.12	14,282,282.77	16,525,208.04	17,001,158.06	Orebody 10
16,293,637.69	17,079,466.45	17,230,014.04	14,682,974.96	13,803,099.89	17,279,196.81	17,098,563.63	15,031,986.65	16,475,388.90	16,741,283.41	Orebody 11
13,100,699.31	13,738,806.47	13,201,880.20	12,676,853.26	11,251,344.63	12,853,814.39	13,432,816.52	12,366,414.81	13,877,698.01	13,605,675.59	Orebody 12
13,310,929.66	13,508,562.83	13,843,974.58	14,340,370.28	12,622,315.47	13,110,035.77	13,417,887.63	13,605,716.85	14,073,244.82	13,918,266.66	Orebody 13
12,486,538.68	12,758,185.16	13,166,819.61	13,211,609.58	11,806,906.33	11,934,747.57	12,884,585.37	12,605,743.49	12,698,037.06	12,733,834.08	Orebody 14
15,627,654.63	16,281,582.08	15,482,501.59	15,599,965.89	14,974,616.59	15,587,510.21	15,590,990.70	14,758,615.36	15,945,782.27	15,973,446.97	Orebody 15
13,859,774.62	14,388,759.79	14,545,185.33	13,912,945.13	13,293,139.61	14,275,468.17	14,164,634.61	14,200,121.59	14,274,063.65	13,887,984.82	Orebody 16
16,311,609.67	16,306,672.90	16,472,450.36	13,362,348.34	14,102,408.72	16,138,458.97	15,659,163.70	12,264,637.52	15,195,641.09	14,684,434.85	Orebody 17
16,467,813.01	16,259,856.30	16,488,936.53	13,292,772.80	14,389,453.23	16,038,665.26	14,977,604.97	13,816,120.03	15,735,315.07	15,421,610.96	Orebody 18
13,921,945.03	14,548,917.98	14,264,530.05	15,078,454.27	13,724,993.45	14,171,438.83	14,037,259.86	13,784,784.31	14,290,958.52	14,184,192.80	Orebody 19
12,154,679.08	12,815,481.49	12,855,339.93	12,685,562.73	12,551,944.14	12,842,954.55	12,487,056.42	12,089,952.04	12,771,044.06	13,175,958.66	Orebody 20
Envelope 1	Envelope 2	Envelope 3	Envelope 4	Envelope 5	Envelope 6	Envelope 7	Envelope 8	Envelope 9	Envelope 10	

Figure G.1: Analysis 1 Cross value stope envelopes 20

17,730,850.36	16,049,815.41	17,330,370.25	14,857,579.83	17,745,436.12	14,794,836.85	18,687,449.35	16,892,875.43	16,088,049.07	16,214,867.57	Orebody 1
18,712,024.31	16,778,966.95	18,387,263.63	15,707,073.27	18,504,583.77	16,739,162.37	19,102,794.44	17,866,511.79	17,334,402.79	16,434,815.55	Orebody 2
17,889,135.40	15,391,891.05	17,164,758.18	16,130,834.96	17,493,566.06	15,550,535.15	18,414,949.64	18,834,880.39	16,875,230.59	15,731,091.00	Orebody 3
13,283,572.55	12,605,990.95	12,965,161.34	11,675,242.67	13,605,715.74	12,597,713.37	13,869,181.29	12,764,490.44	12,970,406.75	12,506,366.21	Orebody 4
16,592,953.80	13,947,486.52	15,992,518.38	13,225,932.05	16,267,557.80	15,142,535.47	16,885,394.91	15,927,423.61	14,496,690.00	15,503,727.71	Orebody 5
16,703,455.96	15,006,102.12	15,828,199.74	13,231,813.58	16,659,262.05	14,736,741.68	16,785,521.09	16,358,650.02	14,962,129.83	14,705,187.16	Orebody 6
15,609,132.70	14,558,332.20	15,059,251.44	13,565,729.35	15,664,697.61	13,753,197.20	16,268,355.20	15,200,628.80	15,322,551.44	13,562,969.35	Orebody 7
14,397,327.55	13,768,421.80	13,962,832.23	13,041,836.20	14,043,942.88	13,297,033.26	14,304,344.25	13,341,373.18	13,344,584.17	12,928,811.09	Orebody 8
15,485,960.38	14,349,904.30	15,548,859.00	14,318,534.55	16,808,834.24	14,119,557.78	15,692,726.81	15,148,063.25	15,535,470.87	13,452,098.46	Orebody 9
16,687,661.64	15,092,593.09	16,048,855.55	14,578,532.49	16,073,450.40	15,212,190.41	16,001,755.57	15,274,122.09	16,316,303.27	15,389,256.69	Orebody 10
18,295,165.58	16,912,951.77	16,466,368.41	15,528,558.35	17,370,998.05	15,366,046.30	16,825,990.12	16,659,889.20	15,968,228.11	14,623,777.79	Orebody 11
13,784,295.96	13,422,389.45	13,405,472.52	12,236,145.41	14,163,496.90	12,121,495.26	13,157,702.86	13,048,538.04	13,422,741.15	12,181,224.78	Orebody 12
13,903,241.92	13,382,997.01	14,554,743.42	12,164,119.31	14,170,891.68	13,464,273.58	13,943,844.51	13,341,358.38	14,118,163.02	12,991,035.52	Orebody 13
12,820,221.79	12,180,599.69	12,853,868.07	12,250,026.45	12,922,036.92	12,585,119.19	12,630,602.88	11,940,107.29	12,664,659.63	11,262,274.84	Orebody 14
16,079,806.73	15,183,442.73	15,775,868.64	14,680,328.49	16,830,881.99	15,033,504.71	16,432,142.95	15,260,486.47	16,293,359.73	14,373,416.30	Orebody 15
14,665,019.65	13,368,957.86	13,528,210.05	13,084,602.28	14,342,071.69	14,673,044.44	13,866,809.72	13,955,382.06	13,767,791.64	12,859,380.46	Orebody 16
16,656,058.69	13,491,227.84	14,525,648.23	13,566,321.95	15,470,772.00	13,426,563.43	17,320,932.72	14,162,896.00	14,292,896.00	14,295,134.80	Orebody 17
16,452,368.14	13,862,324.16	15,020,768.36	13,071,327.46	15,152,089.88	14,695,311.05	15,824,209.31	15,964,242.21	13,790,342.21	14,924,881.58	Orebody 18
15,005,028.36	13,986,097.56	14,894,442.54	12,743,751.12	15,156,521.38	14,656,592.84	14,966,599.57	13,869,105.74	14,716,443.22	13,353,622.23	Orebody 19
13,426,364.48	11,760,698.65	13,149,529.72	11,666,634.24	13,275,486.84	12,955,610.23	13,027,090.76	12,138,637.36	12,573,686.70	12,701,496.48	Orebody 20
Envelope 11	Envelope 12	Envelope 13	Envelope 14	Envelope 15	Envelope 16	Envelope 17	Envelope 18	Envelope 19	Envelope 20	

Figure G.2: Analysis 1 Cross value stope envelopes 20

	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%	
	0.154522	0.151134	0.146417	0.14592	0.14493	0.143614	0.141347	0.142398	0.142042	0.142684	0.14252	0.14493	0.145454	0.148899	0.149974	0.154545	0.157665	0.160889	0.191156	0.231757	Orebody
	0.15704	0.15519	0.150249	0.1501	0.14955	0.147622	0.14613	0.145811	0.145944	0.146028	0.146905	0.151068	0.153107	0.158373	0.163189	0.169743	0.181847	0.198187	0.236318	0.251751	Orebody
	0.14778	0.14537	0.140778	0.140664	0.14037	0.140442	0.141583	0.144798	0.146067	0.14774	0.149774	0.153344	0.155305	0.159562	0.164635	0.171047	0.182659	0.208001	0.208786	0.188326	Orebody
	0.234103	0.22483	0.219479	0.219718	0.2184	0.215044	0.213065	0.214543	0.212562	0.213072	0.210203	0.210791	0.203152	0.193462	0.19001	0.200516	0.212776	0.242175	0.251136	0.376459	Orebody
	0.224958	0.2174	0.21336	0.213383	0.21219	0.211569	0.208348	0.208777	0.208855	0.211842	0.213426	0.203694	0.198685	0.188907	0.179843	0.184178	0.175787	0.15986	0.181305	0.263661	Orebody
	0.183721	0.18016	0.175192	0.175225	0.1741	0.171415	0.16621	0.164686	0.163376	0.16151	0.160154	0.16088	0.159754	0.161837	0.16248	0.170259	0.173843	0.184747	0.212993	0.416315	Orebody
	0.177796	0.174	0.168323	0.168428	0.16712	0.164467	0.162061	0.163293	0.161779	0.163014	0.160932	0.166234	0.167014	0.167115	0.167081	0.172482	0.177916	0.18974	0.20774	0.251058	Orebody
	0.217255	0.21336	0.207656	0.208353	0.20715	0.20307	0.197321	0.192892	0.190307	0.189711	0.184599	0.18582	0.184769	0.178523	0.177527	0.178987	0.181105	0.191371	0.241373	0.256661	Orebody
	0.190087	0.18719	0.181608	0.182174	0.18127	0.178785	0.17754	0.17785	0.17608	0.177797	0.176079	0.181279	0.179682	0.179463	0.179405	0.196066	0.216477	0.278976	0.291235	0.354126	Orebody
	0.176198	0.175	0.170307	0.170802	0.1704	0.168423	0.165761	0.166754	0.16565	0.166461	0.163875	0.164689	0.161368	0.157059	0.149413	0.152159	0.152197	0.156309	0.17783	0.248617	Orebody
	0.169619	0.16658	0.161758	0.161979	0.16162	0.1593	0.155896	0.155227	0.153809	0.152532	0.150346	0.152651	0.15264	0.154074	0.155067	0.161167	0.167866	0.185337	0.235238	0.29887	Orebody
	0.212773	0.20956	0.20389	0.204544	0.20442	0.201979	0.198717	0.19809	0.194023	0.190871	0.18292	0.188897	0.185197	0.181202	0.184356	0.193271	0.211908	0.236287	0.314778	0.330143	Orebody
	0.191304	0.18787	0.183147	0.182881	0.18223	0.180074	0.176262	0.175342	0.174007	0.174352	0.171875	0.172617	0.170176	0.168083	0.16517	0.171405	0.177663	0.191723	0.237213	0.274807	Orebody
	0.192979	0.18665	0.180181	0.180062	0.17893	0.177058	0.17584	0.177439	0.176929	0.179015	0.176939	0.180106	0.177164	0.172133	0.168997	0.174805	0.176898	0.175002	0.207349	0.211493	Orebody
	0.189473	0.18564	0.181354	0.181162	0.18021	0.177713	0.174732	0.17301	0.170468	0.168707	0.165647	0.169426	0.166586	0.163272	0.1626	0.167774	0.168221	0.171447	0.197269	0.214548	Orebody
	0.203281	0.19892	0.193164	0.193337	0.19279	0.188385	0.183565	0.182273	0.179064	0.176253	0.171025	0.171835	0.168957	0.16434	0.163825	0.162664	0.157125	0.14477	0.172497	0.236052	Orebody
	0.161483	0.15905	0.154303	0.154256	0.15392	0.151729	0.14877	0.148697	0.149237	0.150945	0.153274	0.156284	0.156428	0.158963	0.158536	0.165475	0.175901	0.189123	0.229514	0.239741	Orebody
	0.177922	0.16919	0.163218	0.162776	0.16135	0.159941	0.159287	0.162028	0.16344	0.168601	0.174541	0.181869	0.188376	0.194807	0.213116	0.241414	0.298843	0.453913	0.46385	0.277545	Orebody
	0.204237	0.19941	0.194956	0.195225	0.19455	0.190308	0.186987	0.186585	0.182911	0.181234	0.173964	0.175023	0.16876	0.161538	0.157198	0.163639	0.169934	0.179375	0.20794	0.283228	Orebody
	0.212016	0.2076	0.203282	0.203372	0.20297	0.200408	0.193684	0.191965	0.191046	0.190349	0.187409	0.182767	0.177658	0.170653	0.164113	0.172671	0.172771	0.191857	0.220247	0.290079	Orebody

Figure G.3: Analysis 2 Average grade arsenic risk

	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%	
4.034332	4.036127	4.04465	4.081091	4.109458	4.142222	4.159499	4.283375	4.357697	4.479369	4.636037	4.840662	5.013583	5.128474	5.344393	5.507308	5.736988	5.715679	7.031292	7.9131	Orebody	
4.192613	4.196176	4.201164	4.241309	4.272322	4.308344	4.351798	4.462484	4.560564	4.703967	4.929267	5.194855	5.473936	5.69484	6.065906	6.294989	6.796992	7.342029	8.906823	8.6554	Orebody	
3.934444	3.936166	3.94395	3.982988	4.018105	4.072169	4.154663	4.322479	4.436963	4.59525	4.812485	5.055168	5.267669	5.445696	5.771767	5.987133	6.351758	6.779797	7.581408	7.3839	Orebody	
4.88846	4.893749	4.914867	4.967691	5.011702	5.041901	5.071802	5.212278	5.283753	5.417768	5.563834	5.78043	5.876166	5.866741	6.090804	6.356938	6.732407	7.170637	8.065969	10.2585	Orebody	
4.797152	4.801572	4.823458	4.873478	4.91453	4.97307	4.986309	5.106235	5.190409	5.333071	5.535346	5.627679	5.742563	5.726873	5.843221	6.015277	6.084811	5.940666	7.029923	9.0211	Orebody	
4.412638	4.419086	4.431186	4.477427	4.510479	4.540701	4.530863	4.631965	4.706572	4.811723	4.9754	5.180427	5.369338	5.50176	5.738743	5.967021	6.221664	6.384929	7.580385	11.4163	Orebody	
4.28966	4.294578	4.302069	4.347037	4.376986	4.406623	4.431533	4.569535	4.651041	4.799082	4.955509	5.232919	5.437725	5.545311	5.785028	5.981024	6.270559	6.568971	7.682646	8.7314	Orebody	
4.777191	4.789321	4.810917	4.862769	4.89686	4.913123	4.892386	4.976618	5.035916	5.168511	5.270962	5.499978	5.667068	5.683983	5.864935	5.993529	6.236524	6.557226	8.369608	8.8703	Orebody	
4.51128	4.526147	4.548211	4.596932	4.63084	4.658024	4.699063	4.831494	4.901965	5.065495	5.245816	5.525829	5.714232	5.823432	6.049609	6.413116	6.915581	7.807032	8.892362	10.2808	Orebody	
4.406054	4.416005	4.433771	4.481413	4.515205	4.552289	4.571617	4.710485	4.794427	4.942209	5.092988	5.293686	5.434302	5.450385	5.536016	5.660389	5.7659	5.657903	6.669008	8.4893	Orebody	
4.229157	4.236868	4.253287	4.298452	4.334328	4.361989	4.369249	4.480636	4.545207	4.660384	4.808866	5.045215	5.248987	5.375372	5.603342	5.813439	6.401962	6.853882	7.303518	9.5188	Orebody	
4.755952	4.763339	4.782605	4.832777	4.87072	4.898846	4.915693	5.036976	5.085465	5.199437	5.304851	5.593839	5.753491	5.833369	6.148133	6.401962	6.853882	7.303518	9.5188	9.7892	Orebody	
4.379481	4.385179	4.404284	4.445596	4.477888	4.507778	4.526991	4.651534	4.735272	4.880711	5.038839	5.253441	5.440247	5.566364	5.794682	5.999657	6.413202	6.883218	8.537262	8.8983	Orebody	
4.426124	4.430421	4.440621	4.482833	4.515756	4.545876	4.586523	4.733409	4.815809	4.976178	5.134276	5.373836	5.545264	5.79924	5.749933	5.926499	6.078852	5.856195	7.274392	7.5185	Orebody	
4.531608	4.542015	4.563524	4.605734	4.639707	4.668554	4.68297	4.785802	4.845547	4.95713	5.104547	5.347179	5.50732	5.556654	5.788772	5.972643	6.141927	6.174045	7.442654	7.8099	Orebody	
4.673251	4.678437	4.695661	4.743937	4.785382	4.801496	4.795391	4.909988	4.964128	5.068876	5.199153	5.418208	5.583549	5.629646	5.862552	5.913561	5.907067	5.509176	6.738154	8.5048	Orebody	
4.146848	4.150144	4.156102	4.197773	4.231487	4.266486	4.282287	4.388063	4.476271	4.618782	4.821448	5.034285	5.210931	5.344197	5.536906	5.750881	6.037575	6.197732	7.683315	8.2024	Orebody	
4.263676	4.262969	4.267973	4.311004	4.346255	4.388753	4.43423	4.588256	4.688404	4.890782	5.159665	5.467058	5.757877	5.972618	6.44616	6.933161	7.898997	9.919142	11.05715	9.3568	Orebody	
4.6311	4.639199	4.666771	4.714873	4.753297	4.764029	4.773659	4.88445	4.928189	5.02885	5.11566	5.308427	5.422002	5.416067	5.570292	5.777618	6.084967	6.267108	7.557915	9.3351	Orebody	
4.664238	4.667889	4.690159	4.735175	4.774544	4.806591	4.769695	4.869812	4.94048	5.06245	5.20368	5.371922	5.519028	5.543309	5.7062	5.981102	6.15147	6.510839	7.705692	9.1176	Orebody	

Figure G.4: Analysis 2 Average grade copper risk

	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
	30681187	32900721	33915627	34482698	34827750	34489542	33537833	32202214	30877360	28551344	25453649	22013785	17785103	13620924	9876462	6726082	3602192	1523393	655669.1 Orebody 1
	33078487	35299126	36339963	36826259	37121340	36735557	35878465	34165063	32728161	30198806	27252112	23740444	19662974	15341291	11447633	7862583	4389474	2050230	846328.6 Orebody 2
	31994947	34187278	35172890	35686597	36008519	35645758	34634963	33154141	31785678	29258505	26124123	22594581	18317377	14177829	10526130	7198978	3863796	1723211	732340 Orebody 3
	23391500	25659528	26682716	27267000	27694001	27774751	27455831	26640034	26049178	24355810	22423364	19605146	16398061	13220605	10003678	6861359	3701099	1618838	674115.2 Orebody 4
	25827432	28071703	29097997	29627469	29971242	29796089	29366464	28198026	27168408	25188445	22799789	20081988	16625918	13245366	9873908	6786628	3663741	1615796	646745.5 Orebody 5
	28682355	30885861	31946160	32533117	32981655	32776883	32466350	31389964	30373435	28327040	25570978	22368564	18379032	14138486	10389096	7080208	3768037	1582674	632498 Orebody 6
	29047431	31247434	32269390	32803388	33223124	33211898	32507498	31354785	30397462	28383123	25781359	22446878	18458152	14419405	10644729	7258243	3878311	1753235	725812.5 Orebody 7
	23668208	25934177	27031640	27637336	28151418	28211855	28163257	27726266	27080065	25675624	23801606	20867307	17231143	13528900	9937234	6838784	3679166	1687650	696497.8 Orebody 8
	29525482	31745291	32777253	33258430	33645141	33485723	32553877	31268822	30066141	27821072	25343409	21991374	18184414	14162406	10365728	6952371	3671226	1444600	656501.9 Orebody 9
	28018441	30236281	31266980	31803437	32200034	32157813	31745992	30748636	29832675	27920145	25613258	22460495	18521955	14349032	10592262	7200376	3817139	1548044	636011.7 Orebody 10
	28313010	30538127	31587413	32173702	32547769	32496912	32227492	31335769	30149437	28220846	25494154	22271315	18344610	14195243	10426016	7175077	3730098	1589233	637165.1 Orebody 11
	25184856	27393426	28414457	28964362	29382860	29278026	28903758	28000884	27121479	25628396	23599807	20544537	17012661	13156342	9622379	6666208	3335034	1515452	532428.4 Orebody 12
	24985452	27210803	28234273	28847258	29245352	29267728	29096927	28481337	27699708	25994860	23982505	21110795	17633448	14042384	10555774	7198566	4026801	1879732	777528.6 Orebody 13
	24798000	27057334	28113998	28697729	29130944	29076239	28538543	27637359	26754963	24943832	22894726	20023170	16858442	13451737	10122610	6957245	3736678	1598370	675268.9 Orebody 14
	28206109	30417767	31416204	31936269	32358367	32307160	31878537	30924219	29811987	27962265	25681237	22339419	18465726	14418948	10766311	7403049	3974548	1759705	728435.8 Orebody 15
	24966276	27162726	28233295	28826889	29243180	29348752	29212155	28620066	27888513	26499469	24549729	21484395	17892173	14185251	10483470	7132329	3704386	1502563	653622.2 Orebody 16
	30556916	32763949	33756585	34257823	34552763	34140585	33336386	31868057	30505183	27992062	24993046	21409700	17347612	13342830	9701107	6660627	3453771	1491508	627368.9 Orebody 17
	29821259	32079761	33134296	33656759	34037467	33784611	32937734	31665888	30297148	28068107	25242836	21966611	18063235	13896451	10049753	6755064	3455906	1317237	552016.7 Orebody 18
	26162640	28355386	29355161	29886645	30272927	30242213	29668660	28647971	27765915	25820185	23728069	20573064	17333862	13726333	10223498	7007789	3852098	1731439	707930.6 Orebody 19
	22972239	25221742	26262935	26860234	27327926	27402314	27429597	27007028	26268261	249936140	22842808	20624539	17453239	13908645	10526987	7362384	4006667	1801078	737788.4 Orebody 20

Figure G.5: Analysis 2 Economic risk

	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
310675.2	309619.2	306028.8	301171.2	296102.4	29102.4	285331.2	268857.6	243724.8	225350.4	197683.2	166636.8	135379.2	104332.8	77932.8	53856	35692.8	18585.6	8025.6	2745.6	0
306873.6	305606.4	302438.4	297369.6	292300.8	287152.0	281529.6	265900.8	241401.6	223872	196838.4	166003.2	135379.2	104544	78144	54067.2	35904	18585.6	8025.6	2745.6	0
311520	309830.4	305817.6	300748.8	295046.4	288224	284486.4	268224	244147.2	226195.2	198105.6	167270.4	135801.6	104755.2	78144	54067.2	35904	18585.6	8025.6	2745.6	0
273926.4	273292.8	269702.4	265056	259987.2	251116.8	238022.4	216268.8	201273.6	176774.4	150585.6	123552	96729.6	74533.6	52588.8	34848	34848	18163.2	8025.6	2745.6	0
284275.2	282796.8	279206.4	274348.8	269702.4	259564.8	245203.2	222182.4	205497.6	180153.6	151852.8	124608	97363.2	74131.2	51955.2	34425.6	35904	18585.6	8025.6	2745.6	0
299270.4	298425.6	295680	291456	286809.6	276460.8	262944	239289.6	222182.4	195782.4	166003.2	135590.4	104755.2	78144	54067.2	35904	35904	18585.6	8025.6	2745.6	0
300326.4	299270.4	295680	290611.2	285964.8	276672	261676.8	237988.8	220281.6	193670.4	164736	133478.4	103276.8	77721.6	54067.2	35904	35904	18585.6	8025.6	2745.6	0
262944	262099.2	259564.8	255129.6	251328	244780.8	234854.4	216691.2	203385.6	181420.8	155443.2	127353.6	99052.8	75609.6	52800	35059.2	35059.2	18163.2	8025.6	2745.6	0
297158.4	296313.6	292934.4	287865.6	282796.8	273292.8	257875.2	234220.8	217324.8	190924.8	162412.8	131366.4	101587.2	76032	52166.4	34003.2	34003.2	17318.4	6758.4	2745.6	0
297369.6	296313.6	292512	287232	282374.4	272870.4	258720	234854.4	217958.4	191980.8	163257.6	133689.6	103488	77510.4	54067.2	35904	35904	18585.6	8025.6	2745.6	0
303705.6	302649.6	299059.2	294624	289344	278995.2	265267.2	241401.6	223872	197260.8	166636.8	135590.4	104755.2	78144	54067.2	35904	35904	18585.6	8025.6	2745.6	0
285331.2	284486.4	280896	276249.6	271603.2	262099.2	248582.4	226406.4	210566.4	186489.6	160089.6	129676.8	100953.6	75609.6	51955.2	34425.6	34425.6	17107.2	7392	2112	0
285331.2	284275.2	280896	276460.8	272236.8	263366.4	251116.8	230208	215001.6	190080	161568	131788.8	102643.2	76665.6	53644.8	35481.6	35481.6	18585.6	8025.6	2745.6	0
289766.4	288710.4	285331.2	280684.8	276038.4	266323.2	252172.8	228940.8	212044.8	186489.6	158822.4	128832	100320	76876.8	53856	35904	35904	18585.6	8025.6	2745.6	0
294835.2	293990.4	290400	285331.2	280051.2	270336	255974.4	233164.8	216691.2	192192	163468.8	132844.8	102854.4	77299.2	54067.2	35904	35904	18585.6	8025.6	2745.6	0
280473.6	279417.6	276249.6	271392	266112	257241.6	244569.6	222816	207820.8	185222.4	158822.4	129465.6	101376	76454.4	53222.4	35692.8	35692.8	18374.4	8025.6	2745.6	0
311942.4	310886.4	307084.8	301593.6	296102.4	285542.4	269280	244358.4	225984	198105.6	167270.4	135801.6	104544	78144	54067.2	35904	35904	18585.6	8025.6	2745.6	0
304128	303072	299904	295046.4	290188.8	279840	263577.6	239289.6	220915.2	193670.4	162624	131788.8	101587.2	75609.6	51532.8	33369.6	33369.6	16473.6	6124.8	2112	0
295257.6	293990.4	290188.8	285120	280051.2	270547.2	255129.6	231686.4	215212.8	189657.6	161568	131155.2	102854.4	77510.4	53856	35692.8	35692.8	18585.6	8025.6	2745.6	0
274982.4	273715.2	269702.4	264844.8	259987.2	251750.4	240768	220492.8	205497.6	182476.8	155020.8	128620.8	101376	77088	53856	35692.8	35692.8	18585.6	8025.6	2745.6	0

Figure G.6: Analysis 2 Ore Tonnage risk

5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
213100.8	129465.6	89915.2	64627.2	44352	31046.4	20697.6	13516.8	9081.6	5702.4	3801.6	2112	1267.2	211.2	211.2	211.2	0	0	0	0
216902.4	133478.4	92505.6	68428.8	48153.6	34848	23654.4	15840	10560	6547.2	4435.2	2112	1056	0	0	0	0	0	0	0
212256	129254.4	89126.4	65049.6	45408	31891.2	21331.2	13094.4	8236.8	5280	3168	1689.6	844.8	0	0	0	0	0	0	0
249849.6	165792	125241.6	100742.4	80467.2	65260.8	51532.8	40972.8	33158.4	26611.2	19852.8	13939.2	8870.4	3590.4	1478.4	1056	422.4	0	0	0
239500.8	156288	115737.6	91449.6	70752	56812.8	44352	35059.2	28934.4	23232	18585.6	12883.2	8236.8	4012.8	2112	1478.4	422.4	0	0	0
224505.6	140659.2	99264	74342.4	53644.8	39916.8	26611.2	17952	12249.6	7603.2	4435.2	1900.8	844.8	0	0	0	0	0	0	0
223449.6	139814.4	99264	75187.2	54489.6	39705.6	27878.4	19852.8	14150.4	9715.2	5702.4	4012.8	2323.2	422.4	0	0	0	0	0	0
260832	176985.6	135379.2	110668.8	89126.4	71596.8	54700.8	40550.4	31046.4	21964.8	14995.2	10137.6	6547.2	2534.4	1267.2	844.8	422.4	0	0	0
226617.6	142771.2	102009.6	77932.8	57657.6	43084.8	31680	23020.8	17107.2	12460.8	8025.6	6124.8	4012.8	2112	1900.8	1900.8	1267.2	1267.2	0	0
226406.4	142771.2	102432	78566.4	58080	43507.2	30835.2	22387.2	16473.6	11404.8	7180.8	3801.6	2112	633.6	0	0	0	0	0	0
220070.4	136435.2	95884.8	71174.4	51110.4	37382.4	24288	15840	10560	6124.8	3801.6	1900.8	844.8	0	0	0	0	0	0	0
238444.8	154598.4	114048	89548.8	68851.2	54278.4	40972.8	30835.2	23865.6	16896	10348.8	7814.4	4646.4	2534.4	2112	1478.4	1478.4	633.6	633.6	0
238444.8	154809.6	114048	89337.6	68217.6	53011.2	38438.4	27033.6	19430.4	13305.6	8870.4	5702.4	2956.8	1478.4	422.4	422.4	0	0	0	0
234009.6	150374.4	109612.8	85113.6	64416	50054.4	37382.4	28300.8	22387.2	16896	11616	8659.2	5280	1267.2	211.2	0	0	0	0	0
228940.8	145094.4	104544	80467.2	60403.2	46041.6	33580.8	24076.8	17740.8	11193.6	6969.6	4646.4	2745.6	844.8	0	0	0	0	0	0
243302.4	159667.2	118694.4	94406.4	74342.4	59136	44985.6	34425.6	26611.2	18163.2	11616	8025.6	4224	1689.6	844.8	211.2	211.2	0	0	0
211833.6	128198.4	87859.2	64204.8	44352	30835.2	20275.2	12883.2	8448	5280	3168	1689.6	1056	0	0	0	0	0	0	0
219648	136012.8	95040	70752	50265.6	36537.6	25977.6	17952	13516.8	9715.2	7814.4	5702.4	4012.8	2534.4	2534.4	2534.4	2112	1900.8	633.6	0
228518.4	145094.4	104755.2	80678.4	60403.2	45830.4	34425.6	25555.2	19219.2	13728	8870.4	6336	2745.6	633.6	211.2	211.2	0	0	0	0
248793.6	165369.6	125241.6	100953.6	80467.2	64627.2	48787.2	36748.8	28934.4	20908.8	15417.6	8870.4	4224	1056	211.2	211.2	0	0	0	0

Figure G.7: Analysis 2 Waste Tonnage risk



Overview of economic block calculations

1. Calculate the raw block tonnage
Tonnage per Block [tonnes] = Block volume [m^3] * Rock Density [tonnes/ m^3]
2. Calculate the mining cost per tonne
Mining Costs (Cu) [€/tonne] = Mining and Processing Cut-Off (Cu) [€/tonne] – Processing Costs (Cu) [€/tonne]
Mining Costs (Zn) [€/tonne] = Mining and Processing Cut-Off (Zn) [€/tonne] – Processing Costs (Zn) [€/tonne]
3. Calculate the block mining cost
Cost of Mining a Block (Cu) [€] = Mining Costs (Cu) [€/tonne] * Tonnage per Block [tonnes]
Cost of Mining a Block (Zn) [€] = Mining Costs (Zn) [€/tonne] * Tonnage per Block [tonnes]
4. Calculate the recovered metal
Recovered Copper [tonnes] = Copper Grade [%] * Tonnage per Block [tonnes] * Mining Recovery [%] * Processing Recovery (Cu) [%]
Recovered Zinc [tonnes] = Zinc Grade [%] * Tonnage per Block [tonnes] * Mining Recovery [%] * Processing Recovery (Zn) [%]
5. Calculate the metal revenue
Metal Revenue (Cu) [€] = (Metal Price (Cu) [€/tonne] + Refining Costs (Cu) [€/tonne]) * Recovered Copper [tonnes]
Metal Revenue (Zn) [€] = (Metal Price (Zn) [€/tonne] + Refining Costs (Zn) [€/tonne]) * Recovered Zinc [tonnes]
6. Calculate the net block value if milled
Block value if milled (Cu) [€] = (Processing Cost (Cu) [€/tonne] * Tonnage per Block [tonnes]) + Cost of Mining a Block (Cu) [€] + Metal Revenue (Cu) [€]
Block value if milled (Zn) [€] = (Processing Cost (Zn) [€/tonne] * Tonnage per Block [tonnes]) + Cost of Mining a Block (Zn) [€] + Metal Revenue (Zn) [€]
7. Calculate the arsenic penalty
If Arsenic grade < 0.2%
Block value if Milled (Cu) = Block value if Milled (Cu)
Block value if Milled (Zn) = Block value if Milled (Zn)

If Arsenic grade $\geq 0.2\%$ and $\leq 0.5\%$
Block value if Milled (Cu) [€] = Block value if Milled (Cu) [€] – ((Arsenic Grade [%] – 0.2) * (Arsenic Penalty [€/0.01%] * 100))
Block value if Milled (Zn) [€] = Block value if Milled (Zn) [€] – ((Arsenic Grade [%] – 0.2) * (Arsenic Penalty [€/0.01%] * 100))

If Arsenic grade > 0.5%
Block value if Milled (Cu) = -9999999999
Block value if Milled (Zn) = -9999999999

8. Assign block value (decide if Ore / Waste)

Block value (Cu) = Maximum [Block Value of Milled (Cu) [€] or Cost of Mining a Block [€]]

Block value (Zn) = Maximum [Block Value of Milled (Zn) [€] or Cost of Mining a Block [€]]

9. Decide to mine copper ore / zinc ore

Block Value = Maximum [Block Value (Cu) [€] or Block Value (Zn) [€]]