

# Meeting Short-Term Production Targets

Controlling Short-Term Deviations from  
Production Targets by Blending Geological  
Confidence Classes of Reporting Standards

by

M.J. Bijmolt

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Thursday September 22, 2016 at 9:30 AM.

Student number: 4088263  
Project duration: February 1, 2016 – September 1, 2016  
Thesis committee: Prof. dr. ir. M. Buxton, TU Delft, supervisor  
Prof. dr. ir. J. Benndorf, Freiberg University  
Prof. dr. ir. J. D. Jansen, TU Delft  
Prof. dr. ir. M. Rinne, Aalto University  
Prof. dr. ir. B. Lottermoser, RWTH Aachen



# Abstract

Meeting short-term production targets is desired by many companies, since this would enable them to fine-tune the processing operation, meet budget plans and obey contract requirements. Recently stochastic optimization solutions have been developed requiring geostatistical simulations as input. The significant value added has been demonstrated, however, an operational implementation of such approaches for day-to-day use is complex and seems currently difficult as it requires expert knowledge and extensive computational capacity.

To control the short-term deviations, a new fast metaheuristic scheduler is developed that blends Geological Confidence Classes (GCC's) from resource reporting standards. For the scheduler, a new penalty function is developed to schedule for a target blend of GCC's and a new method is developed to enforce smooth mining patterns in three dimensions. The metaheuristic solver uses a Genetic Algorithm and an Ant Colony Optimization algorithm to efficiently converge towards the Pareto optimum. To establish an optimal blend of GCC's, a methodology is developed which creates a range of equally probable scenarios of deviations from production targets for different blends of GCC's. A least-squares estimate can be fitted to these scenarios at the required level of confidence to determine the optimal blend for a maximum allowed deviation.

An historical world class gold deposit is used to show that the monthly and quarterly deviations can be controlled by blending GCC's. Furthermore, the case study shows the possibility to establish an optimal blend of GCC's by using the developed methodology. The scheduler proves to be able to efficiently create and evaluate schedules to blend the GCC's for this case study. For a maximum quarterly deviation of 15% at a 90% confidence level, the established optimal blend is 59% ore tonnage classified as measured resources. For the monthly deviations, a maximum of 15% is too low and cannot be met at a 90% confidence level.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Importance of Meeting Short-Term Production Targets . . . . .	1
1.2 Current Method for Meeting Production Targets . . . . .	1
1.3 Proposed Alternative Approach for Meeting Production Targets . . . . .	2
1.4 Research Goal and Objectives . . . . .	3
1.5 Thesis Outline . . . . .	5
<b>2 A New Fast Scheduler to Blend Geological Confidence Classes</b>	<b>7</b>
2.1 Mine Planning Framework . . . . .	7
2.1.1 Principles of Mine Design . . . . .	7
2.1.2 Principles of Short-Term Mine Planning . . . . .	8
2.1.3 Formulating the Framework . . . . .	9
2.1.4 A New Method to Enforce Smooth Mining Patterns . . . . .	10
2.1.5 A New Method to Meet GCC Target Blends . . . . .	11
2.1.6 A New Fast Method to Validate the Slope Feasibility . . . . .	11
2.2 Metaheuristic Solver . . . . .	14
2.2.1 Principles of Metaheuristics . . . . .	14
2.2.2 Principles of Multi-Objective Optimization . . . . .	15
2.2.3 Principles of Hybridization. . . . .	16
2.2.4 Overview Available Metaheuristic Algorithms . . . . .	16
2.2.5 Establishing the Structure of the New Solver . . . . .	20
2.2.6 Principles of the Genetic Algorithm . . . . .	21
2.2.7 Principles of the Ant Colony Optimization . . . . .	22
2.2.8 Development of the Phases for the New Solver. . . . .	24
<b>3 A Methodology to Establish the Optimal Blend of Geological Confidence Classes</b>	<b>35</b>
3.1 Presenting the Methodology . . . . .	35
3.2 Development of the Artificial Drilling Campaign Algorithm. . . . .	39
3.3 Principles of Resource Simulation and Estimation . . . . .	41
3.3.1 Ordinary Kriging . . . . .	41
3.3.2 Sequential Gaussian Simulation . . . . .	43
3.3.3 Comparing Ordinary Kriging and the E-type Estimator . . . . .	43
<b>4 Case Study</b>	<b>45</b>
4.1 An Introduction to the Yanacocha Case . . . . .	45
4.2 Step 1 - Creating Artificial Drilling Campaigns . . . . .	46
4.3 Step 2 - Performing Ordinary Kriging and Sequential Gaussian Simulations . . . . .	46
4.4 Step 3 - Creating the True Resource Model . . . . .	51
4.5 Step 4-7 - Establishing Relation Between GCC Blends and Deviations. . . . .	51
4.6 Step 8-10 - Establishing the Optimal Blend of GCC's . . . . .	51
<b>5 Discussion</b>	<b>57</b>
<b>6 Conclusions and Recommendations</b>	<b>61</b>
<b>A Nomenclature</b>	<b>63</b>
<b>Bibliography</b>	<b>65</b>



# Introduction

## 1.1. Importance of Meeting Short-Term Production Targets

Short-term production targets are set by mining companies to optimize the processing plant, create budget plans and to meet contract requirements. Not meeting these targets result in budget problems, under-performance of the processing plant and possibly penalties from the smelter. Every company has a different risk profile, defining the acceptable deviations. Deviations exceeding this acceptance level are to be prevented.

Tatman (2001) showed that 35 per cent of the 60 investigated mines did not achieve the planned production. Furthermore, Vallee (2000) discovered that 60% of the mines surveyed had an average rate of production less than 70% of the designed capacity. Many authors relate these deviations to the geological uncertainty not taking into account during mine planning (Dowd et al., 2014; Godoy and Dimitrakopoulos, 2004; Dimitrakopoulos and Godoy, 2012; Benndorf and Dimitrakopoulos, 2009; Vann et al., 2012). The problems related to the geological uncertainty arise mainly on a short-term bases, resulting in a high grade variability throughout the weeks or months, affecting the processing operation and the quality demands (Dowd et al., 2014; Leite and Dimitrakopoulos, 2009; Tavchandijan et al., 2004).

## 1.2. Current Method for Meeting Production Targets

The current theoretical concept to include the geological uncertainty in mine planning is Simulation Based Optimization (SBO) and uses conditional simulations to create multiple, equally probable resource models, which are then used in a Stochastic Integer Programming (SIP) tool to minimize deviations from production targets by considering all the equally probable scenarios instead of one estimation. This technique is able to effectively control the deviations from production target (Benndorf and Dimitrakopoulos, 2013; Dimitrakopoulos and Godoy, 2012; Tavchandijan et al., 2004; Ramazan and Dimitrakopoulos, 2004; Leite and Dimitrakopoulos, 2009; Benndorf and Dimitrakopoulos, 2009; Jewbali and Dimitrakopoulos, 2009; Goodfellow and Dimitrakopoulos, 2014; Spleit, 2014).

The SIP tools use either Mixed-Integer-Linear-Programming (MILP), meta-heuristics or a combination of these two to optimize schedules. MILP uses commercial solvers to find the optimal solution. However, these solvers are limited in the amount of variables, especially integer variables, hereby requiring complex aggregation strategies to solve the problem at once (Froyland et al., 2004; Stone et al., 2004; Menabde et al., 2004; Dejonghe and Boni, 2006). Meta-heuristics are techniques which employ some degree of randomness to find the as optimal solution as possible to hard problems, of which there is no principled way of finding the optimal solution and limited heuristic information to go on. The advantage of using meta-heuristics is that there is no limit in the amount of variables, however, the solution given may not be the global optimum (Luke, 2015).

The significant value added of SBO has been demonstrated, however, an operational implementation of such approaches for day-to-day use is complex and seems currently difficult as it requires expert knowledge and extensive computational capacity. Furthermore, conventional estimation techniques, mainly in the form of Ordinary Kriging, are still used by many mining companies, instead of the more advanced simulation techniques. Hereby eliminating the possibility to use SBO to control the deviations from production target (Godoy, M. – personal communication, Jan. 2015). To bridge theory and practical application, a new approach is needed to meet production targets.

### 1.3. Proposed Alternative Approach for Meeting Production Targets

Geological Confidence Classes (GCL) are instruments of various legally enforced Reporting Standards for stock-listed mining companies, which are used to classify the geological uncertainty. This feature makes these GCC's an ideal, widely available instrument to control the deviations from production target. An introduction to the standards is given first, followed by a discussion of the GCC's and the proposed method to use the GCC's in mine planning.

#### History of Reporting Standards

In the 1960s, a nickel boom in Western Australia resulted in many Resource Reports, describing the findings and estimations of the prosperous Nickel deposits to raise equity on the stock market. However, many of these reports were of unacceptable quality and were misleading the investor (Creech, 2012). This has led to the establishment of the Australasian Joint Ore Reserves Committee (JORC) in 1971 to develop guidelines for stock-listed companies reporting ore reserves (AusIMM, 2012). The JORC published the first standard for resource reporting in 1989. Many countries followed this initiative and created their own standards, which were internationally standardized for many points in 1997 (CRIRSCO, 2013).

#### Mineral Resources and Reserves

The purpose of the Reporting Standards is to enforce the reporting of the certainty of resources and reserves statements, hereby protecting the investors. The Standards differentiate resources and reserves as follows (CRIRSCO, 2013):

**Resource**      *'A concentration of naturally occurring solid, liquid or gaseous materials in or on the Earth's crust in such a form that economic extraction of a commodity is currently or potentially feasible.'*

**Reserve**        *'That portion of the identified resource from which a usable mineral and energy commodity can be economically and legally extracted at the time of determination.'*

#### Geological Confidence Classes

The resources and reserves are divided into different confidence categories. The Geological Confidence Classes define the resource confidence categories and are in order of increasing confidence: inferred, indicated and measured (CRIRSCO, 2013). **Inferred Resources** are estimated on the basis of **limited** geological evidence and sampling, which is sufficient to **imply**, but not verify geological and grade or quality continuity. **Indicated Resources** are derived from **adequately** detailed and reliable exploration, sampling and testing and is sufficient to **assume** geological and grade or quality continuity. **Measured Resources** are the highest confidence level and are derived from detailed and reliable exploration, sampling and testing and is sufficient to **confirm** geological and grade or quality continuity. The measured and indicated resource categories can be converted to reserves. The conversion considers the entire mining operations and its uncertainties and is performed in feasibility studies. Inferred resources are not allowed to be converted to reserves by the Reporting Standards, because of its high geological uncertainty.

The confidence classes are usually assessed according to the drill-hole spacing relative to the range of influence. However, the determination should also include aspects such as the confidence in the geological interpretation and the amount, distribution and quality of the data (Stephenson et al., 2012; Vann et al., 2012).



For example, the Mineral Resource estimate for a nugget gold deposit may never attain the same measure of relative confidence as a stratiform-style copper deposit (Noppe, 2014).

The final decision to include a portion of the mineralization to the Inferred, Indicated or Measured Resource class is made by the competent person. The Competent Person is a suitably qualified and experienced persons who is subject to an enforceable professional code of ethics and rules of conduct (CRIRSCO, 2013). It uses its experience for the type of mineralization to assess the uncertainty given all the available information about the deposit.

If the appropriate classification is made, the stakeholders generally want to know the numerical expected accuracy and confidence level over a certain time. The Reporting Standards do not enforce the use of these numbers, however, do encourage competent persons to discuss and if possible provide at least a qualitative discussion on the relative uncertainties (Noppe, 2014). For the numerical accuracies and confidences, Noppe (2012) and Glacken and Trueman (2012) suggest the following:

**Measured Resource** +/-10-15% at 90% confidence level for three monthly production scales

**Indicated Resource** +/-10-15% at 90% confidence level for annual production scale scales

The accuracies mean that, for the measured resource class, if mined for a quarter long period, the actual production will be between 85% and 115% of the target production for nine out of 10 quarterly periods. For one quarterly period in every 10 quarterly periods, the deviations will be higher than 115% or lower than 85%. It should be noted that the accuracy and confidence level are dependent on the company's risk profile. Meaning that if investors prefer a higher or lower risk profile, these values might be different. The Reporting Standards only ensure that it is clear for the investor how confident the company is about its accuracies.

### **Proposed Method**

The GCC's are assigned block-wise by the Competent Person to report the amount of resources per category. Therefore, these categories form a readily available source of information which can be used by a scheduler that plans the mining period of resource blocks. The idea is to blend the blocks assigned with a measured or indicated resource class in a certain ratio to control the deviations from production target. A blend of these resource classes is commonly available since not all planned drillholes for measured resources succeed. The belief is that when in a period only indicated resources are planned, the uncertainties will be much higher than when a combination of measured and indicated resources are planned. The blend of geological confidence classes is defined as the percentage of ore tonnage classified as measured and indicated resources. The blend is determined by the processing feed because it is the deviation in metal production that is important to optimize the processing plant, meet budget plans and obey contract requirements.

## **1.4. Research Goal and Objectives**

The goal and hypothesis of this research are defined below.

Goal:

**Control the monthly and quarterly deviations from production target by blending Geological Confidence Classes on a monthly basis**

Hypothesis:

**1. The absolute relative deviations from production targets can be controlled by scheduling for monthly blends of Geological Confidence Classes**

**2. It is possible to establish an optimal monthly blend of Geological Confidence Classes to limit the deviations**

The research objectives defined to evaluate the hypothesis are given below.

### **Objective 1 - Development of a Scheduler**

The scheduler consists of two components, namely the mine planning framework and the solver to optimize for the given planning framework.

- 1.1 Develop a mine planning framework for an open pit mine that meets typical mining constraints, while optimizing for the production targets, a smooth mining pattern and a GCC target blend.
- 1.2 Develop a metaheuristic solver that can optimize the mine planning objectives and constraints. Metaheuristics are chosen because it can efficiently handle large amounts of binary variables. The solver should be able to produce acceptable feasible solutions within minutes and should not be site limited.

### **Objective 2 - Development of the Methodology to Establish the Optimal Blend of GCC's**

The methodology is able to evaluate the produced schedules and to establish an optimal blend of measured and indicated resource classes for a maximum allowed deviation at a certain confidence level. The procedure requires the existence of an exhaustive data set. The idea behind the procedure is to first create resource models from the exhaustive data set that contain different blends of geological confidence classes in order to schedule for different target blends. The production figures from the schedules are then compared against the actual production figures by applying the schedules on the exhaustive data set. These deviations are then compared against the blend of GCC's to evaluate the potential relation. Geostatistical simulation techniques are used to establish a range of equally probable deviation scenarios, from which an optimal blend of GCC's can be established by using simple regression techniques on a confidence level of the scenarios. Since this method is designed to overcome the need of simulation techniques, the optimal blend is verified by schedules made for resource models created using Ordinary Kriging. The research objectives defined for this procedure are shown below.

- 2.1 Develop an artificial drilling campaign algorithm that can create drilling campaigns from an exhaustive data set, given a certain GCC target blend. The algorithm relies on the classification of the GCC's in terms of the drillhole spacing, which it uses to extract data points from an exhaustive data set in such a way, that the target blend is met.
- 2.2 Create resource models for the different campaigns using the conventional ordinary kriging method and a simulation method. The simulation methods are used to create a range of equally probable scenarios for the resource model, which are later used to create a range of equally probable deviation scenarios. The kriging resources reflect the widely used industry approach for creating the resource model.
- 2.3 Schedule for different GCC target blends using the kriging and E-type estimate. The simulated resource models are converted back to an estimate by averaging the model block-wise, which is called the E-type estimate. The E-type schedule is then used to determine the range of equally probable production scenarios from the simulated resource models.
- 2.4 Calculate a range of equally probable absolute relative monthly and quarterly deviations per GCC blend, using the exhaustive data set as the true representation of the deposit. The quarterly deviations are calculated by averaging three monthly production figures and comparing it against the exhaustive data set. The deviations per blend of GCC's show the possibility to control the deviations by blending GCC's.
- 2.5 Evaluate the possibility to establish an optimal GCC blend for a maximum allowed monthly and quarterly deviation. The optimal blend is determined by fitting a least-squares estimation to a confidence level of the deviation scenarios and determine the intersection with the required maximum allowed deviation.
- 2.6 Verify the optimal blend by calculating the deviations of the schedules created with the kriged resource model. The percentage of these deviations that meet the optimal blend of GCC's and the maximum allowed deviation indicate whether the optimal blends also holds for the kriged resource models.

## 1.5. Thesis Outline

- Introduction** addresses the need for a new method for meeting short-term production targets. The chapter provides a brief introduction to Geological Confidence Classes and their background and introduces the idea of monthly blending these Geological Confidence Classes to control monthly and quarterly deviations from production targets. Furthermore, it provides the research goal and objectives.
- Scheduler Development** presents a new fast scheduler developed for this research in terms of the planning framework and the metaheuristic solver. The planning framework presents four objective functions, namely minimizing deviations from the production target, minimizing deviations from the target blend and minimizing the smoothing penalty. The constraint function introduced are the slope constraint, the capacity constraints and the reserve constraint. A three phase meta-heuristic approach is developed, consisting of a heuristic diversification stage, an exploration phase with a Genetic Algorithm and an exploitation phase consisting of an Ant Colony Optimization. The Pareto prevalence algorithm is introduced to optimize for the multi-objective optimization problem.
- Methodology** introduces the method to establish the optimum blend of GCC's for a maximum allowed deviation at a particular confidence level. The method consists of 10 steps and requires the existence of an exhaustive data set of a representative area of a deposit. The first steps consists of creating artificial drilling campaigns from the exhaustive data set. From these campaigns, resource models are created using Sequential Gaussian Simulation and Ordinary Kriging in step 2. The true representation of the deposit is estimated using the E-type estimator in step 4 based on the exhaustive data set. The Ordinary Kriging and E-type estimator are then scheduled and the production figures and the realized GCC blends per period are determined in step 4,5 and 6. With the true representation of the deposit, the deviations are calculated and the confidence level is retrieved from the scenarios in step 7 and 8. Step 9 consists of establishing the optimum ratio by using least-squares estimation, which are verified by the schedules created using ordinary kriging in step 10.
- Case Study** presents the results of the Yanacocha case study, showing the drilling campaigns created for this case study and the process of creating resource models from these drilling campaigns using ordinary kriging and sequential Gaussian simulations. The scheduling results for the simulations are shown as boxplots of the absolute relative deviation as a function of the ratio of ore tonnage classified as measured resources. A maximum allowed deviation of 15% at 90% confidence level is set for this case to show the methodology to establish an optimal GCC blend.
- Discussion** discusses the Yanacocha case study results and evaluates the scheduler and the methodology. It starts with the shortcomings of the drilling campaign algorithm and a discussion of the resource modelling results and the performance of the scheduler. Controlling the deviations by blending the GCC's for the case study is discussed next, followed by a discussion on the established optimal GCC blend for a maximum allowed deviation of 15% at a 90% confidence level. The chapter closes with an evaluation of the applicability of the methodology.
- Conclusion** provides the conclusions and recommendation of this research, summarizing the discussion and placing it in a larger context. The recommendation show potential follow-up studies that can be undertaken to built upon the knowledge gained during this research.



# 2

## A New Fast Scheduler to Blend Geological Confidence Classes

The task of the scheduler is to create feasible mine plans that are optimized for a set of objectives. The mine planning framework section provides the objectives and the formulations to guarantee feasibility, called constraints. The actual optimization is performed by the solver, discussed and developed in the corresponding section.

### 2.1. Mine Planning Framework

To develop the planning framework, a discussion about the mine design and the commonly found constraints and objective functions for short-term mine planning is given first based on a literature review. A new method to enforce smooth mining patterns is discussed next, followed by the introduction of a new method to optimize for a target blend of GCC's. The framework is then presented in terms of the objective functions and constraints. For the constraints, a new efficient method to evaluate the slope constraint is discussed. Appendix A provides the nomenclature for the formulations.

#### 2.1.1. Principles of Mine Design

In order to schedule for an open pit mine, the design of an open pit is to be considered first. The pit design relies on the following factors (Newman et al., 2010):

1. An orebody model in which the deposit is discretized into a grid of blocks, each of which contain the mineral properties and tonnages of the respective locations.
2. A geometric model of the deposits

The block size is an important parameter in the mine design. The smallest possible block size that can be selected is the Smallest Mineable Unit (SMU), which is the smallest size that the mining equipment and method can selectively mine. The block size chosen should be linked to the available drillhole spacing and not be made too small, which could give a false sense of accuracy. The block size should be between  $\frac{1}{2}$  and  $\frac{1}{3}$  of the drill hole spacing (Reid, 2015). A detailed discussion on the block size and its influence on dilution and economic impact is not given here, the interested reader is referred to (Jara et al., 2006).

The design of a pit, together with the properties of the pit, are shown in Figure 2.1. The pit geometry is controlled by the following items (Atlas-Copco, 2012):

- Ore body shape
- Grade distribution
- Stability of the slopes
- Need to provide access

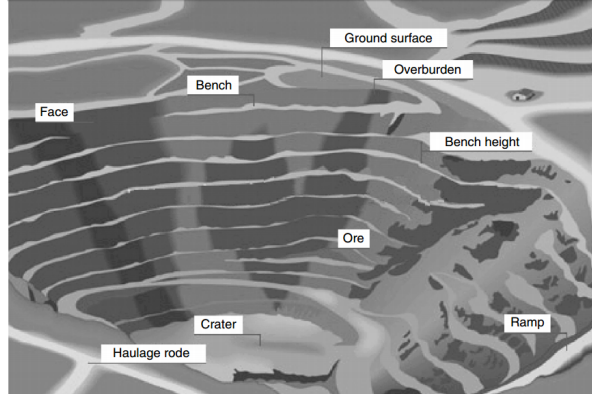


Figure 2.1: Design of an open pit with its properties (Newman et al., 2010)

- Operating consideration

The geometric constraints applicable to the design of a mine are twofold (Newman et al., 2010):

1. Ensuring stable walls
2. Ensuring that the equipment can access the pit

The stability is commonly ensured by including a slope angle in the mine planning phase, which is later manually translated to the benches and faces as shown in Figure 2.1. Furthermore, the ramp is also commonly designed manually after the mine planning exercise. It is important to correctly determine the slope angle, since it has a dramatic effect on the tonnage mined and the ratio of ore to waste. The slope angle is represented by the block model, inevitable resulting in approximation errors (Hall, 2009). The determination of the appropriate slope angle is not discussed here. The interested reader is referred to Fleurissen (2012).

The ore is separated from the waste using a cut-off grade (Newman et al., 2010). In this research, the cut-off grade is determined based on the idea that the revenue created by selling the concentrate should cover the mining and processing costs. This is shown in equation 2.1, in which  $N_{mine}$  represents the totality of blocks in the pit which are to be mined. Therefore, the method requires a preliminary mine design which describes which blocks are to be mined. Solving for  $N_{select}$  in equation 2.1 using equation 2.2 and 2.3 gives the cut-off grade. More sophisticated and dynamic cut-off grade methods exist, but are not discussed here. The interested reader is referred to Rendu (2009).

$$\sum_{n=1}^{N_{select}} g_n \cdot R \cdot P - N_{select} \cdot E_{proc} - N_{mine} \cdot E_{mine} = 0 \quad (2.1)$$

$$N_{select} = \sum_{n=1}^N BOW_n(g_c) \quad (2.2)$$

$$BOW_n(g_c) = \begin{cases} 1 & \forall g_n \geq g_c \\ 0 & else \end{cases} \quad (2.3)$$

### 2.1.2. Principles of Short-Term Mine Planning

Mine scheduling can be categorised into three main time phases, of which only the short-term case is discussed here (King, 2014):

1. Strategic – life of mine
2. Long-term – in years
3. Short-term – in days to months

All scheduling tasks aim to define the best mine plan subject to the constraints imposed by the physical and geological conditions, policies and operational mining approach, for which the term best is to be defined by the management (Benndorf and Dimitrakopoulos, 2009). The tasks of mine planning are as follows ((Dagdelen, 2004):

- Whether a given block in the model should be mined or not
- If it is to be mined, when it should be mined
- Once it is mined, how it should then be processed

The created schedule should guarantee a safe and feasible operation while meeting the predefined production targets. A safe operation is provided by a slope constraint, enforcing a block configuration that obeys a maximum slope angle for a certain underlying block (Xu 2009). A feasible schedule is enforced by the reserve constraint, enforcing a block to be mined only once throughout the mine life. Furthermore, capacity constraints are used to obey the mining and processing capacities (Xu 2009). To improve the feasibility of the schedule, a smoothing penalty is applied by some authors which penalizes not mining adjacent blocks in current or previous periods of the block under consideration (Ramazan and Dimitrakopoulos, 2004).

Meeting production targets is achieved by assigning penalties when the target is not met (Benndorf and Dimitrakopoulos, 2009). Several authors implement a geological discount factor for the penalties to control the risk over time, resulting in a schedule that is less risky in early periods than in later periods. The idea is that in later periods, more information is available, therefore, making these risks manageable. The discount rates used are between 8 and 20 per cent (Benndorf and Dimitrakopoulos, 2013; Ramazan and Dimitrakopoulos, 2004; Leite and Dimitrakopoulos, 2009; Jewbali and Dimitrakopoulos, 2009; Spleit, 2014).

Once the short-term schedule is in production, grade control drilling is applied to accurately evaluate the grade of the blocks. Based on this information, the destination of the block can be changed, from the waste plant to the processing plant if the actual grade is above the cut-off grade, or vice versa (Dimitrakopoulos and Godoy, 2012). Misclassifying a block as ore or waste always results in a net loss, which is largely attributable to the lack of perfect knowledge about real grade distributions. To take into account these losses, economic classification functions exist, which describe the costs of misclassifying a block. These functions are not discussed here, the interested reader is referred to Dimitrakopoulos and Godoy (2012).

### 2.1.3. Formulating the Framework

The scheduler has to produce schedules, describing whether a block is mined in a certain period or not. The blocks are gathered in a blockmodel, which forms the input of the scheduler. The blockmodel contains information about the blocks, such as the density and grade. This information is used to create optimal mining schedules. The optimality is determined based on objective function, while taking into account the constraints.

The idea behind the framework is to meet a predefined GCC target ratio, while at the same time optimizing for a target production and maximizing the feasibility of the schedule. Optimizing the GCC target ratio is done in an objective function, which minimizes the penalty reflecting the deviation from the GCC target. The target production is on the one hand optimized by an objective function if the production from a schedule is below the production capacity. This deviation is discounted in time by the geological discount factor. The idea behind implementing a geological discount rate is that throughout the mining operation, more information becomes available, which might result in an updated mine plan. Therefore, making it more important to meet the production targets in the earlier periods than in later periods. On the other hand, the maximum capacity is enforced by using a hard constraint. The production is divided into mining and processing. The processing target is divided into a feed target and a metal production target. The hard constraint for both the mining and the processing operation are only aimed at the maximum allowed throughput in tonnage. Maximizing the feasibility is done by minimizing a smoothing penalty. Furthermore, a slope and reserve constraint are added which are enforced to be obeyed. The objective function and constraints are discussed on the next page.

### Objectives

- min(Mining Deviations)** The deviations from the target mining capacity are discounted by the geological discount factor and summed, as shown in formula 2.5. The idea behind the formula is to sum the block mass per period, with  $X_{n,t}$  the binary variable to mine block  $n$  in period  $t$  (1) or not (0),  $BD_n$  the density of block  $n$  and  $V_B$  the volume of a block. If block  $n$  is not mined in period  $t$ , the block contributes no mass to that period due to the binary variable. The summed block mass per period is then compared against the mining capacity of that period,  $C_{mine,t}$ . The absolute difference is discounted by  $d_{geo}$  to ensure that differences in capacity is penalized more heavily in earlier periods than in later periods.
- min(Processing Deviations)** The deviations from the target processing capacity in terms of metal production and ore feed are discounted by the geological discount factor and summed, as shown in formula 2.6 and 2.7. The formulas work in similar fashion as formula 2.5, except that a difference is made between ore and waste by the binary parameter  $BOW_n$ , which is 1 if block  $n$  is ore and 0 if block  $n$  is waste. The metal production is calculated by considering all ore blocks and their grade,  $BG_n$ . If the block is waste, there is not metal contribution.
- min(Smoothing Penalty)** Smooth mining patterns are enforced by the implementation of the smoothing penalties as discussed in section 2.1.4.
- min(Blending Penalty)** The blend of Geological Confidence Classes is penalized if it deviates from a predefined GCC ratio as discussed in section 2.1.5.

### Constraints

- Slope Angle** The slope angle constraint is implemented using a new fast evaluation method, as discussed in section 2.1.6.
- Reserve Constraint** The reserve constraint enforces that a block can only be mined once throughout the life of the mine. Formula 2.14 shows the reserve constraint. The idea behind the formula is to sum  $X_{n,t}$  for all periods and verify if the block is not mined more than once. If the block is not mined in all periods, the sum equals 0 and the constraint is met. If the block is mined once, the sum equals 1 and the constraint is also met. However, if the blocks is mined more than once throughout the mine life, the sum equals more than 1 and the constraint is not met.
- Capacity Constraint** For all periods, the maximum processing and mining capacity should be obeyed. These are enforced by formula 2.12 and 2.13. These formulas work the same as formula 2.5, 2.6 and 2.7, except that here the schedule is evaluated per period to verify if the maximum capacity is exceeded.

#### 2.1.4. A New Method to Enforce Smooth Mining Patterns

The current method to enforce smooth mining patterns involves a horizontal rectangle surrounding a mined block (Benndorf and Dimitrakopoulos, 2013). The rectangle is made up of zones containing penalty weight for not mining the surrounding blocks in the same period. The penalties for all blocks are summed and provide an indication of the smoothness of the schedule. The objective function used for this purpose is shown in equation 2.8. The idea behind the formula is to determine the maximum penalty by  $A \cdot N_{nb1,n} + B \cdot N_{nb2,n}$ , in which  $N_{nb1}$  is the number of blocks in the inner circle and  $N_{nb2}$  the number of blocks in the outer circle, with the respective penalty weights of A and B. This penalty is reduced by considering the number of surrounding blocks that are mined in the same period as the centre block. If all surrounding blocks are mined in



the same period, the penalty is zero, while if no surrounding blocks are mined in the same period, the penalty equals the maximum penalty. The penalty is discounted in time, to ensure that earlier periods receive a higher penalty for not meeting the smoothing objective. No penalty is given if the surrounding block is not mined in period  $t$ , by multiplying the entire function by the binary variable  $X_{n,t}$ .

The drawback of this method is that it is biased for horizontal layers given the horizontal rectangle. Therefore, a 3-dimensional cube is introduced in this research to enforce unbiased smooth mining patterns. The cross sections of the cube are shown in figure 2.3a and 2.3b, which contain the penalty weights. The additional advantage of this method is that the same objective function can be used as for the original approach. Only  $nb1$  and  $nb2$  need to be expanded to include all surrounding blocks instead of only in 2 dimensions.

### 2.1.5. A New Method to Meet GCC Target Blends

The idea is to penalize schedules which do not meet a set GCC target ratio. For this, the ore feed per period is calculated for the measured and indicated class separately. This provides the actual ratio of the measured and indicated class in terms of mined ore in a certain period. The difference from this ratio and the target ratio is squared and summed for both classes and all periods to provide an indication on how well the schedule is meeting the target GCC ratio. The squared differences are shown in equation 2.10 and 2.11, while the objective function for meeting the GCC target ratio is shown in equation 2.9. The idea behind the first two formulas is to determine the amount of ore blocks mined in period  $t$  and translate this to a tonnage using the density  $BD_n$  and the volume  $V_B$  of these blocks. The percentage of measured and indicated classified ore tonnage is calculated using  $M_n$  and  $IND_N$ , which are 1 if block  $n$  is classified as measured and indicated, respectively, or 0, if not. The ratio is compared against the predefined ratio,  $R_m$  and  $R_{ind}$ , which are to be met. To ensure that negative deviations are treated equally as positive deviations, the deviations are powered by  $P_{gcl}$ . The stronger the power factor, the more large deviations are penalized. The penalties for all periods and for all classifications are summed to create one penalty value for a specific schedule.

The GCC are assigned based on a classification scheme, which assigns the measured, indicated or inferred class to a block based on the euclidean distance of the block to the nearest drillhole and the classification of that nearest drillhole. Meaning that if the drillhole is part of the inferred class, the block is also classified as being inferred, disregarding the distance to the drillhole. If the drillhole is classified as indicated, then the block is classified as indicated if its euclidian distance is not greater than the maximum allowed drillhole spacing for classifying indicated resource classes. If the distance is greater, the block is classified as inferred. If the drillhole is measured and the euclidean distance is not greater than the maximum allowed drillhole spacing for classifying measured resource classes, the block is classified as measured. If the distance is greater, but less than the maximum allowed drillhole spacing for classifying indicated resource classes, the block is classified as indicated. Else, the block is classified as inferred.

### 2.1.6. A New Fast Method to Validate the Slope Feasibility

The schedule should obey the maximum allowed slope angle for all periods. The slope angle within a block-model is determined as shown in figure 2.2. The equation of the slope angle is given by formula 2.4, in which  $\alpha$  is the slope angle,  $D$  the maximum depth and  $L$  the smallest horizontal length to reach the maximum depth. In order to quickly determine the slope feasibility, the block dependencies are determined first, which describes blocks that should be mined in order to mine an underlying block. The idea behind the method to determine the feasibility is to use a matrix multiplication between the schedule and the block dependencies to provide the amount of overlying blocks mined and comparing this with the required amount of overlying blocks per mined block. If not enough overlying blocks are mined, the slope feasibility is not met. The method is quick because it does not require computationally expensive for-loops. The actual procedure is slightly more complex due to the fact that some block cannot be mined if it is the underlying block, but can be mined if part of a larger pit. To take this into account, the blocks that are part of a larger pit are not considered individually as an underlying block. The procedure to determine the block dependencies is shown in algorithm 1, while the fast feasibility check is shown in algorithm 2.

$$\alpha = \text{TAN}^{-1}\left(\frac{D}{L}\right) \quad (2.4)$$

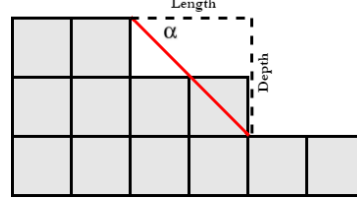


Figure 2.2: Example of how the slope angle is determined

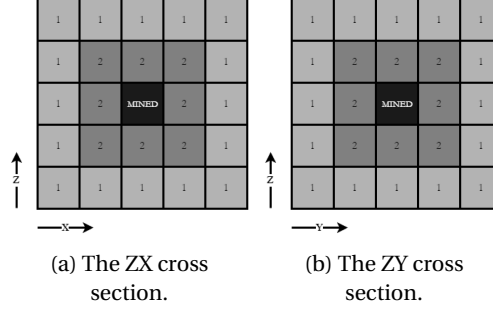


Figure 2.3: Cross sections of the smoothing cube. The numbers represent the penalty weights.

$$Y_{mine} = \sum_{t=1}^T \frac{ABS[\sum_{n=1}^N [X_{n,t} \cdot BD_n \cdot V_B] - C_{mine,t}]}{(1 + d_{geo})^t} \quad (2.5)$$

$$Y_{proc}^{ore} = \sum_{t=1}^T \frac{ABS[\sum_{n=1}^N [BOW_n \cdot X_{n,t} \cdot BD_n \cdot V_B] - C_{proc,t}^{ore}]}{(1 + d_{geo})^t} \quad (2.6)$$

$$Y_{proc}^{metal} = \sum_{t=1}^T \frac{ABS[\sum_{n=1}^N [BOW_n \cdot X_{n,t} \cdot BD_n \cdot V_B \cdot BG_n \cdot R] - C_{proc,t}^{metal}]}{(1 + d_{geo})^t} \quad (2.7)$$

$$Y_{smoothing}^{n,t} = \frac{[A \cdot N_{nb1,n} + B \cdot N_{nb2,n}] \cdot X_{n,t} - A \cdot \sum_{r \in nb1} X_{r,t} - B \cdot \sum_{r \in nb2} X_{r,t}}{(1 + d_{geo})^t} \cdot X_{n,t} \quad (2.8)$$

$$Y_{confidence} = \sum_{t=1}^T Y_{measured}^t + Y_{indicated}^t \quad (2.9)$$

$$Y_{measured}^t = \left[ \frac{\sum_{n=1}^N X_{n,t} \cdot M_n \cdot BOW_n \cdot V_B \cdot BD_n}{\sum_{n=1}^N X_{n,t} \cdot BOW_n \cdot V_B \cdot BD_n} - R_m \right]^{P_{gcl}} \quad (2.10)$$

$$Y_{indicated}^t = \left[ \frac{\sum_{n=1}^N X_{n,t} \cdot IND_n \cdot BOW_n \cdot V_B \cdot BD_n}{\sum_{n=1}^N X_{n,t} \cdot BOW_n \cdot V_B \cdot BD_n} - R_{ind} \right]^{P_{gcl}} \quad (2.11)$$

$$\sum_{n=1}^N X_{n,t} \cdot V_B \cdot BD_n \leq C_{mine,t} \quad \forall t \quad (2.12)$$

$$\sum_{n=1}^N BOW_n \cdot X_{n,t} \cdot V_B \cdot BD_n \leq C_{proc,t} \quad \forall t \quad (2.13)$$

$$\sum_{t=1}^T X_{n,t} \leq 1 \quad \forall n \quad (2.14)$$

**Algorithm 1** Creating the block dependencies for the slope angle

---

```

1 procedure CONSTRUCTDEPENDENCIES
2   for all  $n \in N$  do
3      $Z_{\text{overlying}} \leftarrow \text{DETERMINEOVERLYINGLAYERS}(n)$ 
4     for all  $z \in Z_{\text{overlying},n}$  do
5        $Radius \leftarrow \frac{z \cdot S_z}{\text{TAN}(\alpha)}$ 
6       if  $Radius + BX_n > O_x + N_x * S_x$  then
7          $NBL[n] \leftarrow \infty$ 
8         break
9       if  $BX_n - Radius < O_x$  then
10         $NBL[n] \leftarrow \infty$ 
11        break
12       if  $Radius + BY_n > O_y + N_y * S_y$  then
13         $NBL[n] \leftarrow \infty$ 
14        break
15       if  $BY_n - Radius < O_y$  then
16         $NBL[n] \leftarrow \infty$ 
17        break
18       for all  $n_z \in z$  do
19          $Distance \leftarrow \sqrt{[BX_{n_z} - BX_n]^2 + [BY_{n_z} - BY_n]^2}$ 
20         if  $Distance \leq Radius$  then
21            $BL[n, n_z] \leftarrow 1$ 
22            $NBL[n] \leftarrow NBL[n] + 1$ 

```

---

**Algorithm 2** Verify the slope feasibility

---

```

1 procedure VERIFYSLOPEFEAS(Chrome)
2   for all  $t \in T$  do
3      $tmp \leftarrow NBL$ 
4      $CM \leftarrow \sum_{r=1}^t Chrome$ 
5      $LP \leftarrow \text{DOTPRODUCT}(BL, CM)$ 
6      $ILP \leftarrow \text{NONZERO}(LP)$ 
7      $tmp[ILP] \leftarrow 0$ 
8      $CrNBL \leftarrow \text{MULTIPLY}(Chrome[:, t], tmp)$ 
9      $CrBL \leftarrow \text{DOTPRODUCT}(BL, Chrome[:, : t])$ 
10    if  $CrNBL - CrBL \leq 0 == \text{False}$  then return False

```

**return** True

---

## 2.2. Metaheuristic Solver

As discussed in the introduction, two types of solvers are used for Simulation Based Optimization, namely Mixed Integer Linear Programming (MILP) and metaheuristics. MILP cannot handle large amounts of variables, commonly associated with mine planning solver. Therefore, MILP solvers require aggregation algorithms to reduce the variables. Aggregation is the technique of combining several variables to create one variable. This can be done by creating mining-cuts based on rock-types and grade ranges (Askari-Nasab et al., 2010). Aggregating blocks for both the grade and the GCC's in such a way that the aggregates do not severely limit the ability of the solver to optimize for the production target and the GCC target blend is expected to be complicated. Therefore, it is chosen in this research to develop a metaheuristic solver instead of a MILP solver.

The principles of metaheuristic algorithms are discussed first, followed by a discussion about multi-objective optimization and hybridization of metaheuristic algorithms. Based on this work and the mine planning framework, the requirements for the solver framework are discussed and the general structure of this framework is presented and justified. The principles of the metaheuristic algorithms chosen for this framework are elaborated in more detail before the the individual phases of the solver are developed.

### 2.2.1. Principles of Metaheuristics

The tasks of a solver for global optimization are(Weise, 2009):

1. To find solutions that are as good as possible
2. To find solutions that are widely different from each other

Metaheuristics are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space (Gendreau and Potvin, 2010). Metaheuristic problems are commonly inverse problems, for which a fitness function is available, but not the inverse of the function, meaning that it is impossible to analytically derive a solution given a certain required outcome (Luke, 2015). Metaheuristics are designed to overcome inverse problems without making strong or even to no assumptions about the nature of the fitness function. For this, metaheuristic algorithms rely on statistics derived from samples from the search space or a model of a natural phenomenon or physical process (Weise, 2009).

The fitness functions describes the optimality of a candidate solution, given a set of objective functions and boundary conditions. The convention in optimizations is that the optimization problem is defined as a minimization problem (Weise, 2009). Maximizations are then simply minimizations in the negative solution space. The solutions space describes the function outcome against a set of variables. Figure 2.4 shows a two-dimensional solution space, with local and global optima. A local optimum is an optimum of a subset of variables  $\mathbb{X}$ , while the global optimum the optimum for the entire domain is (Weise, 2009). Methods to evaluate the fitness for multi-objective function are discussed in the section about multi-objective optimization.

Two characteristics of the solutions space are important to be considered. The first characteristic is the ruggedness, which defines the fluctuations and steepness of the solutions space. The more rugged the space, the more difficult it is for the solver to find optima. The second characteristic is the epistatis effect, which is defined as the dependency of the contribution of one variable to the objective functions on the state of other variables. Large epistatis effects have a negative effect to the performance of the solver because of the complex contribution of one variable to the fitness. A possible countermeasure for these two characteristics is the use of population based metaheuristic algorithms (Luke, 2015).

Metaheuristic solvers typically consist of two distinct phases (Weise, 2009):

**Phase 1** Exploration of the solution space

**Phase 2** Exploitation of the solution space

In the exploration phase, points in the undiscovered areas of the solutions space are investigated, finding novel solutions. On the other hand, in the exploitation phase, the known promising areas of the solution space are explored more thoroughly to find more optimal solutions (Weise, 2009). All optimization methods

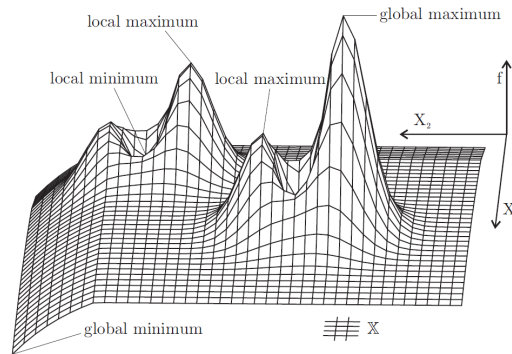


Figure 2.4: The two dimensional solution space of function  $f$  with local and global optima Weise (2009)

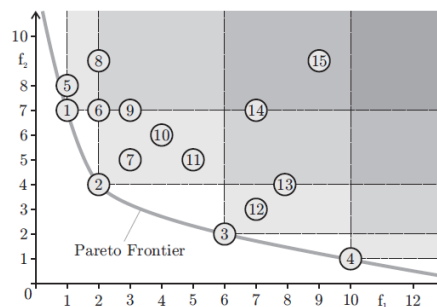


Figure 2.5: The Pareto Front of two functions for a minimization problem (Weise, 2009)

can be tuned for one of the two phases (Luke, 2015). Different ways to combine metaheuristics for this purpose are discussed in the section about hybridization.

### 2.2.2. Principles of Multi-Objective Optimization

The fitness represents in many cases a combination of several objective functions, which commonly conflict with each other, therefore, require a sound method to evaluate their optimality.

The simplest way to evaluate multi-objective optimizations is to bundle all the objective functions together in weighted form in the fitness function, as shown in equation 2.15, where  $F$  is the fitness function,  $f_{nf}$  the objective function and  $w$  the weights for the objective function. The signs of the weights make it possible to combine minimization and maximization objective functions. The drawback of this method is the selection of weights, which is especially difficult if the objectives are non-linear and rise and fall with different speeds, causing differences in order of magnitudes (Luke, 2015). Therefore, this method is only suitable if all objective functions share the same order of magnitude (Weise, 2009).

$$F(\mathbb{X}) = \sum_{nf=1}^{NF} w_{nf} \cdot f_{nf}(\mathbb{X}) \tag{2.15}$$

Another approach is the use of the Pareto optimality, which trades off conflicting objectives in an optimal manner. This method relies on the dominance principle, which states that a solution dominates another solution if it performs equally well or better for all objective functions, and outperforms the other solutions for at least one objective function (Luke, 2015). The set of Pareto optimal solutions is called the Pareto front. A solution is in the Pareto front if it is not dominated by any other method. Figure 2.5 shows such a Pareto front. The task is to create a Pareto-Efficient rank list which orders the solutions based on the dominance and diversity (Ribeiro, 2013).

One method is to create the Pareto Front Rank, which is the front number a solution is in. Different fronts

are created by iteratively removing all dominant solutions. The lower the rank, the better the solution. This type of sorting is called **Non-Dominated Sorting** (Luke, 2015). The drawback of this method is that it does not take into account the diversity in terms of the achieved objective functions. For this, the notion of wimpiness can be used, which is defined as the number of solutions which dominate a particular solution (Luke, 2015). A densely sampled part of the solution space receives higher wimpiness values, because solutions are dominated by many others. On the other hand, lesser sampled parts receive lower wimpiness values. Therefore, this method is able to capture both the diversity in objective functions and the dominance. The algorithm is called the **Pareto Prevalence Algorithm** and is shown in algorithm 3.

The lowest Pareto Prevalence solutions can be stored after each run, comparing them against the currently stored optimal solutions, removing the ones that are not considered optimal anymore. When this set becomes too large, pruning algorithms are available to filter a number of optimal solutions to maintain a maximum number of optimal solutions. These approaches are not discussed here. The interested reader is referred to Knowles and Corne (2003). Besides the Pareto dominant solutions, the optimal solution for each of the objective function can also be stored, as shown in algorithm 4.

### 2.2.3. Principles of Hybridization

Combining metaheuristic algorithms is the domain of hybrid metaheuristics. The idea is to exploit the advantages of individual metaheuristic, heuristic and deterministic techniques to obtain a better performance. Four criteria are used to distinguish hybridization methods (Gendreau and Potvin, 2010):

**Hybridized algorithms** The core of hybridization is the combination of different algorithms. The first class combines metaheuristics with other metaheuristics. The second class combines a metaheuristic algorithm with a problem specific algorithm. The third class combines metaheuristics with deterministic (or exact) methods or heuristic and soft-computing methods, such as fuzzy logic and statistical techniques. The fourth class involves a human interaction stage.

**Level of hybridization** The level of hybridization describes the strength of coupling. High-level combination retain the individual identity of the original algorithm, while low-level combinations exchange components of the algorithms.

**Order of execution** The combination of algorithms can be executed in a sequential way or in a parallel way. Parallel hybrid algorithms can be categorized and described in a more detailed manner. The interested reader is referred to Gendreau and Potvin (2010)

**Control strategy** The control strategy described the entanglement of the combination of algorithm. Integrative hybrid algorithms are combinations in which one algorithm is embedded in another algorithm. In collaborative hybrid metaheuristics, the algorithms are not embedded but only share information. Collaborative metaheuristics can be homogeneous, in which several instances of one algorithm are run or heterogeneous, in which different metaheuristics algorithms are used.

### 2.2.4. Overview Available Metaheuristic Algorithms

The metaheuristic algorithms are divided differently by many authors, two of which are discussed here. Luke (2015) divides the algorithms in single state algorithms, Population based algorithms, Co-Evolution algorithms and combinatorial optimization algorithm. On the other hand, Weise (2009) divides the algorithms as shown in figure 2.6. The single state methods, as well as the Evolutionary Computation algorithms are briefly discussed here. The interested reader is referred to Weise (2009) and Luke (2015) for a detailed discussion of all the available algorithms.

The general outline of **Single State** algorithms is as follows (Luke, 2015):

1. Provide one or more initial candidate solutions
2. Assess the quality of a candidate solution

**Algorithm 3** Pareto Prevalence Algorithm - adapted from (Weise, 2009)

---

```

1  procedure PPA(OB)
2    ParFront  $\leftarrow$  []
3    PPR  $\leftarrow$  zeros[P]
4    DeleteList  $\leftarrow$  []
5    while OB! = [] do
6      for all  $p \in P$  do
7        CurObj  $\leftarrow$  OB[p,:]
8        ParFeas  $\leftarrow$  True
9        for all  $q \in P$  do
10         if  $p \neq q$  then
11           if DETDOM(CurObj, OB[q,:]) == False then
12             ParFeas = False
13             break
14         if ParFeas == True then
15           PPR[p]  $\leftarrow$  DETPREVALANCE(CurObj, ParFront)
16           APPEND(DeleteList, p)
17           APPEND(ParFront, CurObj)
18       DELETE(OB, DeleteList)
19       DeleteList  $\leftarrow$  []
20       return PPR

20 procedure DETDOM(CurObj, NObj)
21   if ALL(CurObj  $\geq$  NObj) & ANY(CurObj > NObj) == True then
22     return True
23   else
24     return False

23 procedure DETPREVALANCE(CurObj, ParFront)
24   if ParFront == [] then
25     return 0
26   CountPrev  $\leftarrow$  0
27   for all  $l \in$  ParFront do
28     if DETDOM(ParFront[l,:], CurObj) then
29       CountPrev  $\leftarrow$  CountPrev + 1
30   return CountPrev

```

---

**Algorithm 4** Save the optimal solutions per objective function

---

```

procedure UPDATAAGA(OB, Chrome)
  AGAv  $\leftarrow$  Current optimal values per objective function
  AGAs  $\leftarrow$  Current optimal schedules per objective function
  for all  $o \in O$  do
    if OB[o] > AGA[o] then
      AGAv  $\leftarrow$  OB[o]
      AGAs  $\leftarrow$  Chrome

```

---

3. Copy the candidate solution
4. Modify a candidate solution to produce a randomly slightly different candidate
5. Select some candidates to repeat the procedure

The solutions provided as initial candidates can be completely random or heuristically designed to ensure their feasibility. The quality assessment can be done by the methods as described in the section about multi-objective optimization. The modification step can be done by adding noise to the solution if the solution consists of real values. For binary valued solutions, modification can be achieved by bit-flipping one or more binary values of the solution. Exploration is achieved by large modifications, while exploitation is achieved by making small modifications. This can be done by drawing realizations from a Gaussian distribution to decide the spread of the noise (Luke, 2015). **Hill Climbing With Restart** uses exactly the procedure given for the single state methods, modifying for exploitative purposes with the addition of a large modification every once in a while, restarting the procedure.

**Simulated Annealing** differs from the Hill Climbing Algorithm in the fact that it implements an acceptance criteria for a modification. If the new solution has a higher quality, the solution is always accepted. However, if the new solution is worse than the old solutions, the probability the modification is accepted depends on formula 2.16. The probability of acceptance depends on how much worse the new solution is and  $t$ , describing the phase of the optimization. When starting the optimization, the factor can be set to a high number, accepting many worse solutions. Gradually reducing the factor increases the degree of exploitation of the optimization(Luke, 2015).

**Iterated Local Search** expands the ideas of Hill Climbing With Restart, by keeping a local optimum as a home base, from which a restart is performed in the vicinity of this homebase to find a new local optimum. After the restart, the local optimum of the neighbourhood is found by small modifications. This new local optimum is selected as the new home-base with a certain probability (Luke, 2015).

**Tabu Search** is different from the previous methods by preventing previously created solutions to be selected again. Two main implementations of this tabu list exist, saving either entire solutions or only the modification moves made. If the solution consists of real values, the modification moves are usually kept, because producing the exact same solution as historical solutions is rare. The length of the tabu list is restricted, forcing the oldest entries out at some point, making them available again for future solutions (Luke, 2015).

$$P(t, New, Old) = e^{\frac{Quality(New) - Quality(S)}{t}} \quad (2.16)$$

The methods of **Evolutionary Computation** are based on the ideas of population biology, genetics and evolution and can be subdivided in Evolutionary Algorithms and Swarm Intelligence algorithms. Some of these methods are briefly discussed below.

**Evolutionary Algorithms** are considered as population based metaheuristic optimization methods that use mutation, crossover, natural selection and survival of the fittest in order to refine a set of solutions iteratively Weise (2009). The general outline of evolutionary algorithms is as follows Weise (2009):

1. Create an initial population to seed the optimization task
2. Evaluate the fitness of the population
3. Select candidates from the population where fitter candidates have a higher chance of being selected
4. Create new candidates using the selected individuals by performing crossover and mutation operators
5. Select available candidates as feed for a new run of the algorithm

The crossover operator combines two candidates by swapping some of the characteristics of both candidates. The mutation operator is similar to the modification operator for the single state methods and randomly changes the characteristics of the solution.



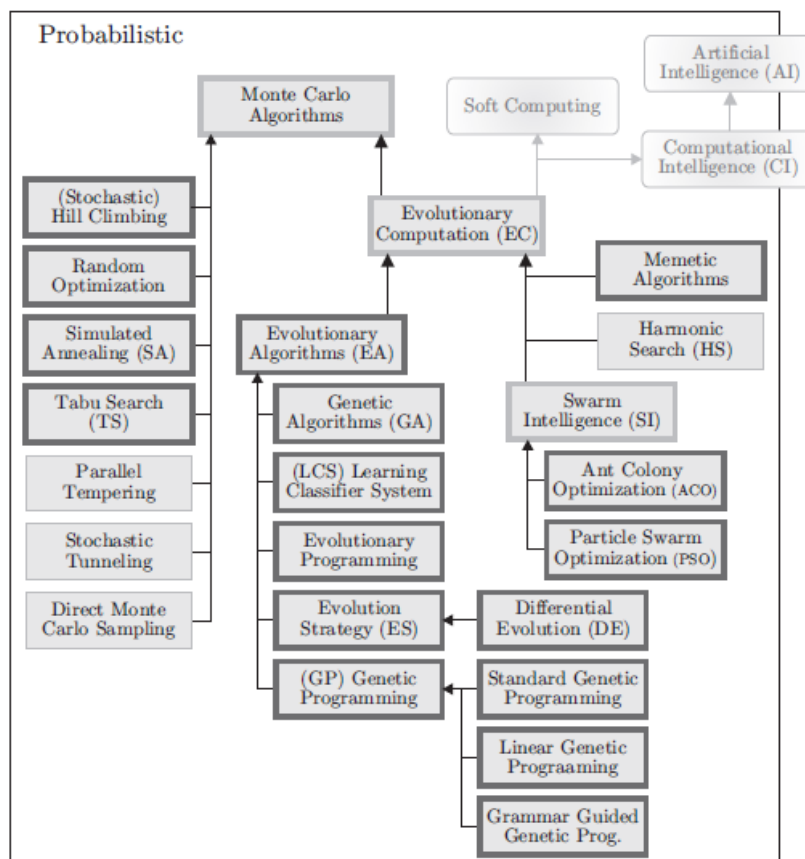


Figure 2.6: Overview of the metaheuristic algorithms (Weise, 2009)

Different Evolutionary Algorithms exist for different problem instances. **Genetic Algorithms** are developed for which solutions are represented by bit strings, while **Genetic Programming** algorithms can solve for optimizing algorithms and programs. **Evolution Strategies** explore and optimize real vectors and **Learning Classifier Systems** optimize machine learning tasks using the same principles (Weise, 2009). The Evolutionary Algorithms can be distinguished by the number of offspring created and the number of parent individuals used to create offspring. Furthermore, it can be decided to keep the parents for a new run, creating a preservative scheme, or only consider the offspring for the new run, creating an extinctive selection scheme.

**Swarm Intelligence** use the properties of animal swarm behaviour to guide the search for good solutions. Two popular methods exist, namely **Ant Colony Optimization** and **Particle Swarm Optimization**. Both methods rely on sharing knowledge of the solution space to the swarm to allow members of the swarm to create new solutions. For this, the components of a solution are valued based on historical performance and its normal value and chosen by a member of the swarm based on this value. The Ant Colony Optimization uses ants to form paths that form the solutions. The fitness of the path is shared with the colony. To prevent premature convergence, the stored fitness value is reduced in time or averaged. Particle Swarm Optimization uses particles with a certain direction and speed to create new solutions. The particles share their knowledge of the solution space with neighbouring particles, which use this information to update their speed and direction. The best position ever visited by any particle in the algorithm is also stored and can also be accessed by the particles.

### 2.2.5. Establishing the Structure of the New Solver

Section 2.1 established a multi-objective optimization task, for which the capacity, blending penalty and smoothing penalty are of different order and are expected to change at different speeds for some modifications. Therefore, it is chosen to use the Pareto Prevalence Algorithm to properly assess the fitness of the entire set of objective functions while enforcing diversity in the ranking. The variable to optimize is  $\mathbb{X}$ , representing a  $N \times T$  matrix of  $N$  blocks to be mined in  $T$  periods. The blocks are indexed according to an inverse gslib format, as described by formula 2.17. At every index of the matrix, a binary value describes whether block  $n$  is mined in period  $t$ , with the value 1, or not mined in this period, with the value 0.

$$\begin{aligned}
 I_{gslib,n} = & \frac{BX_n - O_x}{S_x} \\
 & + N_x \cdot \text{MATH.FLOOR}\left(\frac{BY_n - O_y}{S_y}\right) \\
 & + N_y \cdot N_x \cdot \text{MATH.FLOOR}\left(\frac{E_z - BZ_n}{S_z}\right)
 \end{aligned} \tag{2.17}$$

It is expected that the solution space of the mine planning framework is rugged and will show signs of epistasis, because of the variables that depend on each other by the slope angle constraint and the small domain of feasible solutions. Therefore, it is chosen to only use population based methods. Lamghari and Dimitrakopoulos (2012) and Godoy (2003) do differently and use Tabu Search and Simulated Annealing algorithms, respectively. For the exploration phase, the Genetic Algorithm is chosen from the evolutionary algorithms due to its ability to solve for binary represented solutions and the explorative features of the mutation operator. For the exploitation phase, the Ant Colony Optimization algorithm is chosen because of its ability to preserve and exploit historical knowledge of the solution space. Furthermore, the implementation is expected to be easier than for the Particle Swarm Optimization, given the bit representation of the solution used here. The Ant Colony Optimization is further improved by storing the knowledge of the solution space gained during the exploration stage. Given this combination of algorithms, the solver can be named a sequential, low-level collaborative linked GA/ACO. Besides the exploration and exploitation stage, a seed phase is implemented to create feasible, diverse solutions for the exploration stage. The different stages are listed below:

**Phase 1 - Seed**                      The seed stage produces a diversified population of feasible schedules for the exploration stage.

**Phase 2 - Exploration**            The exploration stage uses a Genetic Algorithm to explore the solution space.

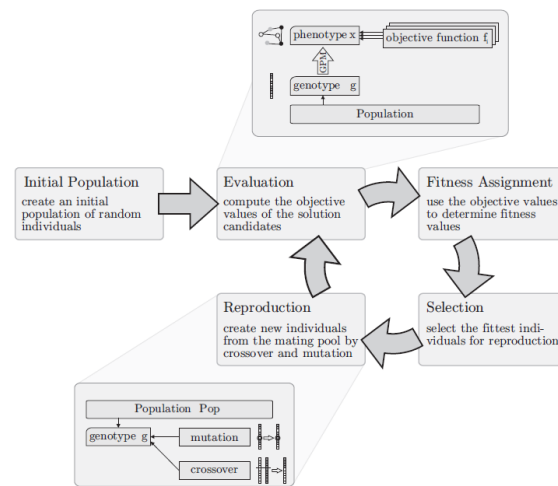


Figure 2.7: General overview of the Genetic Algorithm (Weise, 2009)

**Phase 3 - Exploitation** The exploitation stage uses Ant Colony Optimization to exploit the solutions space, using the information about the solution space as gathered by the exploration stage.

The key element in all phases will be the implementation of heuristic components to the standard algorithms to enforce feasible solutions given the small feasible domain. Heuristics are strategies derived from experience with similar problems, using readily accessible information to control the problem solving (Pearl, 1983). These heuristics are discussed in the development section of the solver. First, a more detailed discussion on the principles of the Genetic Algorithm and the Ant Colony Optimization is given based on a literature review. This provides a platform to integrate the heuristic components in the Genetic Algorithm and Ant Colony Optimization.

### 2.2.6. Principles of the Genetic Algorithm

Genetic algorithms are based on the evolutionary process, in which offspring is created from parents. The offspring will retain several characteristic properties of the parents, but can also mutate, randomly changing some of the DNA of the offspring. The offspring is then evaluated by nature for its ability to survive. Based on this survival ability, some of the offspring is able to mate and the process is repeated. The population is the group of parents available for selection. In genetic algorithms, DNA of the offspring or parents are called genomes or chromosomes. The chromosome maps the properties of the solution, which are in the phenotype space. The different elements of the chromosome are called genes, which can represent many structures, such as bits and real numbers. An allele is a value of a gene and the locus the position of the gene (Weise, 2009). A general overview of genetic algorithms is shown in figure 2.7 and algorithm 5.

The crossover operation blends two parent chromosomes to form a new chromosome. The crossover operation should ideally spread valuable building blocks of the phenotype by swapping alleles in two selected chromosomes. Three classical types of crossover operations exist for this purpose (Luke, 2015):

**One-Point Crossover** One random locus is chosen, which is used to swap a part of one parent to the other parent

**Two-Point Crossover** Two random loci are chosen, which selects a subset of the chromosome to be swapped.

**Uniform Crossover** All genes have a certain probability to be swapped.

By only performing crossovers, the global solution space cannot be considered. The mutation operation diversifies the population by randomly modifying one or more alleles at random loci, hereby being able to explore the entire solution space. If the allele is a binary value, a bit-flip operation can be performed. For a real number allele, a random realization from a specific distribution can be used to change the allele.

---

**Algorithm 5** Genetic Algorithm - adapted from (Gendreau and Potvin, 2010)

---

```

1 procedure GA
2   ChromePop ← SEEDINITIALPOP
3   while True do
4     for all  $p \in \textit{ChromePop}$  do
5       if CROSSOVERCONDITION == True then
6          $Pa_1, Pa_2 \leftarrow \text{SELECTTWO PARENTS}(P)$ 
7          $Child \leftarrow \text{CROSSOVER}(Pa_1, Pa_2)$ 
8       if MUTATIONCONDITION == True then
9          $Child \leftarrow \text{MUTATE}(Child)$ 
10       $F_p \leftarrow \text{EVALUATECHROMOSOME}(Child)$ 
11    if STOPCRITERIA == True then
12      Break

```

---

The selection procedure to select parent chromosomes can select one chromosome more than once. Commonly used selection algorithms are the Fitness Proportionate Selection (FPS) algorithm and the Tournament Selection (TS) algorithm. The FPS randomly selects a chromosome in proportion to their fitness, meaning that a fitter parent has a higher chance of being selected than a less fit parent (Luke, 2015). The fitness proportionate selection algorithm is shown in algorithm 6. The TS algorithm randomly selects a chromosome one at a time, for a  $TO_{size}$  number of times. A new chromosome is accepted if its fitness is higher than the currently stored chromosome. For very long tour lengths, this algorithm will select the best performing chromosome, while at small tour lengths, the selection is almost random. This feature can be used to move from exploration to exploitation (Weise (2009)). The TS algorithm is shown in algorithm 7.

---

**Algorithm 6** Fitness Proportionate Selection - adapted from Luke (2015)

---

```

procedure FITPROPSELECT(Values)
   $n \leftarrow \text{SIZE}(\textit{Values})$ 
   $maxV \leftarrow \text{MAX}(\textit{Values})$ 
  while True do
     $i \leftarrow \text{RANDOMINT}(1, n)$ 
    if  $N(0, 1) < \textit{Values}[i]$  then return  $i$ 

```

---



---

**Algorithm 7** Tournament Selection - adapted from Weise (2009)

---

```

1 procedure TOURSELECT( $PPR, TO_{size}$ )
2    $Cand1 \leftarrow \text{RANDOM}(PPR)$ 
3   for all  $t \in TO_{size}$  do
4      $Cand2 \leftarrow \text{RANDOM}(PPR)$ 
5     if  $Cand2 > Cand1$  then
6        $Cand1 \leftarrow Cand2$ 

```

**return**  $Cand1$

---

### 2.2.7. Principles of the Ant Colony Optimization

In nature, ants spread pheromones to mark routes to food sources. The amount of pheromones spread by an ant is constant, meaning that shorter paths will receive a higher amount of pheromones. The ants create the paths based on the pheromones, where a path component with a higher pheromone value has a higher

chance of being selected. Shorter routes will therefore attract a larger amount of ants, which increase the pheromones even more. This positive feedback strategy is combined with a negative feedback strategy, by evaporating the pheromones over time, to remove the incentive to follow a route if the food source is emptied. This way, the ant colony can find their shortest path to a food source and share the knowledge to the entire colony (Kalinli and Sarikoc, 2007).

The Ant Colony Optimization (ACO) algorithm runs in similar fashion, keeping a population of ants forming the colony that construct ant trails (or solutions) by selecting components one by one, based on the value of the component and the assigned pheromone. The trail can be in the genotype or directly in the phenotype space. When all trails are developed, the trails are evaluated and its components will be assigned a new value based on the fitness of the trail. Optionally, an additional exploitation step can be added after the construction of the ant trail. However, also without this extra exploitation step, the ACO algorithm shows very good performance (Gendreau and Potvin, 2010). An overview of the general layout of the algorithm without an additional exploitation step is shown in algorithm 8.

---

**Algorithm 8** Ant Colony Optimization - adapted from (Luke, 2015)

---

```

1  procedure ACO
2     $C \leftarrow C_1, \dots, C_n$ 
3     $A \leftarrow$  Colony Size
4     $p \leftarrow \langle p_1, \dots, p_n \rangle = 0$ 
5     $Best, BestFitness \leftarrow []$ 
6    while True do
7      for all  $a \in A$  do
8         $P_a \leftarrow$  SELECTCOMPONENTS()
9         $F_a \leftarrow$  EVALFITNESS( $P_a$ )
10       if  $Best == []$  OR  $F_a > BestFitness$  then
11          $Best \leftarrow P_a$ 
12          $BestFitness \leftarrow F_a$ 
13       UPDATEPHEROMONES( $F$ )
return  $Best$ 

```

---

The values of the components are called desirability values and are the product of the pheromone and the regular value of the component (Luke, 2015). The values of the components can also be updated according to an elitist strategy, in which better solutions found during the search contribute stronger to the pheromones than weaker solutions (Gendreau and Potvin, 2010).

The selection of the components can be done in a greedy heuristic way by using the FPS or TS algorithm as shown in algorithm 6 and 7. The heuristics ensure that only feasible components can be selected. It is also possible to allow the creation of infeasible solutions and penalize these according to the degree of infeasibility. The probability of a component to be selected can be calculated using equation 2.18 and is dependent of the previously selected components  $s_p$ .  $\tau$  is the pheromone value of variable  $i$  for a certain follow-up moves  $j$ . The function  $\eta$  assigns a heuristic value to the solution component  $c_i^j$ , given the previously created solution space  $N(s_p)$ . The power factor  $\alpha$  controls the weight of the pheromones, while  $\beta$  controls the weight of the heuristic information (Gendreau and Potvin, 2010).

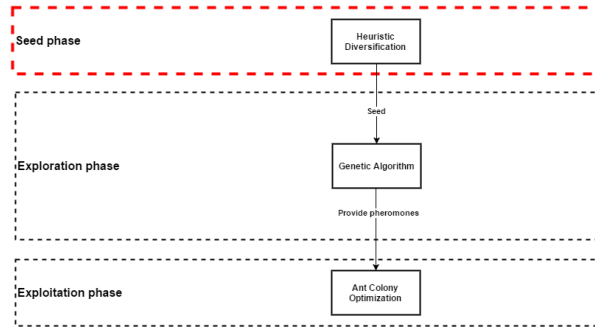
$$p(c_i^j | s_p) = \frac{\tau_{ij}^\alpha \cdot [\eta(c_i^j)]^\beta}{\sum_{c_i^l \in N(s_p)} \tau_{i,l}^\alpha \cdot [\eta(c_i^l)]^\beta} \quad (2.18)$$

The evaporation can be done by removing a fixed portion of the pheromones of the components after every run. Or one could take the average of the pheromones assigned so far to a component and add a portion

or the full fitness of the current ant trail (Luke, 2015).

Different methods exist to adapt the ACO algorithm for multi-objective optimization, differing in the number of ants and the pheromone structure. The best performing variant uses a single colony, in which at every run, a random objective function is chosen, which are used by the ants to create a path. Every objective function has its own pheromone structure and are updated separately. This method is called m-ACO(1,m) (Alaya, 2007).

### 2.2.8. Development of the Phases for the New Solver



The goal of the **seed phase** is to produce diverse, feasible solutions to the Exploration phase. The idea behind the algorithm developed is the use of cones at different locations of different depths and slope angles to include blocks to a certain period. Five schedules are created using a random slope angle, which is constrained to the maximum slope angle and two schedules are created with a minimum and maximum slope angle. The diversity is then determined based on the compaction, centre of mass and number of blocks per quadrant of the deposit. The production of a new schedule is repeated until a diverse population is created. The seed phase consists of four steps, described below and shown in algorithm 9:

1. The blockmodel is divided into quintants as shown in figure 2.8. From each quintant, a starting point is selected for the cone algorithm. The starting points are forced to obey the mining boundary, which is based on the slope angle.
2. The cone algorithm, discussed later in this section, is used to create feasible schedules with random cone slopes from the five starting points. Furthermore, a schedule is made from the middle of the block model with a small slope angle and the maximum slope angle.
3. The diversity of the created population of schedules is calculated using a compaction, centre of mass and blocks per quadrant measurement.
4. The diversity of the population is increased till the improvement is smaller than 1% for 3 sequential runs by changing the angle of the cone and rerunning the cone algorithm.

The cone algorithm consists of two phases, described here and shown in algorithm 11. The first phase creates a cone and adds all blocks within the cone to the first period. The cone depth is calculated using the selected cone slope angle and the smallest available capacity (i.e. mining or processing) in the period considered, using formula 2.19. The cone slope is not allowed to exceed the maximum slope angle to enforce the slope feasibility. The minimum capacity is scaled with  $F_{reduc}$  to enforce a small enough cone that is not exceeding the capacity for that year. From the starting point, the cone depth is subtracted and from that point, a closest available base block within the mining boundary is selected for the base of the cone template. To determine whether a block is part of the cone, the radius of the cone is determined at all the block levels and the euclidean distance is calculated from all blocks to the base block. An example of such a cone is shown in 2.9. The radius is calculated using formula 2.21.

The second phase consists of adding blocks individually within the cone and within the mining boundary to a certain period until the period has reached its capacity. If the capacity is not reached with the current cone set-up, the cone depth is lowered by one block and the procedure is repeated until the capacity is

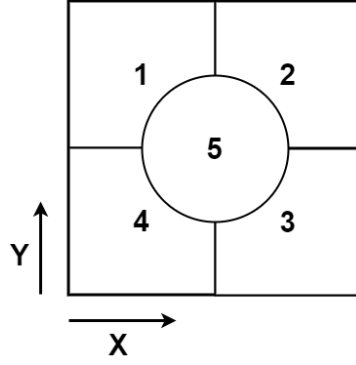


Figure 2.8: Overview of the quintants created for the starting points.

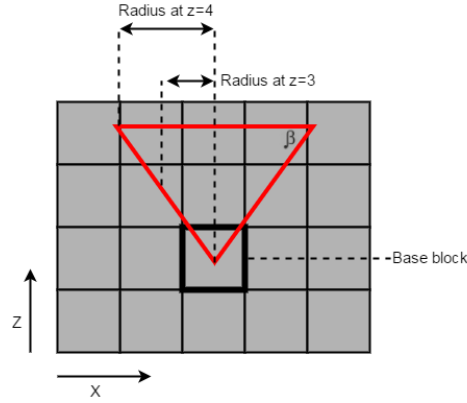


Figure 2.9: The cone radius is determined per z-level to determine which blocks are captured by the cone.

reached or until all the blocks within the boundary are mined. Furthermore, if a block is selected, which has overlying blocks which are not mined yet, these are forced to be mined in the same period if there is enough capacity in the current period. If this is not the case, the blocks with its overlying blocks are skipped for the current period.

$$D_{cone}^t = [V_t \cdot \frac{TAN(\alpha)^2 \cdot 3}{\pi}]^{\frac{1}{3}} \quad (2.19)$$

$$V_t = \sum_{r=1}^t \min[C_{mine,r}, C_{proc,r}] \cdot F_{reduc} \quad (2.20)$$

$$R_{cone} = \frac{D_{cone}}{TAN(\beta)} \quad (2.21)$$

The diversity of a population is measured using three characteristics, namely the compaction, the centre of mass and the number of blocks per quadrant. The characteristics are inversely scaled with the period to produce one value per schedule. The combination of these characteristics give a unique profile of a schedule. The formulations of the three characteristics are given in formula 2.22, 2.23, 2.24, 2.25, 2.26, 2.27, 2.28 and 2.29. The diversity of the population is then determined by normalizing the characteristics and summing the closest differences of the characteristics, creating the diversity matrix. The procedure for calculating the diversity is shown in algorithm 10. The least diverse schedule is chosen to be produced again, using a new randomly chosen slope angle and starting point.

$$CMX_t = \frac{\sum_{n=1}^N BX_n \cdot X_{n,t}}{\sum_{n=1}^N X_{n,t}} \quad (2.22)$$

$$CMY_t = \frac{\sum_{n=1}^N BY_n \cdot X_{n,t}}{\sum_{n=1}^N X_{n,t}} \quad (2.23)$$

**Algorithm 9** Seed a diverse population of solutions for the exploration phase

---

```

1 procedure SEEDPOPULATION
2   ChromePop  $\leftarrow$  zeros(N, T, P)
3   S  $\leftarrow$  RETURNQUINTANTSTARTINGPOINTS
4   for s  $\in$  S do
5      $\beta$   $\leftarrow$  RANDOMSLOPEANGLE
6     ChromePop[:, :, s]  $\leftarrow$  CREATESCHEDULE( $\beta$ , S[s])
7      $\beta$   $\leftarrow$  WORSTSLOPE
8     ChromePop[:, :, 6]  $\leftarrow$  CREATESCHEDULE( $\beta$ , S[5])
9      $\beta$   $\leftarrow$  BESTSLOPE
10    ChromePop[:, :, 7]  $\leftarrow$  CREATESCHEDULE( $\beta$ , S[5])
11    for p  $\in$  P do
12      OB[p, :]  $\leftarrow$  CALCULATEFITNESS(ChromePop[:, :, p])
13      PPA(OB[p])
14      UPDATEAGA(ChromePop[:, :, p], OB[p, :])

15 procedure DIVERSIFYPOPULATION
16   DIV  $\leftarrow$  zeros(P)
17   while True do
18     if VERIFYDIVERSEPOPULATION(DIV)==True then
19       Break
20     DIV  $\leftarrow$  CALCULATEDIVERSITY(ChromePop)
21     di  $\leftarrow$  MAX(DIV)
22      $\beta$   $\leftarrow$  RETURNRANDOMSLOPEANGLE
23     ChromePop[:, :, di]  $\leftarrow$  CREATESCHEDULE( $\beta$ , S[di])
24     OB[di, :]  $\leftarrow$  CALCULATEFITNESS(ChromePop[:, :, di])
25     PPA(OB[di])
26     UPDATEAGA(ChromePop[:, :, p], OB[p, :])

```

---



**Algorithm 10** Calculate the diversity of the population

---

```

1 procedure CALCULATEDIVERSITY(ChromosomePop)
2    $C_o, C_m, C_q \leftarrow \text{ZEROS}(P)$ 
3    $Div \leftarrow \text{ZEROS}(P,3)$ 
4   for all  $p \in P$  do
5      $C_m[p] \leftarrow \text{CALCULATECENTREMASS}(ChromosomePop)$ 
6      $C_o[p] \leftarrow \text{CALCULATECOMPACTION}(ChromosomePop, C_m)$ 
7      $C_q[p] \leftarrow \text{CALCULATEQUADRANTBLOCKS}(ChromosomePop)$ 
8    $Di \leftarrow [C_m, C_o, C_q]$ 
9    $Di \leftarrow \text{NORMALIZE}(Di)$ 
10  for all  $d \in Di[0, :]$  do
11     $tmp \leftarrow \text{SORT}(Di[:, d])$ 
12    for all  $p \in P$  do
13       $Div[p, :] \leftarrow tmp[p, :] - tmp[p + 1, :]$ 
14  for all  $p \in P$  do
15     $Div_m[p] \leftarrow \text{MEAN}(Div[p, :])$ 
return  $1 - Div_m$ 

```

---

$$CMZ_t = \frac{\sum_{n=1}^N BZ_n \cdot X_{n,t}}{\sum_{n=1}^N X_{n,t}} \quad (2.24)$$

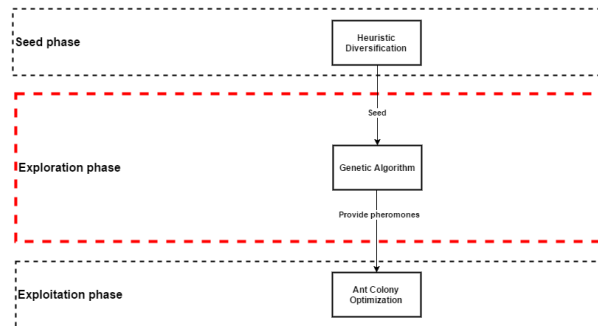
$$CP = \sum_{t=1}^T \frac{1}{t} \cdot \sum_{n=1}^N [(BX_n \cdot X_{n,t} - CMX_t)^2 + (BY_n \cdot X_{n,t} - CMY_t)^2 + (BZ_n \cdot X_{n,t} - CMZ_t)^2] \quad (2.25)$$

$$NQ_1 = \sum_{t=1}^T \frac{1}{t} \cdot [N_t \in Q1] \quad (2.26)$$

$$NQ_2 = \sum_{t=1}^T \frac{1}{t} \cdot [N_t \in Q2] \quad (2.27)$$

$$NQ_3 = \sum_{t=1}^T \frac{1}{t} \cdot [N_t \in Q3] \quad (2.28)$$

$$NQ_4 = \sum_{t=1}^T \frac{1}{t} \cdot [N_t \in Q4] \quad (2.29)$$



---

**Algorithm 11** Create an initial schedule from scratch

---

```

1 procedure CREATESCHEDULE( $\beta, s, t$ )
2   for all  $t \in T$  do
3      $D \leftarrow$  CALCULATECONEDEPTH( $\beta$ )
4      $b \leftarrow$  DETERMINEBASEBLOCK( $D, s$ )
5      $Chrome[:, t] \leftarrow$  INCLUDEOVERLYINGBLOCKS( $b, t$ )
6     while VERIFYCAPACITY( $Chrome[:, t]$ ) == True do
7        $Chrome[:, t] \leftarrow$  INCLUDECONEBLOCKWISE( $Chrome, \beta, D, s, t$ )
8        $D \leftarrow D + S_z$ 
9        $b \leftarrow$  DETERMINEBASEBLOCK( $D, s$ )
   return  $Chrome$ 

10 procedure INCLUDECONEBLOCKWISE( $Chrome, \beta, D, s, t$ )
11    $Radii \leftarrow$  CONERADIUSPERLEVEL( $\beta, D, s$ )
12   for all  $n \in N$  do
13     if VERIFYBLOCKFEAS( $n, Radii, Chrome, s$ ) then
14       if VERIFYOVERLYINGBLOCKCAP( $Chrome, n, t$ ) then
15          $Chrome[n, t] \leftarrow 1$ 
16          $Chrome \leftarrow$  INCLUDEOVERLYINGBLOCKS( $n, t$ )
17     if VERIFYWITHINCAPACITY( $Chrome$ ) then
18       Break
   return  $Chrome$ 

19 procedure INCLUDEOVERLYINGBLOCKS( $b, t$ )
20   for all  $b_o \in LB[b, :]$  do
21     if  $\sum_{r=1}^t Chrome[b_o, r] == 0$  then
22        $Chrome[b_o, t] \leftarrow 1$ 
23     INCLUDEOVERLYINGBLOCKS( $b_o, t$ )
   return  $Chrome$ 

24 procedure VERIFYBLOCKFEAS( $n, Radii, Chrome, s$ )
25   if  $\sum_{r=1}^t Chrome[n, r] = 0$  then return False
26   else if  $BB[n] = 1$  then return False
27   else if VERIFYWITHINRADIUS( $n, Radii, s$ ) == False then return False
28   else return True

```

---

The goal of the **Genetic Algorithm** is to explore the solution space and improve locally where possible. The explorations are carried out by the mutation operator, which applies a common bit-flip operation to *Chrome* within the mining boundary and repairs the schedule afterwards. The local improvements are performed by the crossover operator, which compares two selected parents based on one objective function and selects the worst period to push a random amount of blocks to the next or previous period. The population size is kept the same as the number of solutions provided by the seed phase, which also equals the number of offspring created in the crossover stage.

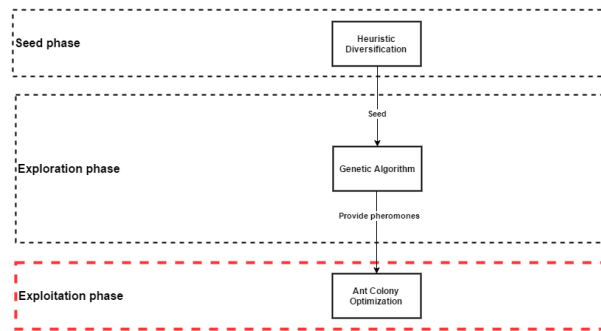
The layout of the exploration phase algorithm is similar to the general GA algorithm (5), with some adaption for the crossover and mutation operator and is shown in algorithm 12. For the crossover operator, two parents are selected based on their Damped Pareto Prevalence Rank (DPPR) (equation 2.31). Damping is performed by a nichefactor, describing the diversity of a solution within a population. For this, the same algorithm is used as described in the seed phase (10). The nichefactor is itself damped in time by equation 2.32, to move from exploration towards exploitation. The selection of the parents is done using Tournament Selection as given in algorithm 7. The  $TO_{size}$  is calculated by formula 2.30, which also moves from exploration towards exploitation. Given two parents, a period is selected using tournament selection for a randomly chosen objective function, for which worse periods have a higher chance of being chosen. Once a period is selected, a random amount of blocks is chosen to be pushed forward or backward. When a block is pushed forward, it is selected from the highest z levels of the current period and when a block is pushed backward, it is selected from the lowest z levels of the current period, as to perform a feasible move. The capacities are then repaired by iteratively moving excess blocks forward per period and then backwards until a feasible schedule is created. The crossover is shown in algorithm 13

The mutation algorithm selects a random block and changes the period it is mined in. The schedule is then repaired in two steps: 1) the slopes are repaired by adding the overlying blocks and infeasible underlying blocks to the same period as the mutated block. 2) The second step repairs the capacities by iteratively moving excess blocks forward per period and then backward until a feasible schedule is created. Due to the destructive power of the mutation and the computational effort, the mutation chance  $M_{chance}$  is set to a low number. The outline of the mutation procedure is shown in algorithm 12. After the mutation, the objective function and DPPR is evaluated. The GA saves the knowledge of the solution space as pheromones in terms of the objective functions for the exploitation phase. The pheromones are in the end of the GA averaged block-wise to prevent large differences in the pheromones if a block is chosen more often than other blocks.

$$TO_{size} = TO_{min} + \frac{(GAL - TO_{start})^2}{(TO_{max} - TO_{min}) \cdot TO_{quit}} \quad (2.30)$$

$$DPPR = PPR \cdot (1 - NF) \quad (2.31)$$

$$DA_{GAL} = \sqrt{[(1 - (GAL - DA_{start})^2) \cdot (DA_{start} + DA_{quit})]} \quad (2.32)$$



The **Ant Colony Optimization** (ACO) performs the exploitation part of the optimization, using the pheromones created in the GA. It uses the (1,m) type of ACO to optimize for the multi-objective problem. The (1,m) type

**Algorithm 12** Heuristic Genetic Algorithm

---

```

1 procedure HGA(ChromPop)
2   GAL  $\leftarrow$  0
3   Children  $\leftarrow$  zeros[N, T, P]
4   while STOPCRITERIA == False do
5     TOsize  $\leftarrow$  UPDATETOURDAMPFACTOR(GAL)
6     OB  $\leftarrow$  EVALUATEFITNESS(ChromPop)
7     PPR  $\leftarrow$  PPA(OB)
8     DPPR  $\leftarrow$  CALCULATEDPPR(PPA)
9     UPDATEPHEROMONES(OB)
10    UPDATEAGA(OB, ChromPop)
11    for all  $p \in P$  do
12      P1, P2  $\leftarrow$  TOURSELECT(DPPR, TOsize)
13      Children[ $[:, :, p]$ ]  $\leftarrow$  HEURCROSSOVER(ChromPop[ $[:, :, P1]$ ], ChromPop[ $[:, :, P2]$ ], TOsize)
14      if  $M_{chance} > N(0, 1)$  then
15        Children[ $[:, :, p]$ ]  $\leftarrow$  HEURMUTATE(Children[ $[:, :, p]$ ])
16    ChromPop  $\leftarrow$  Children
17    Children[ $[:, :, :]$ ]  $\leftarrow$  0
18    GAL  $\leftarrow$  GAL + 1

```

---

**Algorithm 13** Heuristic crossover

---

```

1 procedure HEURCROSSOVER(Chrom1, Chrom2, TOsize)
2   Child  $\leftarrow$  zeros[N, T]
3   frandom  $\leftarrow$  SELECTRANDOMOBJECTIVE
4   Yf,t  $\leftarrow$  CALOBJECTIVEDIFFERENCE(Chrom1, Chrom2)
5   tworst  $\leftarrow$  TOURSELECT(Yf,t, TOsize)
6   Nmove  $\leftarrow$  RANDOMINT(1, CRmaxmove)
7   if  $0.5 > N(0, 1)$  then
8     Child  $\leftarrow$  MOVEBLOCKFORWARD(Nmove, Chrom, tworst)
9   else
10    Child  $\leftarrow$  MOVEBLOCKBACKWARD(Nmove, Chrom, tworst)
11  Child  $\leftarrow$  REPAIRCAP(Child) return Child

```

---

**Algorithm 14** Heuristic Mutation

---

```

1  procedure HEURMUTATION(Child)
2    while True do
3       $i \leftarrow \lfloor N(0, 1) \cdot N \rfloor$ 
4      if  $BB[i] == 1$  then
5        Break
6       $t_{move} \leftarrow \lfloor N(0, 1) \cdot T \rfloor$ 
7       $Child[i, :] \leftarrow 0$ 
8       $Child[i, t_{move}] \leftarrow 1$ 
9       $Child \leftarrow \text{REPAIRPUSH}(Child, i, t_{move})$ 
10     return Child

10 procedure REPAIRPUSH(Child, i,  $t_{move}$ )
11 if  $\text{VERIFYSLOPEFEAS}(Child) == \text{False}$  then
12    $Child \leftarrow \text{REPAIRSLOPE}(Child, i, t_{move})$ 
13 if  $\text{VERIFYCAPFEAS}(Child) == \text{False}$  then
14    $Child \leftarrow \text{REPAIRCAP}(Child)$ 
15   return Child

15 procedure REPAIRSLOPE(Child, i,  $t_{move}$ )
16    $i_{down} \leftarrow i - N_x \cdot N_y$ 
17    $t_{i_{down}} \leftarrow \text{NONZERO}(Child[i_{down}, :])$ 
18   if  $i_{down} > N$  then
19     if  $t_{i_{down}} < t_{move}$  then
20        $Chrome[i_{down}, :] \leftarrow 0$ 
21        $Chrome[i_{down}, t_{move}] \leftarrow 1$ 
22   for all  $b \in BB[i, :]$  do
23     if  $t_b > t_{move}$  then
24        $Chrome[b, :] \leftarrow 0$ 
25        $Chrome[b, t_{move}] \leftarrow 1$ 
26        $Chrome \leftarrow \text{REPAIRSLOPE}(Child, b, t_{move})$ 
27   return Child

27 procedure REPAIRCAP(Child)
28 while  $\text{VERIFYCAPFEAS}(Child) == \text{False}$  do
29   for all  $t \in T$  do
30      $N_{move} \leftarrow \text{DETERMINEOVERCAP}(Child[:, t])$ 
31      $Child \leftarrow \text{MOVEBLOCKBACKWARD}(Child, N_{move}, t)$ 
32   for all  $t \in T$  do
33      $dt \leftarrow T - t$ 
34      $N_{move} \leftarrow \text{DETERMINEOVERCAP}(Child[:, dt])$ 
35      $Child \leftarrow \text{MOVEBLOCKBACKWARD}(Child, N_{move}, dt)$ 
36   return Child

```

---

has  $m$ -ants that optimize for one randomly chosen objective at a time. When all the ants have found a solution, the process is repeated with a new randomly selected objective function.

The ant creates a schedule using the same cone algorithm as used in the seed phase. However, the difference is that per layer, starting at the top layer, a predefined number of cones is created, using randomly selected cone angles. For each of the cones, the summed objective value of the chosen objective function is calculated from the pheromone storage and used to choose a cone with the fitness proportionate selection algorithm. The value of a cone is only determined by the pheromones and not by the direct value of the cone calculated from the objective functions, because the value of the schedule produced in the end depends largely on the sequence in which it is mined, which is captured by the pheromones but not by the direct value of a cone. A higher number of cones results in a higher exploitation. The selected cone is added to the schedule and the depth of the cone is increased by the block size in the  $z$  direction. The process is repeated until the capacities of the periods are met or until all available blocks are mined.

The created ant routes are evaluated for all objectives and saved in a buffer pheromone matrix. After a set Merge Time  $T_{merge}$ , the buffer pheromones are averaged per block for the number of times used during the buffer time and added to the global pheromones. The buffering is done to prevent premature convergence to a local optimum. This way, the ants have enough time to fully exploit the solution space before converging to a smaller subset. The ACO is shown in algorithm 15.

**Algorithm 15** Ant Colony Optimization (1,m)**procedure** ACO(Ph) $AntRoutes \leftarrow \text{zeros}(N, T, A)$  $PhBuf, PhBC \leftarrow \text{zeros}(N, T, O)$  $GAL \leftarrow 0$ **while** STOPCRITERIA == False **do** $o \leftarrow \text{RANDOMINT}(1, O)$ **for all**  $a \in A$  **do** $AntRoutes[:, :, a] \leftarrow \text{SELECTANTROUTE}(Ph[:, :, o])$  $OB[a, :] \leftarrow \text{EVALUATEFITNESS}(AntRoutes[:, :, a])$  $PhBuf \leftarrow OB[a, :] \cdot AntRoutes[:, :, a]$  $PhBC[:, :, o] \leftarrow PhBC[:, :, o] + AntRoutes[:, :, a]$ 

UPDATEAGA(OB)

PPA(OB)

**if**  $GAL > T_{merge}$  **then** $Ph \leftarrow Ph + \frac{PhBuf}{PhBC}$  $PhBuf, PhBC \leftarrow \text{zeros}(N, T, O)$  $T_{merge} \leftarrow T_{merge} + GAL$ **procedure** SELECTANTROUTE( $Ph_o$ ) $C \leftarrow \text{NUMBERCONESELECT}$  $d \leftarrow 0$  $t \leftarrow 0$ **while** True **do** $n_s \leftarrow \text{FITPROPSELECT}(Ph_o[0 : N_x \cdot N_y])$ **if**  $BB[n_s] == 1$  **then**

Break

**while**  $t \leq T$  **do** $SchedCone \leftarrow \text{ZEROS}(N, T, C)$  $PherSum \leftarrow \text{ZEROS}(C)$ **for all**  $c \in C$  **do** $\beta \leftarrow \text{RANDOMSLOPE}(\alpha_{max})$  $k \leftarrow 1$ **while**  $BB[n_s] == 0$  **do****if**  $N(0, 1) < 0.5$  **then** $n_s \leftarrow n_s + \lfloor [N(0, 1) - 0.5] \cdot k \rfloor$ **else** $n_s \leftarrow n_s + N_x \cdot N_y \cdot \lfloor [2 \cdot N(0, 1) - 1] \cdot k \rfloor$  $SchedCone[:, :, c] \leftarrow \text{INCLUDECONEBLOCKWISE}(Chrome, \beta, d, n_s, t)$  $PherSum[c] \leftarrow \sum_{n=1}^N \sum_{t=1}^T Ph_o[n, t, c] \cdot SchedCone[n, t, c]$  $cone \leftarrow \text{FITPROPSELECT}(PherSum)$  $Chrome \leftarrow Chrome + SchedCone[:, :, cone]$ **if** CAPACITYREACHED( $Chrome$ ) **then** $t \leftarrow t + 1$  $d \leftarrow d + S_x$





# 3

## A Methodology to Establish the Optimal Blend of Geological Confidence Classes

In the previous chapter, a scheduler is developed that optimizes mine plans for GCC target blends and production targets. To use the scheduler to find the optimal GCC blend for a certain maximum allowed deviation at a given confidence level, a new methodology is developed, presented in this chapter. The idea behind the methodology is to establish a range of equally probable scenarios of metal production per period, which are optimized by the scheduler for a certain blend of GCC's. Using an exhaustive data set, the deviations from production target can be calculated, providing the relation between the GCC blend and the deviations. The required confidence level can be obtained from the scenarios, through which a line can be fitted to establish the optimum GCC blend for the maximum allowed deviation. The reason for choosing a scenario based model is that this enables one to determine the confidence level from the equally probable scenarios. However, the method is designed for cases in which no simulated model is available. Therefore, the results are verified by using an estimated resource model. The methodology is discussed in more detail below.

### 3.1. Presenting the Methodology

The methodology is shown in figure 3.3 and in algorithm 16 and 17. The **first step** of the method is the creation of artificial drilling campaigns from an exhaustive data set, which will commonly be the grade control data. The creation of these campaigns are necessary in order to have resource models with different blends of GCC's, which can be used to schedule for different target GCC blends. The workings of the drilling campaign algorithm are discussed in section 3.2. The assumption that grade control data can be used to create representative exploration campaigns is not validated here, but should be done in follow-up studies. In the **second step**, the resource models are created for the different drilling campaigns. The resource models are created using Sequential Gaussian Simulations and Ordinary Kriging. Ordinary Kriging is chosen because of the widespread use throughout the industry, as stated in section 1.2, while Sequential Gaussian Simulation is chosen because many papers are written about its application for the mining industry (Leuangthong et al., 2004; Albuquerque et al., 2014; Manchuk and Deutsch, 2012; Soltani et al., 2004). These resource modelling techniques are discussed in more detail in section 3.3. For the simulated models, the E-type estimator is calculated in order to be able to schedule using the resource model. The E-type estimator is the average of the realizations per block. The E-type estimate is also used in the **third step** to create the true representation of the deposit represented by a blockmodel from the exhaustive data set. The E-type estimator is used instead of the kriging estimator for the true resource model to prevent any possible smoothing effects of the kriging estimator, which might affect the true resource model.

Monthly mine plans are scheduled using the kriged and E-type resource models for different GCC target blends in **step 4**. The target blends should be similar to the blends which were used to create the resource models by the drilling campaign algorithm. The scheduler then gives the monthly and quarterly production figures. The more schedules created for different blends, the better the relation between the GCC's and the deviations from production target can be established. For every optimized schedule, the realized GCC blend

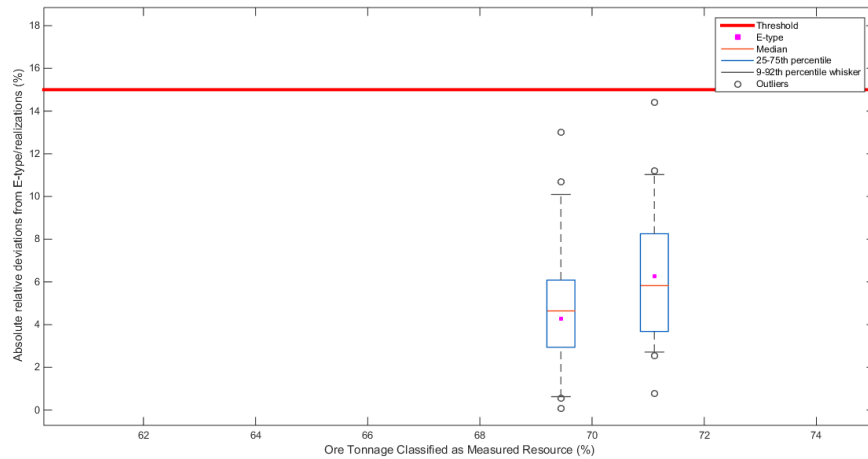


Figure 3.1: Example graph for the absolute relative deviations of the realizations per ore tonnage percentage classified as measured resources.

per period is determined in **step 5**, since this may differ from the target blend. The production figures for the realizations of the resource model are created by applying the E-type schedule and ore-waste ratio to the simulated resource models (**step 6**), which provide the metal production per period. The realizations of the resource model are not scheduled since the aim is to evaluate all equally possible production scenarios if one were to schedule for a single estimate. The true representation of the deposit is used to calculate the monthly and quarterly deviations from the expected metal production figures for all realizations and estimations (**step 7**). The deviations are expressed as the absolute relative deviations per period and calculated using equation 3.1, where  $\gamma_t$  is the real or estimate metal production in period  $t$  and  $\delta^{abs}$  the absolute relative deviations. The absolute relative deviations are chosen because one wants to find the optimal blend of GCC's for a maximum allowed deviations, which can be positive or negative. For each of these periods, a GCC blend is given. Therefore, the absolute relative deviations can be plotted against the GCC blend. The GCC blend is expressed here as the ratio of ore tonnage classified as measured with respect to the total ore tonnage. This is possible, since the GCC blend only consists of measured and indicated classes, which by definition total to a 100%.

The scenarios of the absolute relative deviations per ratio of ore tonnage that is classified as measured are plotted as boxplots, using the 9<sup>th</sup> – 92<sup>th</sup> percentile of the scenarios for the whiskers and the 25<sup>th</sup> – 75<sup>th</sup> percentile of the scenarios for the boxes. An example of such a graph is shown in figure 3.1. The x-axis shows the percentage of ore tonnage being classified as measured resources. Since the ore tonnage can be either classified as measured or indicated resources, the remainder is classified as indicated resources. The y-axis shows the absolute relative deviations from the realizations and E-type estimate. The example graph shows 2 boxplots, meaning that it represents a case in which a schedule with 2 periods is created or two schedules with one period is created. For every period, the absolute relative deviations are calculated and the ratio of ore tonnage classified as measured resources is determined. The absolute relative deviations are then plotted as boxplots against the ratio of ore tonnage classified as measured. A line showing a maximum allowed deviation is added, which can be used to determine the minimum ratio of ore tonnage classified as measured to stay under this maximum allowed deviation.

**Step 8** extracts the required confidence level from the absolute relative deviation scenarios. Meaning that if a 90% confidence level is required, the 90<sup>th</sup> percentile is used from the scenarios. From all the schedules, the confidence levels are gathered and a least-squares estimate is used to fit a line through these points (**step 9**). Because the line is not expected to intersect all data points, a confidence interval is determined from the least-squares estimate to be able to include all data points and establish a conservative optimal GCC blend. The optimal blend is determined based on the intersection of the maximum allowed deviation and the confidence interval of the least-squares estimate. An example of such a graph is shown in figure 3.2, in which some data points and their least-squares estimate are displayed. The least-squares estimate is determined based on a prototype function, of which multiple can be evaluated based on the sum squared error between

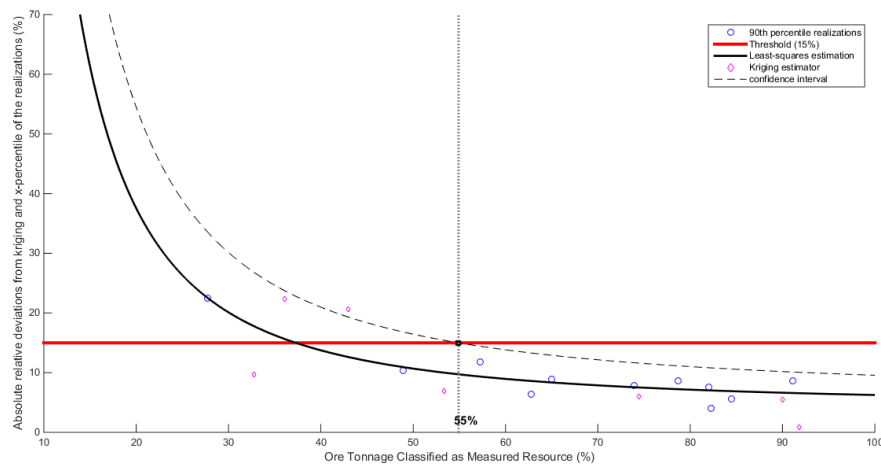


Figure 3.2: Example graph for establishing the minimum ratio of ore tonnage classified as measured for a certain maximum allowed deviation.

the line and the data points, to establish the best prototype function. As can be seen from the graph is that the least-squares estimate does not intersect all data points. Therefore, the confidence interval of the least-squares estimate is used to create a line which includes all data points. This is done to establish a conservative estimate of the minimum ratio of ore tonnage classified as measured. The intersection of this conservative line with a maximum allowed deviation, here 15%, gives the minimum ratio of ore tonnage classified as measured. The methods to create the least-squares estimate and confidence intervals are not discussed here, the interested reader is referred to Teunissen et al. (2009).

Since the approach developed in this research to meet production targets is designed to work for cases in which the resource model is created using Ordinary Kriging, the relation established by the simulations is verified by the schedules created with the Ordinary Kriging resource models (**step 10**). For this purpose, the absolute relative deviations per GCC blend, that were created with the Ordinary Kriging resource model as input for the scheduler, are also shown in figure 3.2, to verify the optimum GCC blend established by the realizations. The verification is done by looking at the portion of the graph from the minimum ratio of ore tonnage classified as measured and onwards. The percentage of the data points within this area from the schedules created with the kriged resource model that also stay below the particular maximum deviation are calculated and should be similar to the required confidence level. In the example graph, all data points within this area stay below the maximum allowed deviation, creating a 100% confidence level, albeit for a very small data set. Differences in confidence level might occur due to the smoothing effect of the Kriging operator as discussed in section 3.3, which results in a different ore-waste classification than the classification made by the E-type estimator.

The described procedure to establish the minimum ratio of ore tonnage classified as measured is performed for a case in which the ore-waste classification is made based on the estimated resource model and a for a case in which the classification is done based on the true resource model. The latter case represents the scenario in which a grade control step is added before a block is sent to the processing plant or not. The grade control step reclassifies the block as ore or waste, based on blasthole data, as discussed in section 2.1.2. It is expected that this will result in larger deviations, since entire blocks are added and removed from the production data. The reason for adding a scenario with a grade control step is because it is also done in real operations. The case with a grade control step shows to what extent the real deviations can be controlled by blending the GCC's, since the deviations established without grade control will not be the deviations obtained in a real operation.

$$\delta_t^{abs} = \text{ABS}\left(\frac{\gamma_t^{real} - \gamma_t^{estimate}}{\gamma_t^{estimate}}\right) \quad (3.1)$$

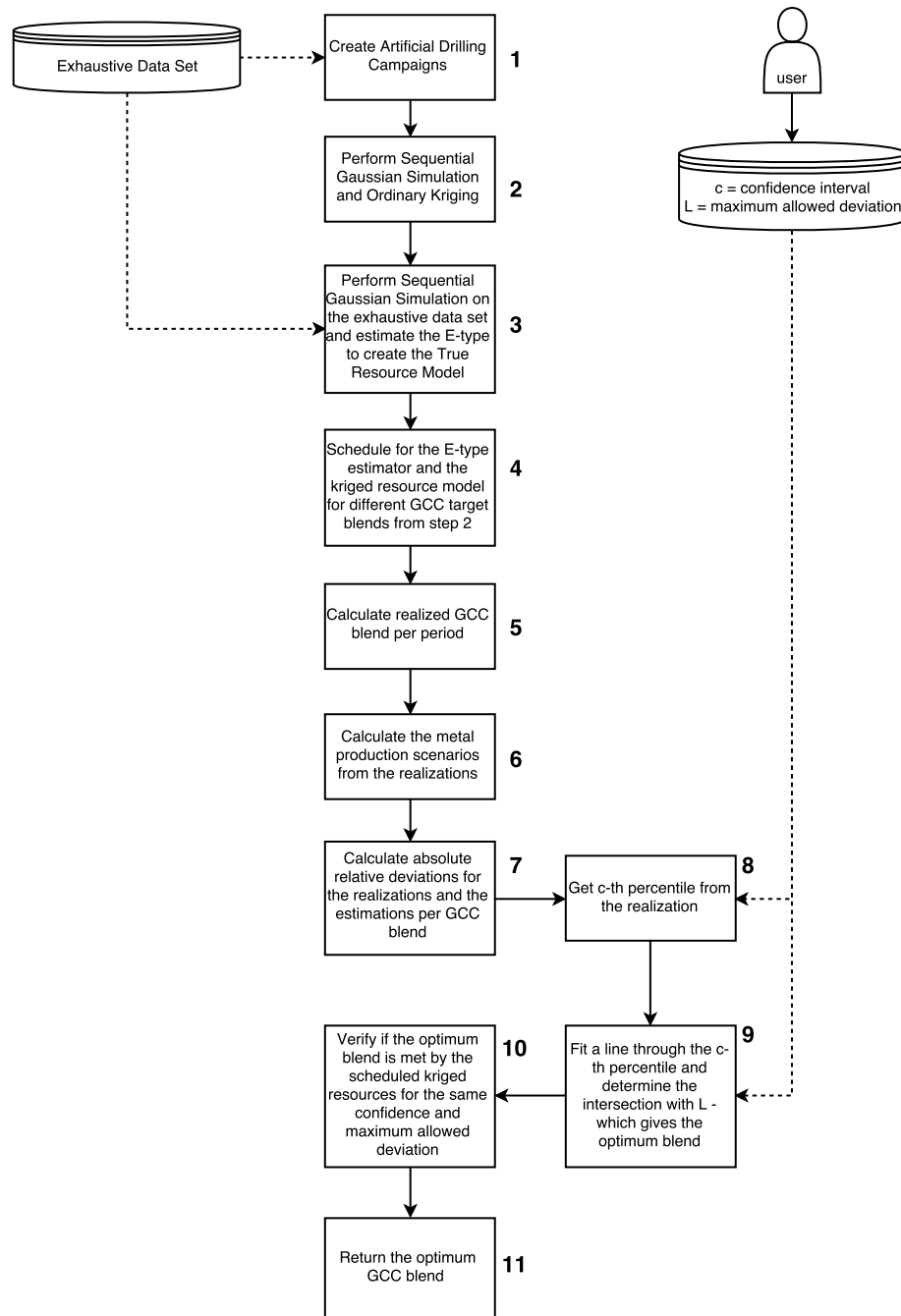


Figure 3.3: Methodology to establish the optimum Geological Confidence Class blend

### 3.2. Development of the Artificial Drilling Campaign Algorithm

Using an exhaustive data set such as grade control data, drilling campaigns are created by the artificial drilling campaign algorithm to produce scenario's of different GCC blends. This is necessary if one wants to optimize schedules for different target GCC blends. It is not possible to meet such targets for all periods if a different blend of GCC's is present. The idea behind the algorithm is to use dense infill drilling in high grade zones until the target ratio of tonnage classified as measured is met. The rest of the area is then filled with drilling according to the indicated resource class. Only measured and indicated classes are blended, because these are the ones that can be converted to reserves, as described in section 1.3. The GCC ratios for the artificial drilling campaign are based on the total coverage of a drillhole in terms of its tonnage. The coverage is determined based on the drillhole spacing associated with the GCC's and the drillhole length. This drillhole spacing gives the maximum spacing in which drillholes can be placed for the resources interpolated in between these drillholes to be classified as the corresponding Geological Confidence Class. Half of this spacing length is used as the radius of a circle around the drillhole. Together with the length of the drillhole and the density of the material, the tonnage covered by the drillhole can be calculated. The algorithm requires an exhaustive data set at a measured and indicated level spacing and a target blend ratio to plan for.

The algorithm is shown in 18 and starts with creating a two dimensional grid from the exhaustive data set, containing the x and y coordinates of the data columns, for both the measured and indicated spaced data. For every column, the length is determined, after which the tonnage is calculated, using formula 3.2.  $D_b$  is the length of column  $b$  and  $r_{GCL}$  is the range in meters of the GCL. Per column, a mean grade is calculated, for which the 2D grid is sorted in a descending manner. The columns are then iteratively added to the campaign until the target ratio is met. High grade zones are preferentially drilled using closer infill drilling than low grade zones using the sorted data grids. Lastly, 5 per cent of the drill holes are randomly removed to simulate actual drilling campaigns as closely as possible, because normally, some drill hole locations will not be reachable. It should be noted that by using the entire tonnage of the drillhole coverage to determine the GCC ratio, the actual ratio for the ore tonnage can be different.

$$T = D_b \cdot \pi \cdot \frac{r_{GCL}^2}{2} \quad (3.2)$$

**Algorithm 16** Procedure to evaluate the deviation from production target per GCL blend

---

```

1 procedure CALCULATE DEVIATIONS PER GCL
2    $GCL_{target} \leftarrow$  GCL target blends
3    $S^{exhaustive} \leftarrow$  Exhaustive data set
4    $S^{actual} \leftarrow$  ORDINARYKRIGING( $S^{exhaustive}$ )
5    $I \leftarrow$  number of realizations per simulation

6   for all  $g \in GCL_{target}$  do
7      $D \leftarrow$  CREATEDRILLINGCAMPAIGN( $GCL_{target}[g]$ ,  $S^{actual}$ )
8      $S^{OK} \leftarrow$  ORDINARYKRIGING( $D$ )
9      $S^{sim} \leftarrow$  SEQUENTIALGAUSSIANSIMULATION( $D$ )
10     $S^{Etype} \leftarrow \frac{1}{I} \sum_{i=1}^I S^{sim}$ 

11     $OW^{OK} \leftarrow$  DETERMINEOREWASTE( $S^{OK}$ )
12     $OW^{Etype} \leftarrow$  DETERMINEOREWASTE( $S^{Etype}$ )
13     $OW^{actual} \leftarrow$  DETERMINEOREWASTE( $S^{actual}$ )

14     $Chrome^{Etype} \leftarrow$  SCHEDULER( $S^{Etype}$ ,  $g$ )
15     $Chrome^{OK} \leftarrow$  SCHEDULER( $S^{OK}$ ,  $g$ )

16     $P^{Etype} \leftarrow$  CALCULATEMETALPRODUCTION( $Chrome^{Etype}$ ,  $S^{Etype}$ ,  $OW^{Etype}$ )
17     $P^{OK} \leftarrow$  CALCULATEMETALPRODUCTION( $Chrome^{OK}$ ,  $S^{OK}$ ,  $OW^{OK}$ )

18    for all  $i \in I$  do
19       $P_i^{sim} \leftarrow$  CALCULATEMETALPRODUCTION( $Chrome^{Etype}$ ,  $S_i^{sim}$ ,  $OW^{Etype}$ )

20     $\gamma^{actual,gc,OK} \leftarrow$  CALCULATEMETALPRODUCTION( $Chrome^{OK}$ ,  $S^{actual}$ ,  $OW^{actual}$ )
21     $\gamma^{actual,wgc,OK} \leftarrow$  CALCULATEMETALPRODUCTION( $Chrome^{OK}$ ,  $S^{actual}$ ,  $OW^{OK}$ )
22     $\gamma^{actual,gc,Etype} \leftarrow$  CALCULATEMETALPRODUCTION( $Chrome^{Etype}$ ,  $S^{actual}$ ,  $OW^{actual}$ )
23     $\gamma^{actual,wgc,Etype} \leftarrow$  CALCULATEMETALPRODUCTION( $Chrome^{Etype}$ ,  $S^{actual}$ ,  $OW^{Etype}$ )

24     $\delta_g^{abs,gc,OK} \leftarrow$  ABS( $\frac{\gamma^{actual,gc,OK} - \gamma^{OK}}{\gamma^{OK}}$ )
25     $\delta_g^{abs,wgc,OK} \leftarrow$  ABS( $\frac{\gamma^{actual,wgc,OK} - \gamma^{OK}}{\gamma^{OK}}$ )
26     $\delta_g^{abs,gc,Etype} \leftarrow$  ABS( $\frac{\gamma^{actual,gc,Etype} - \gamma^{Etype}}{\gamma^{Etype}}$ )
27     $\delta_g^{abs,wgc,Etype} \leftarrow$  ABS( $\frac{\gamma^{actual,wgc,Etype} - \gamma^{Etype}}{\gamma^{Etype}}$ )

28    for all  $i \in I$  do
29       $\delta_{g,i}^{abs,wgc,sim} \leftarrow$  ABS( $\frac{\gamma^{actual,wgc,Etype} - \gamma_i^{sim}}{\gamma_i^{sim}}$ )
30       $\delta_{g,i}^{abs,gc,sim} \leftarrow$  ABS( $\frac{\gamma^{actual,gc,Etype} - \gamma_i^{sim}}{\gamma_i^{sim}}$ )

31     $GCL_g^{OK} \leftarrow$  DETERMINEGCLPERPERIOD( $Chrome^{OK}$ )
32     $GCL_g^{Etype} \leftarrow$  DETERMINEGCLPERPERIOD( $Chrome^{Etype}$ )

```

---

**Algorithm 17** Procedure to determine the minimum GCL blend

---

```

1 procedure DETERMINE MINIMUM GCL BLEND
2    $k \leftarrow$  Confidence level
3    $L \leftarrow$  Deviation limit

4    $\delta_{90th}^{abs, wgc, sim} \leftarrow$  K-THPERCENTILE( $\delta_i^{abs, wgc, sim}$ )
5    $\delta_{90th}^{abs, gc, sim} \leftarrow$  K-THPERCENTILE( $\delta_i^{abs, gc, sim}$ )

6    $F_{wgc}^\delta \leftarrow$  LEASTSQUAREESTIMATE( $GCL^{Etype}, \delta_{90th}^{abs, wgc, sim}$ )
7    $F_{gc}^\delta \leftarrow$  LEASTSQUAREESTIMATE( $GCL^{Etype}, \delta_{90th}^{abs, gc, sim}$ )

8    $\varphi_{wgc} \leftarrow$  LEASTSQUARESCONFIDENCE( $F_{wgc}^\delta$ )
9    $\varphi_{gc} \leftarrow$  LEASTSQUARESCONFIDENCE( $F_{gc}^\delta$ )

10   $R_{wgc} \leftarrow$  INTERSECT( $\varphi_{wgc}, L$ )
11   $R_{gc} \leftarrow$  INTERSECT( $\varphi_{gc}, L$ )

```

---

### 3.3. Principles of Resource Simulation and Estimation

Resource estimation is the process of defining and modelling the Mineral Resources and its related uncertainty. A Resource model is a simplification of the reality, derived from partial data and incomplete background knowledge of the mineralisation (Vann et al., 2012). The tasks required for a well performed estimation are:

- Geological Interpretation**    The geologist must explain the geological interpretation to the geostatistician. Errors resulting from incorrect and inappropriate interpretation can be orders of magnitude larger than errors associated with grade estimation (Vann et al., 2012). However, the process of geological interpretation and its uncertainties are not further discussed here. The interested reader is referred to Cowan (2012).
- Smallest Mineable Unit**     The Mining engineer must consider the Smallest Mineable Unit (SMU) for the block size, discussed in section 2.1.1.
- Geostatistical Estimation**   The Geostatistician must explain in a clear and detailed manner the geostatistical analysis and the estimation or simulation method with the results to the Mining Engineer and the geologist, to ensure that the relevant mining and geological aspects were taken into account by the geostatistician.

The geostatistical estimation is discussed for Ordinary Kriging and Sequential Gaussian Simulation. The principles of geostatistical analysis required for the estimations and simulations are not given here. The interested reader is referred to Teunissen et al. (2009). The section closes with a discussion about the difference between the ordinary kriging and E-type estimator, since these two estimators are used in the scheduler.

#### 3.3.1. Ordinary Kriging

Ordinary Kriging is a spatial linear interpolation method, which generates best linear unbiased estimates, based on regression against observed values of surrounding data points, weighted according to spatial covariance values (Bohling, 2005). The unbiasedness property means that the average of the estimation error will be zero. Furthermore, the best property is obtained by minimizing the variance of the errors. These two formulations are used by a least-squares estimation to determine the weights for the linear estimation. A derivation of the kriging estimator is not given here, but can be found in Remy et al. (2011). Kriging recognizes two components for the estimation, a trend component, the mean, and a residual component. The Kriging methods estimate the residual component at the required grid points based on the weights obtained by the

**Algorithm 18** Artificial Drilling Campaign

---

```

1 procedure CREATE ARTIFICIAL DRILLING CAMPAIGN
2    $M_{data}, I_{data} \leftarrow \text{ASSIGNEXHAUSTIVEDATA}$  ▷ M is measured; I is Indicated
3    $Remove \leftarrow 0.05$ 
4    $M_{target}, I_{target} \leftarrow \text{SETTARGETRATIO}$ 
5    $D \leftarrow \text{SETINTERVALLENGTH}$ 
6    $M_{range}, I_{range} \leftarrow \text{SETCONFIDENCERANGE}$ 

7    $Grid_M, Grid_I \leftarrow \text{CREATE2DGRID}(M_{data}, I_{data})$ 
8    $Grid_{depth,M}, Grid_{depth,I} \leftarrow \text{DETERMINEHOLEDEPTH}(M_{data}, I_{data})$ 
9    $Tonnage_M, Tonnage_I \leftarrow \text{CALCULATEHOLETONNAGE}(Grid_{depth,M}, Grid_{depth,I}, M_{range}, I_{range})$ 
10   $T \leftarrow \text{SUM}(Tonnage_M)$ 
11   $MeanGrade_M, MeanGrade_I \leftarrow \text{CALCULATEMEANHOLEGRADE}(M_{data}, I_{data})$ 

12   $Grid_M, Grid_I, Tonnage_M, Tonnage_I \leftarrow \text{SORTHOLESFORGRADE}(MeanGrade_M, MeanGrade_I)$ 
13   $T_M, T_I, i_M, i_I, r_M, r_I \leftarrow 0$ 
14   $S_M \leftarrow \text{SIZE}(Drill_M)$ 
15   $S_I \leftarrow \text{SIZE}(Drill_I)$ 
16  while  $T_M < T \cdot M_{target}$  do
17     $T_M \leftarrow T_M + Tonnage_M[i_M]$ 
18     $i_M \leftarrow i_M + 1$ 
19   $Drill_M \leftarrow M_{data} \in Grid_M[0 : i_M]$ 
20  for all  $h \in Drill_M$  do
21     $index \leftarrow \text{FIND}(Grid_I \in Drill_M[h] \pm M_{range})$ 
22     $Grid_I[index] \leftarrow []$ 
23  while  $S_M \cdot Remove > r_M$  do
24     $s_M \leftarrow \text{SIZE}(Drill_M)$ 
25     $index \leftarrow [N(0, 1) \cdot s_M]$ 
26     $Drill_M[index] = []$ 
27     $r_M \leftarrow r_M + 1$ 
28  while  $T_I < T \cdot I_{target}$  do
29     $T_I \leftarrow T_I + Tonnage_I[i_I]$ 
30     $i_I \leftarrow i_I + 1$ 
31   $Drill_I \leftarrow I_{data} \in Grid_I[0 : i_I]$ 
32  while  $S_I \cdot Remove > r_I$  do
33     $s_I \leftarrow \text{SIZE}(Drill_I)$ 
34     $index \leftarrow \text{RANDOM}(0, s_I)$ 
35     $Drill_I[index] = []$ 
36     $r_I \leftarrow r_I + 1$ 

```

---



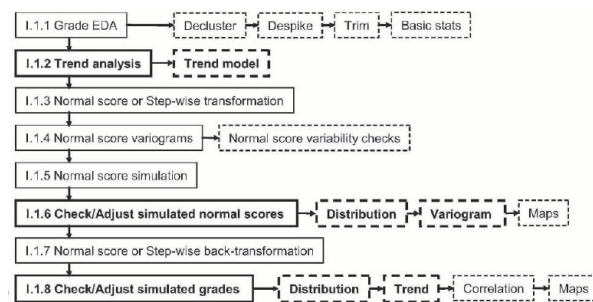


Figure 3.4: Practical procedure for sequential gaussian simulations (Nowak and Verly, 2004)

least-squares estimation (Bohling, 2005). Ordinary Kriging assumes a constant mean in the local neighbourhood of each estimation (Li and Heap, 2008).

### 3.3.2. Sequential Gaussian Simulation

Simulation methods are designed to create realizations that reproduce the statistical properties of the original data, such as the mean, variance and the spatial correlation, as given by the semi-variogram (Safikhani et al., 2016). Sequential gaussian simulation simulates the distribution function at the grid locations sequentially, conditional to the data values and the previously simulated realizations. Realizations are created by drawing random values from these distributions. The method requires normally distributed data values in order to create the distribution functions. The distribution functions are created using the Kriging estimator for the mean and the kriging variance for the variance. The mathematical background and workings of sequential gaussian simulation are not discussed here. The interested reader is referred to Remy Remy et al. (2011). Instead, the practical procedure to perform sequential gaussian simulations is given.

The procedure for sequential gaussian simulations is described by Nowak and Verly (2004) and shown in figure 3.4. The first step of declustering is required to remove the effects of preferential sampling of high grade zones, which changes the distribution of the grades. However, declustering techniques are not discussed here, because the true grade distribution is provided by the exhaustive data set. Therefore, despiking and trimming of the data set is also not considered here. The trend can be analyzed by calculating the moving average for the grid points. If a trend is present, the data can be divided into a trend and residual component. The residual component is then normally transformed conditional to the trend (Leuangthong et al., 2004). Based on the normally transformed data, the semi-variogram is computed to determine the spatial correlation in the gaussian domain. The realizations are evaluated for their distribution, which should be close to the normal distribution and their spatial relation, which should be similar to the spatial relation of the normal transformed data. The realizations are back-transformed to the original (declustered) data distribution. These values should have a similar distribution as the original distribution. The average value of the realizations per grid point is called the E-type estimate.

### 3.3.3. Comparing Ordinary Kriging and the E-type Estimator

The linear estimation technique used by Ordinary Kriging results in a reduced variability in the estimated data, causing a smoothing effect, in which small samples are overestimated and large samples underestimated, hereby not reproducing the histogram or the spatial variability (Yamamoto, 2005). Sequential Gaussian Simulation represents the spatial distribution more realistic than the kriging estimator, by creating equally probable scenarios, which all stay true to the sample statistics. The E-type estimator calculates the average of the created realizations per block. This differs from the kriging estimator, since it not directly tries to estimate a local mean, but rather a block-mean, from a distribution that conforms the global sample statistics. The differences between the two types of estimation techniques can be large if a cut-off grade is applied, since the kriging estimator tends to over- or underestimates the grades more heavily than the E-type estimator.



# 4

## Case Study

### 4.1. An Introduction to the Yanacocha Case

The Yanacocha deposit is a world class gold deposit in South America, consisting of mainly low grade Au-Cu Porphyry deposits, from which over 15 Moz gold have been mined to date. The deposit considered here is a depleted open pit, from which the blast-hole data is collected to create an exhaustive data set of the deposit. The open pit operation was halted due to the high level of sulphidization, affecting the heap leaching process. Three exhaustive data set are given in a 25x25x5m grid, 60x60x5m grid and 120x120x5m grid, representing the measured, indicated and inferred resource models, respectively. Within the data set, a smaller test area is selected for the case study. A blockmodel is created with 12696 blocks of 25x25x10m, of which 7482 blocks are not air. For the deposit, a cut-off grade of 0.023oz/t is calculated using formula 2.1. For the mining operation, the following parameters are assumed:

$C_{mine,t}$	40 Mt/year
$C_{proc,t}$	10 Mt/year
$R$	84.7 %

For the case study, a maximum allowed absolute monthly and quarterly deviation is set to 15% at a 90% confidence level, to show the use of the methodology to establish an optimal GCC blend. The following sections describe the workflow of the methodology as shown in figure 3.3. It is chosen not to set a gold production target to ease the scheduling process. The gold production figures produces by the scheduler are seen as the targets, from which the deviations are calculated.



Figure 4.1: Location of the Yanacocha mining district in Peru (Hennessy, 2005).

## 4.2. Step 1 - Creating Artificial Drilling Campaigns

Seven drilling campaigns are made, with a blend of 20, 30, 50, 60, 70 and 80% of tonnage material classified as measured resources. The campaigns are shown in figure 4.2. The percentages are chosen to evaluate the entire spectrum of target blends. The blocks that are classified as inferred resources due to the random removal of drillholes are assigned as waste blocks.

## 4.3. Step 2 - Performing Ordinary Kriging and Sequential Gaussian Simulations

### Ordinary Kriging

For Ordinary Kriging, the spatial correlation of the exhaustive data is used, resulting in the variogram as shown in 4.3 for the direction of greatest spatial correlation. The variogram is fitted using three structures, with the parameters as shown below. The Kriging results are shown in figure 4.4.

- Anisotropy parameters

- Plunge:  $0^0$
- Azimuth:  $160^0$
- Dip:  $10^0$
- Major/Semi-Major: 1.182
- Major/Minor: 1.374
- Nugget: 0.11

- Model

– Structure	<b>1</b>	<b>2</b>	<b>3</b>
– Type	Spherical	Spherical	Spherical
– Sill (Contribution)	0.16	0.1	0.94
– Range (max)	42 m	93 m	324 m
– Range (Medium)	35 m	78 m	274 m
– Range (min)	30 m	68 m	235 m

### Sequential Gaussian Simulation

A representative histogram of the selected test area is made using the exhaustive data set, hereby removing the need for declustering algorithms. The trend is analyzed using a moving average ellipsoid. The ellipsoid has the same axis lengths as the maximum structure of the variogram model. The XZ and YZ cross sections crossing the centre are shown in figure 4.5. A minor trend is visible in the XZ cross section towards the centre of the deposit. However, this trend is not removed from the data.

The normal-score transformation is done with the program SGeMS, using the representative grade distribution of the selected area. The results of these normal transformations are shown in figure 4.6. The spatial correlation is determined for the normalized exhaustive data set using the semi-variogram. It is assumed that this is representative for all the normal score transformations. The variogram in the principle direction is shown in figure 4.7. The experimental variogram is fitted using one structures as given below:

- Anisotropy parameters

- Plunge:  $0^0$
- Bearing:  $180^0$
- Dip:  $0^0$

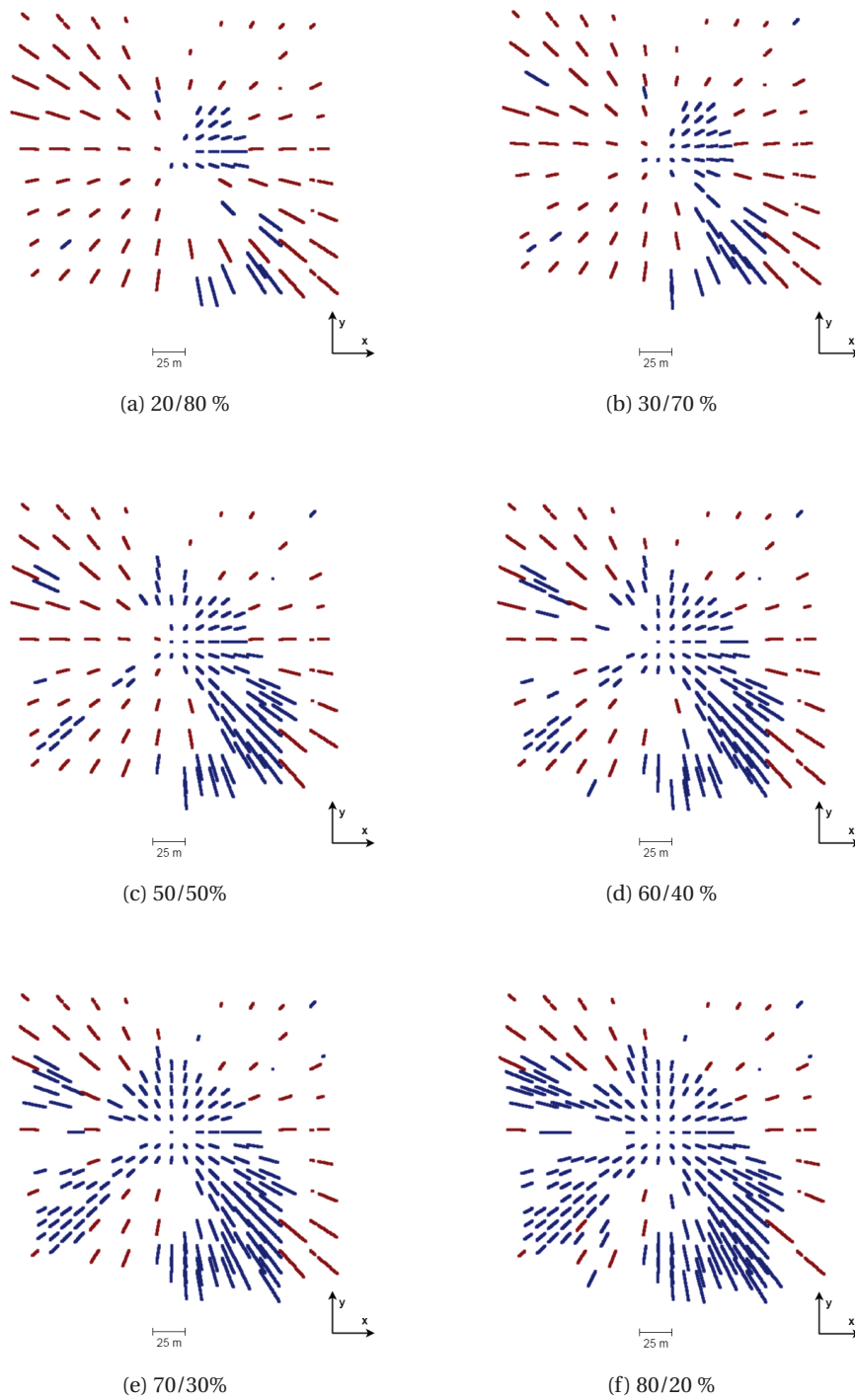


Figure 4.2: Drilling campaigns for different ratios of tonnage classified as Measured (blue) and Indicated (red) - isometric projection from top.

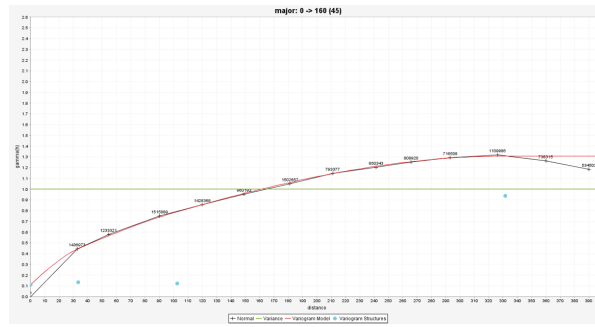


Figure 4.3: Experimental and theoretical variogram of the exhaustive data set in the direction of greatest spatial correlation.

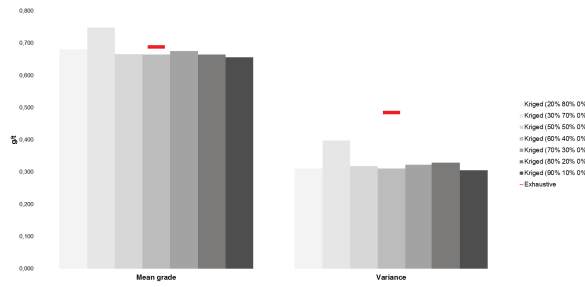


Figure 4.4: Mean and variances of the resource models using ordinary kriging

- Nugget: 0.3
- Model
  - Structure **I**
  - Type Spherical
  - Sill (Contribution) 0.7
  - Range (max) 405
  - Range (Medium) 270 m
  - Range (min) 261 m

For the simulations, 200 realizations are drawn per drilling campaign, of which the average mean and variance are shown in figure 4.8. A random realization is chosen per drilling campaign for the comparison of the variogram against the input variogram. These are shown in figure 4.9. The normally distributed realizations are back-transformed to the original, representative grade distribution. The average mean and variance of these back-transformations are shown in figure 4.10.

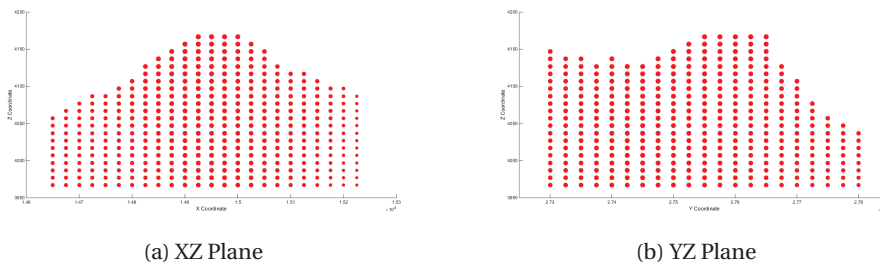


Figure 4.5: Cross sections of the moving average, shown as circles that grow larger with an increasing average gold grade.

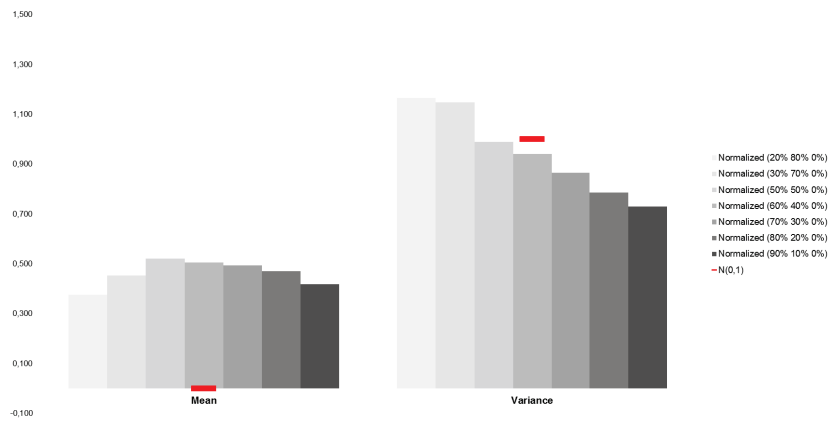


Figure 4.6: Mean and variance per drilling campaign after the normal-score transformation.

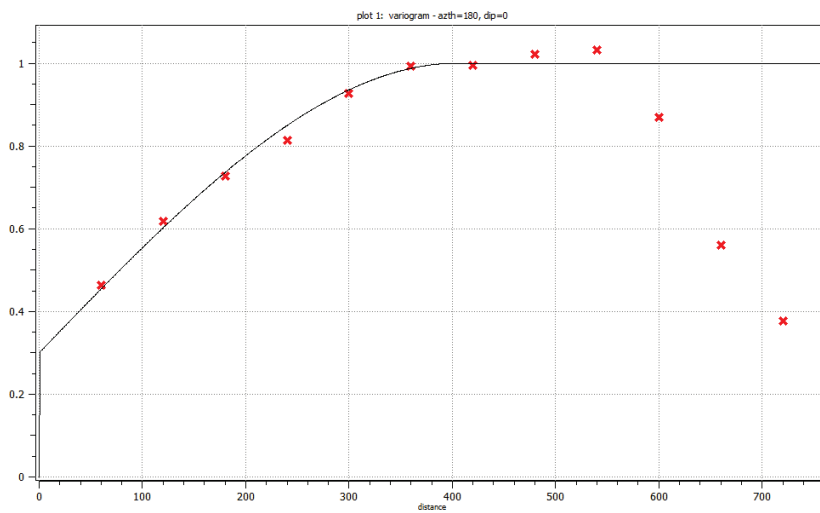


Figure 4.7: Experimental and theoretical variogram of the normal transformed exhaustive data set in the direction of greatest spatial correlation.

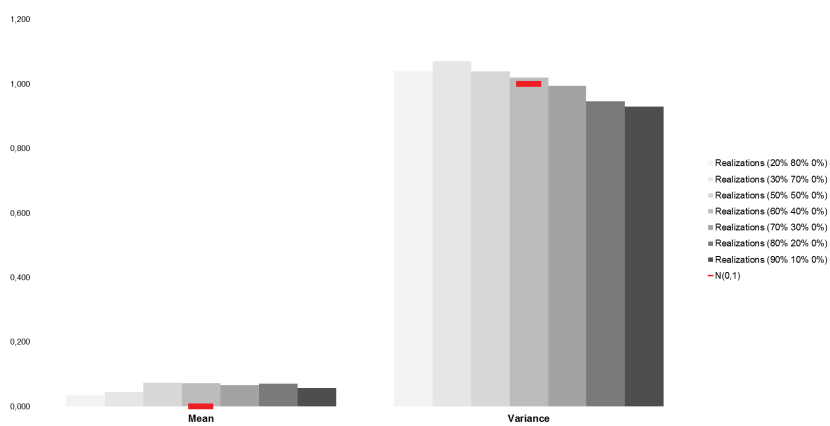


Figure 4.8: Average mean and variance of the realizations per drilling campaign.

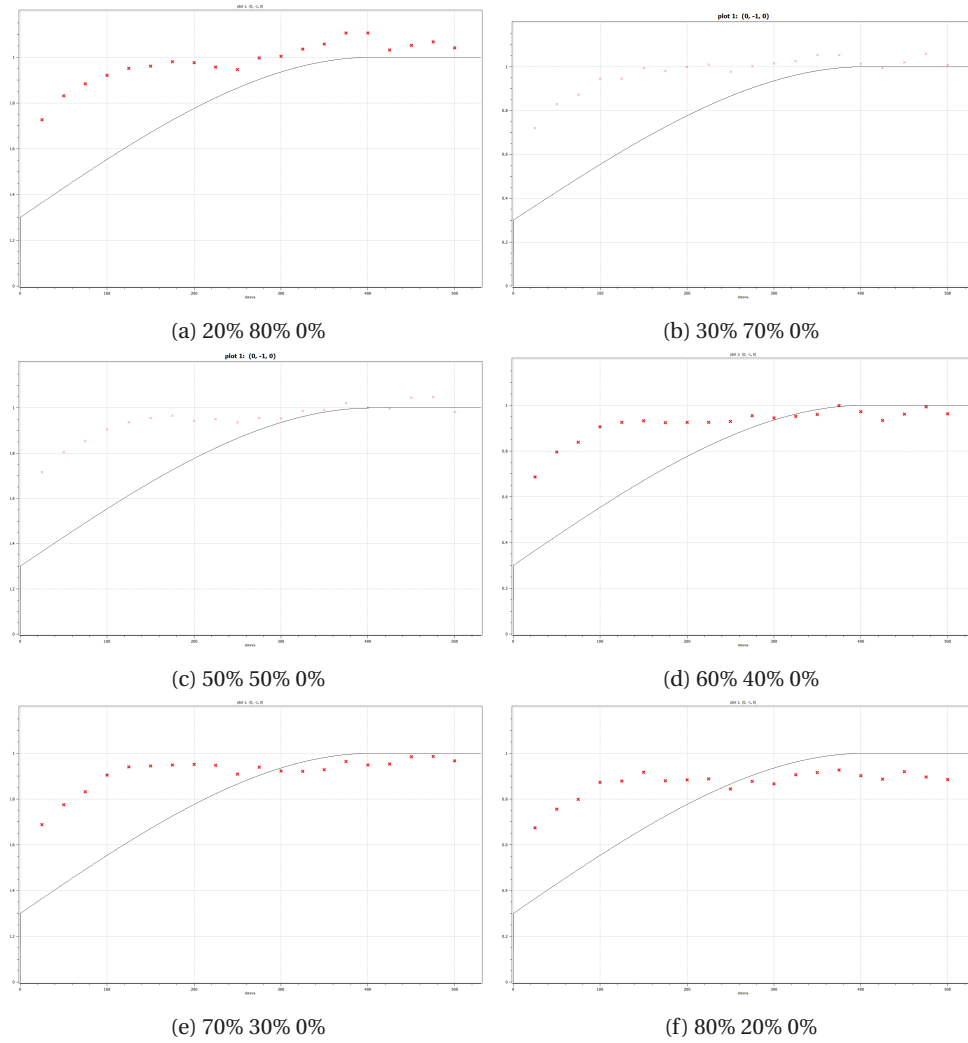


Figure 4.9: Variograms of the realizations compared against the input variogram model.

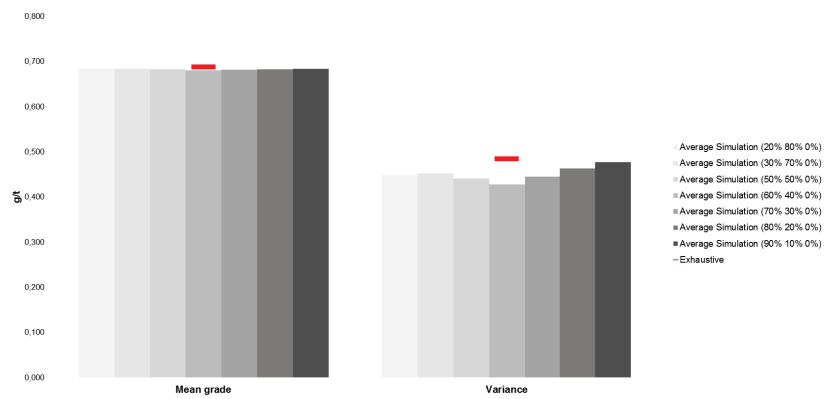


Figure 4.10: Average mean and variance of the back-transformed realizations per drilling campaign.



Drilling Campaign (Ratio measured/indicated classified ore tonnage)	Target ratio of ore tonnage classified as measured resources	Average realized ratio of ore tonnage classified as measured resources (Etype)	Average realized ratio of ore tonnage classified as measured resources (Kriging)
20%/180%	30.0%	44%	48%
20%/180%	40.0%	44%	46%
20%/180%	45.0%	44%	46%
30%/170%	45.0%	58%	63%
30%/170%	50.0%	60%	59%
30%/170%	55.0%	62%	64%
30%/170%	60.0%	62%	61%
30%/170%	65.0%	70%	65%
30%/170%	70.0%	77%	69%
50%/150%	45.0%	87%	78%
50%/150%	60.0%	75%	78%
50%/150%	62.5%	75%	77%
50%/150%	65.0%	76%	77%
50%/150%	70.0%	77%	80%
50%/150%	72.5%	80%	81%
60%/140%	75.0%	82%	88%
60%/140%	77.5%	84%	88%
60%/140%	80.0%	84%	87%
60%/140%	82.5%	84%	88%
60%/140%	85.0%	84%	89%
60%/140%	87.5%	84%	89%
70%/130%	90.0%	88%	90%
80%/120%	95.0%	93%	97%

Figure 4.11: Comparison of the created schedules in terms of the target ratio of the ore tonnage classified as measured resources.

#### 4.4. Step 3 - Creating the True Resource Model

From the exhaustive data set, the true resource model is created by the E-type estimator for 200 simulations. The mean and variance of the exhaustive data set are 0.69 and 0.48, respectively. The normalized exhaustive data set has a mean of 0.00 and a variance of 0.99. The same variogram (figure 4.7) is used for the Sequential Gaussian Simulation. The average mean and variance of the realizations are  $-0.09$  and  $1.03$ , while the mean and variance of the back-transformations are 0.69 and 0.47, respectively.

#### 4.5. Step 4-7 - Establishing Relation Between GCC Blends and Deviations

The resource models created with the different drilling campaigns are scheduled for different target blends. Scheduling is done for 18 month periods, with the production targets as discussed in the introduction of the case. The GCC targets with the average realized ratio of ore tonnage classified as measured resource per schedule is shown in figure 4.11. The targets are chosen to create a large spread in realized GCC blend for all resource models. The realized production figures are not shown due to confidentiality reasons. The absolute relative deviations of the realizations and E-type estimate are shown in figure 4.13. An example of a created schedule is shown in figure 4.12.

The performance of the scheduler is measured as the time it needs on average to create one single schedule and evaluate this schedule for its feasibility in terms of the constraints and its performance in terms of the objective functions and the Damped Pareto Prevalence value. One schedule consists of roughly 134.000 variables, excluding the air blocks which are not considered as variables by the program. The average performance is shown below:

**Seed phase**                    7 seconds / schedule

**Exploration phase**        3 seconds / schedule

**Exploitation phase**       8 seconds / schedule

#### 4.6. Step 8-10 - Establishing the Optimal Blend of GCC's

A 90% confidence level is chosen for the accuracy level for this case study, therefore, the 90<sup>th</sup> percentile of the realizations are gathered and fitted using a least-squares estimate with the prototype function as shown in equation 4.1. Only data points with a minimum ratio of 50% ore tonnage classified as measured resources are fitted with a least-squares estimate, since not enough data points are available for the lower ratios. A confidence interval is chosen for the least-squares fit to include all data points. The intersection of this confidence interval with the maximum allowed deviation of 15% set for this case study gives the optimal GCC blend. These optimal blends in terms of the ratio of ore tonnage classified as measured resource are shown on the graphs if the intersection exists. The absolute relative deviations per GCC blend from the schedules created

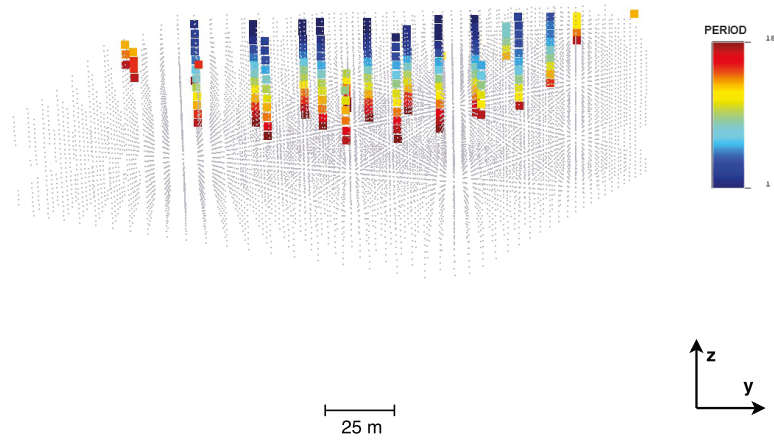
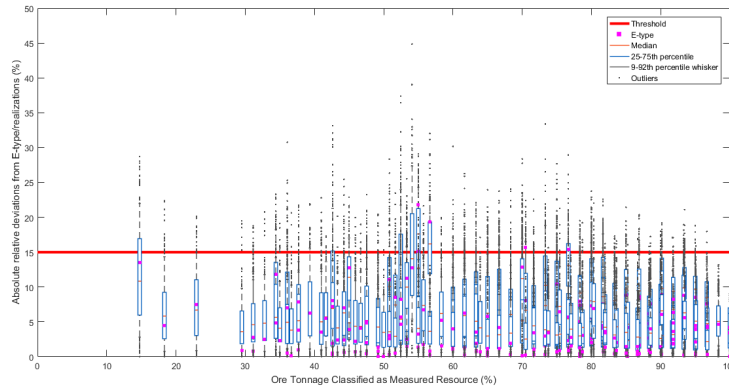


Figure 4.12: Schedule created for a target ratio of 50% in terms of the ore tonnage classified as measured resources.

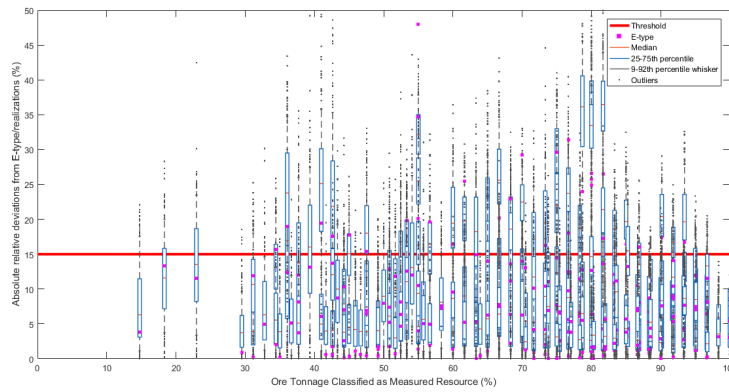
with the kriged resource models are also shown on these graphs to verify the optimal GCC blend.

It is only possible to establish an optimal GCC blend for the quarterly periods, since the maximum allowed deviation of 15% appears to be too low for the monthly cases. For this quarterly case without grade control, the optimal GCC blend is set at 54% ore tonnage classified as measured resources. The number of periods from schedules created using the kriging estimated resource model with a higher ratio than the optimal blend is 134, of which 6 periods are above the maximum allowed deviation. This means that 95.5% of the periods from the kriging estimated resource model obey the optimal GCC blend.

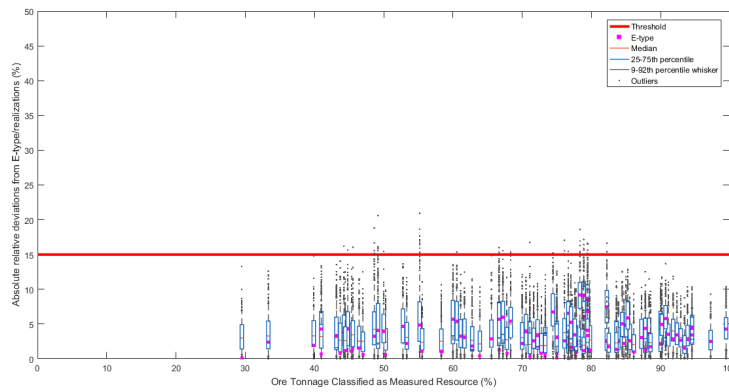
$$y = a \cdot \frac{1}{x^2} + b \cdot \frac{1}{x} + c \quad (4.1)$$



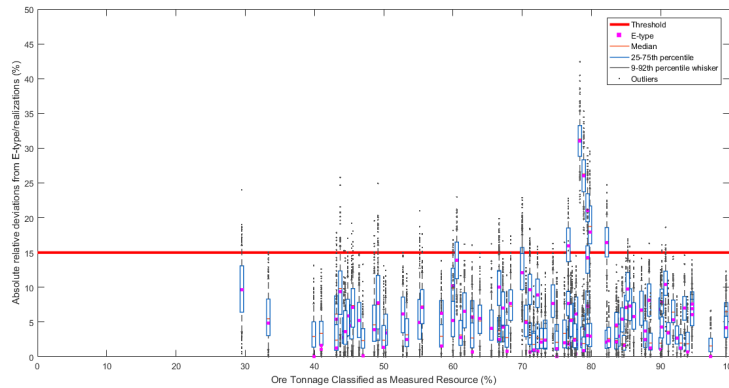
(a) Monthly periods without grade control



(b) Monthly periods with grade control

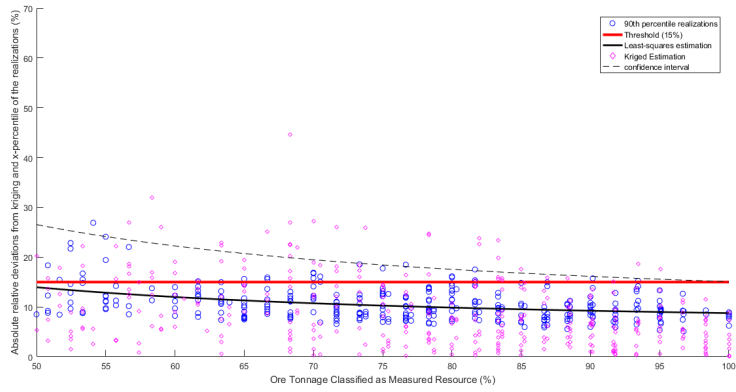


(c) Quarterly periods without grade control

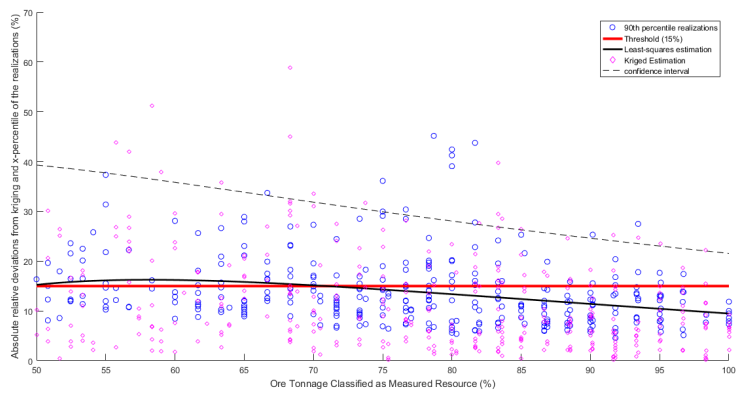


(d) Quarterly periods with grade control

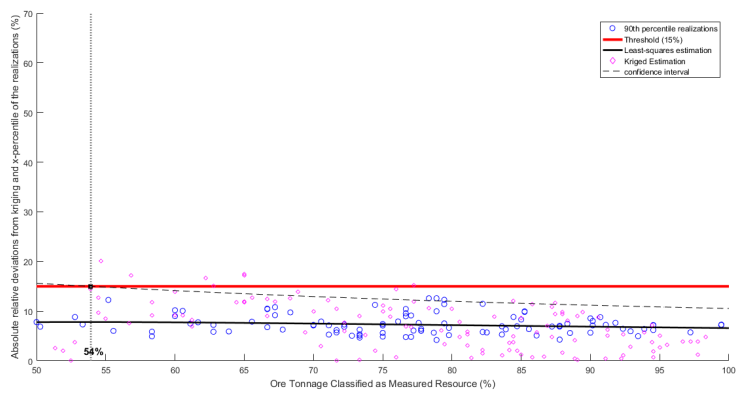
Figure 4.13: Combined absolute relative deviations from Etype/realizations



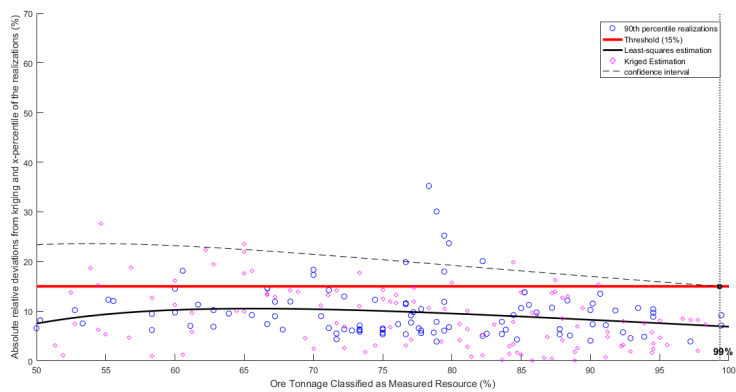
(a) Monthly periods without grade control



(b) Monthly periods with grade control



(c) Quarterly periods without grade control



(d) Quarterly periods with grade control

Figure 4.14: Graphs to Determine and Verify the Optimal GCC Blend.



# 5

## Discussion

### **Drilling Campaign**

The drilling campaigns are created with grade control data because different campaigns need to be created with different GCC blends. The assumption that this method created representative drilling campaigns is not validated, since the original exploration data set of the Yanacocha mine is not available. Future studies should investigate whether representative campaigns can be made from grade control data.

The drilling campaign algorithm effectively takes into account different drill-hole lengths to create a campaign from an exhaustive data set, however, it is unable to create a campaign for a GCC blend based on the ore tonnage. Furthermore, the algorithm cannot consider the ultimate pit, hereby taking into account blocks for the ratio which will not be mined. The result is that the realized GCC ratios of the schedules are significantly different from the apparent ratios provided by the campaign algorithm. Therefore, different target blends compared to the blend provided by the drilling campaigns are set for the scheduler. This will not affect the approach to establish the optimal blend, since the goal is to produce schedules with different realized GCC blends, which is still achieved with the drilling campaigns. The drilling campaign algorithm can be improved in the future to take into account the ratio of ore tonnage classified as measured and indicated resources and to take into account the ultimate pit.

### **Resource Modelling**

Ordinary Kriging is performed based on the spatial correlation of the exhaustive data set, since it is assumed that this is true spatial correlation within the deposit, determined with over 165 thousand points. The mean grade of the resource models created using Ordinary Kriging shows a good fit with the mean grade of the exhaustive data set, except for the model created for the drilling campaign with 30% tonnage classified as measured resources. This might be caused by the preferential denser drilling of high grade zones, which for this campaign results in an too large biased. The variances of the resource models created using Ordinary Kriging are significantly lower than the variance of the exhaustive data set. This is probably caused by the smoothing effect of the Kriging operator, which tends to overestimate small samples and underestimate large samples. The drilling campaign with 30% tonnage classified as measured resources forms the exception again, showing a variance that is closer to the variance of the exhaustive data set. An explanation other than the earlier given possible explanation of the higher mean is not found here, since the procedure to create the drilling campaigns and to perform the kriging operation is the same for all campaigns.

Sequential Gaussian Simulation (SGS) is also based on the spatial correlation of the normal transformed exhaustive data set for the same reason as for the Kriging operator. The normal score transformation of the drilling drilling campaigns show a non-normal distribution, because the distribution used for the transformation is from the exhaustive data set. The mean grade of the campaigns is much higher than the mean grade of the exhaustive data set, due to the preferential denser drilling in high grade zones. Transforming this with the exhaustive set, which has a lower mean, results in a higher mean in the Gaussian space. The realizations from the simulation show a normal distribution again, since the bias of high grade zones is removed once all grid points are simulated. 200 realizations are drawn for the simulations, which are assumed

to be sufficient to represent the true grade distribution of the resource model and sufficient to determine the 90<sup>th</sup> percentile from these realizations to determine the confidence level. The spatial correlation of the realizations in Gaussian space are off from the true spatial correlation. This might be caused by a poorly fitted variogram model. The sill is well represented by the semi-variograms, but not the range and the nugget effect. The back-transformations from the Gaussian space show a similar mean and variance as the exhaustive data set.

In order to calculate the actual production, a true representation of the resource model based on the exhaustive data set is created. This results in the fact that the true representation of the resource model is only known with a certain accuracy. SGS produces 200 equally probable resource models from the exhaustive data set, from which the E-type estimate is chosen as the true representation. However, all 200 scenarios are equally probable. Therefore, to be precise, all scenarios should be used to determine all possible deviations from the target production. However, this would lead to far too many results to comprehend, because then for every realization of the resource model from the drilling campaigns, 200 possible deviations can be calculated, resulting in a total of 40.000 possible deviations per period. Therefore, the E-type is used, while realizing that this is not the only possible true representation of the deposit.

### **Scheduler**

The scheduler is able to produce a feasible schedule with over 134.000 variables within eight seconds. Feasibility in all three stages is guaranteed by the heuristic procedure. However, the diversity of the schedules is limited due to the use of a cone algorithm, making it only possible to mine in circle-like shapes. This could be improved in further research by using an ellipsoid template. More complex designs can be achieved by combining multiple ellipsoids or cones. Another improvement would be to let the ants decide per block which period it should be mined in order to achieve more diverse and complex designs. For this approach, the challenge is to enforce feasible schedules to be created. The creation of multiple pits is not supported by the scheduler. However, for short-term production schedules, these are not preferred designs. The performance of the scheduler in terms of the convergence towards the global optimum and the movement of the Pareto front during the optimization are not discussed here. Further research should evaluate the performance of the scheduler.

### **Controlling Deviations by Blending Geological Confidence Classes**

Both the monthly and quarterly absolute relative deviations calculated without grade control as shown in figure 4.13 show a downwards trend for an increasing ratio of ore tonnage classified as measured resources. This trend is especially visible for the monthly deviations, from a ratio of 50% ore tonnage classified as measured resources. Data points with a lower ratio do not clearly show this trend. A possible reason for not showing this trend is that not enough schedules are created for these lower ratios. The quarterly periods show lower absolute relative deviations, since the monthly periods are averaged, which results in averaged deviations. The trend between the ratio of ore tonnage classified as measured resources and the absolute relative deviations is less clear than for the monthly periods, but do show that for the higher ratios, lower deviations are obtained. These results show that it is possible for the Yanacocha case study to control the deviations from production targets by blending the Geological Confidence Classes.

The absolute relative deviations calculated with grade control also show a downwards trend, albeit for higher deviations. These higher deviations are caused by creating a new ore-waste classification based on the true representation of the resource model, which results in blocks being classified as ore, which are waste blocks according to the resource model created for a drilling campaign, and vice versa. Therefore, entire blocks are added and removed to establish the actual production figures, resulting in larger deviations. The significantly higher deviations around 80% ore tonnage classified as measured resources are probably caused by a high amount of misclassified blocks. The deviations around the higher ratios of ore tonnage classified as measured resources are lower than at the lower ratios, showing that even for the cases which take into account misclassified blocks, the deviations from production targets for the Yanacocha case study can be controlled by blending the Geological Confidence Classes.

It is important to stress the fact that the results are obtained by doing a case study on one mine. These results only indicate the possibility to control the deviations from production target by blending geological



confidence classes. More case studies should be done to strengthen this belief. Furthermore, the case study only evaluates the absolute relative deviations, but one may be only interested in the negative relative deviations. More research is needed to evaluate whether similar relations between the GCC blend and the negative relative deviations exist.

### **Establishing the Optimal Blend of Geological Confidence Classes**

For the Yanacocha case, a maximum allowed deviation of 15% at a 90% confidence level is set for both the monthly and quarterly periods to show the methodology of establishing the optimal blend of GCC. Therefore, the 90<sup>th</sup> percentile of the realizations is used to fit a least-squares estimate. Only the data points from a ratio of 50% ore tonnage classified as measured resources and above are used for the least-squares estimate, since the trend below this ratio is less clear, which would negatively influence the least-squares fit. To include all the 90<sup>th</sup> percentile data points, a confidence interval of the least-squares estimate is created.

The developed methodology shows to be able to define an optimal blend of Geological Confidence Classes for a maximum allowed deviation, although only an optimal ratio could be established for the quarterly periods, since the maximum allowed deviation was too low for the monthly periods. However, for a higher maximum allowed deviation, one would also be able to define an optimal GCC blend for the monthly absolute relative deviation. The verification step with the schedules created using the kriging estimate resource model shows whether the relation obtained also holds for the kriging estimates. Furthermore, the required confidence level can be verified by the percentage of kriging estimated schedules that stay below the maximum allowed deviation for all periods that meet the optimal GCC blend. For the quarterly absolute relative deviations, which were calculated without a grade control step, the optimal GCC blend is 54% ore tonnage classified as measured resources, based on the intersection of the maximum allowed deviation of 15% and the confidence interval of the least-squares estimate. 95% of the schedules created with the kriging estimated resource models that meet this optimal GCC blend stay below the maximum allowed deviation, hereby indicating that the required confidence level is met.

The drawback of this method is that it requires a least-squares estimate, which is based on an arbitrary prototype function. Different prototype functions can be created and evaluated for their fit, however, the shape of the fit does not necessarily represent the true shape of the relation between the absolute relative deviations and the GCC blend, if such a relation even exists. The choice of the prototype function has a large influence on the optimal blend. Furthermore, the confidence interval of the least-squares estimate is fitted to include all data points, but does not provide ways to evaluate peaks in the deviation, which might be outliers or not. These can be included, which might result in a very conservative optimal GCC blend, or excluded, which might result in a too optimistic optimal blend. Therefore, the optimal GCC blend given by this procedure should be evaluated by visual inspection of the graph to establish a conservative and realistic optimal blend of Geological Confidence Classes. The visual inspection should be done by a Competent Person, who has detailed knowledge of the resource model.

### **Applicability**

The relation between the absolute relative deviations and the Geological Confidence Classes is shown for a test area within the depleted Yanacocha mine. This relation can be used to indicate the possibility of such a relation at different mines. To strengthen this belief, more case studies for different mines are needed. The influence of the confidence classes should be case-wise considered and are expected not to be able to be proven for the general case, since the Geological Confidence Classes are ultimately assigned by the Competent Person and not by general analytical formulations. Furthermore, the desired maximum allowed deviations is company specific based on their risk profile. Therefore, every company should evaluate their own optimal GCC blend, if a relation between the absolute relative deviations and the blend of Geological Confidence Classes can be established. This would mean that eventually, the simulations still need to be performed to adequately determine the optimal GCC blend, limiting the use of this technique for companies that are not familiar with geostatistical simulation techniques. Ordinary Kriging might also be used to indicate a minimum measured ratio for a project, by fitting a least-squares estimate to the absolute relative deviations of the kriged resource models. However, the drawback of this approach is that only an indication of the achieved confidence level can be determined after establishing an optimal blend of GCC's by determining the percentage of schedules that stay under the maximum allowed deviation.

For existing mining operations, the production data and the grade control data of the previously mined areas can be used to establish an optimal blend of GCC's for areas yet to be mined. An optimal blend of GCC's cannot be determined for new operations, since grade control data is required for the methodology established in this research. New operations should first have some periods of production in order to establish an optimal blend of GCC's, which would probably need to be updated if new grade control data is available.

The approach of blending Geological Confidence Classes should not be considered an alternative to the features of Simulation Based Optimizations, such as the creation of equally probable scenarios, which can be used to evaluate the mining and processing operations. This is not possible by blending GCC's. The method introduced here should be purely seen as a tool to control the deviations from production targets. By using the Kriging estimator, the resource model remains limited to one realization from many, equally probable models. Scheduling for a minimum measured ratio might control the deviations, but the mine planner is still kept in the dark for the range of possible outcomes, hereby unable to numerically assess the confidence and deviations. Therefore, this method should be seen as an intermediate step, to be used until the simulation methods are more developed and accepted by the mining industry.

# 6

## Conclusions and Recommendations

### Conclusions

The goal of this research is to control the short-term deviations from production target by blending Geological Confidence Classes (GCC's) from the resource reporting standards. An exhaustive data set is used to evaluate 1) the relation between the absolute relative deviations from production target and the blend of GCC's and 2) the possibility to establish an optimal blend of GCC's for a maximum allowed deviation at a certain confidence level. The Yanacocha case study shows a clear downward trend of the absolute relative deviations from 50% ore tonnage classified as measured resources and onwards. Furthermore, the case study showed that it is possible to establish a verifiable optimal blend of GCC's that limits the absolute relative deviation within a confidence level.

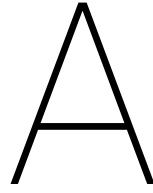
A new fast scheduler is developed that can blend GCC's. For the scheduler, a new fast method to evaluate the slope feasibility is developed, together with a new smoothing objective function which considers the surrounding blocks in three dimensions and a penalty function for not meeting the periodic GCC target blend. A newly developed 3-stage metaheuristic solver is able to create and evaluate a schedule with over 134.000 variables within eight seconds. The scheduler is limited by the cone algorithm, therefore, no complex mine designs can be created.

A methodology is developed to establish an optimal blend of GCC's. The method requires the use of simulation techniques to establish a range of equally probable scenarios in order to evaluate the required confidence level of the optimal blend. Estimation methods might also be used instead of simulation techniques, hereby losing the ability to evaluate the confidence level. A drilling campaigns algorithm is developed in order to create resource models with different blends of GCC's. The algorithm is limited in its use since the ratio of ore tonnage classified as measured or indicated resources cannot be well captured in the resource model. However, this proves not to limit the methodology to establish the optimal blend of GCC's. The resource models created with Ordinary Kriging show significant lower variances than the ones created with the E-type estimate, due to the smoothing effect of the kriging estimator. This leads to differences in the ore-waste classification and therefore, in the deviations. The absolute relative deviations per realized blend of GCC's show that the deviations can be controlled by blending GCC's. Fitting a least-squares estimate and its confidence interval to the scenarios of absolute relative deviations is able to establish an optimal blend of GCC's. However, the method is strongly dependent on the prototype function for the least-squares estimate. Therefore, the method should be seen as a tool to help the Competent Person to establish an optimal blend of GCC's.

Blending Geological Confidence Classes proves to be able to control the absolute relative deviations of the Yanacocha mine. The case study indicates the potential for other mines, however, a general proof of the relation between the GCC's and the absolute relative deviations is expected not to be possible, since the GCC's are assigned by the Competent Person and not analytically formulated. The method should be seen as a tool for the Competent Person to control the deviations and should not be used as a black box to establish an optimal blend of GCC's. For existing mining operations, the production data and the grade control data of the previously mined areas can be used to establish an optimal blend of GCC's for areas yet to be mined.

## Recommendations

- Strengthen Relation** More case studies should be done to investigate whether the relation between the Geological Confidence Classes and the absolute relative deviations also holds for other mines. Furthermore, the scope of the relation can be expanded by including multi-element mines.
- Update Ratio** The optimal blend of GCC's might be improved by reconciliation programs, which continuously evaluate the deviations from production targets. Based on a new ratio, the schedules can be updated, taking into account the previously mined areas. The process of continuous updating the ratio and developing a scheduler that can efficiently update a previous schedule for a new ratio can be investigated.
- Improve Scheduler** The scheduler can be improved by implementing a multi-ellipsoid algorithm to create more diverse and feasible schedules. The ellipsoids could be used in combination with cones to enforce the slope angle. Furthermore, research could investigate the possibility of creating the ant-routes in a block-wise manner while enforcing feasible solutions at the same time. Follow-up studies should also consider the performance of the scheduler in terms of the ability to converge to the global optimum and the movement of the Pareto front.
- Artificial Drilling Campaign** In this research, it is assumed that representative drilling campaigns can be created using grade control data. However, this is not validated, which could be done in future studies. These studies could investigate whether grade control data shows similar grades as exploration campaigns around boreholes of the exploration campaign and what the differences are in terms of the resources for a grade control created exploration campaign and a real exploration campaign for the same drillhole set-up.
- Investigate Costs** The costs of not meeting production targets are not considered in this thesis. Further research could investigate these costs, which will be especially interesting for multi-element targets with penalty elements, since these costs are expected to be more affected by the deviations. For such research, the stockpiling costs, processing efficiency and contract penalties can be considered.
- Minimize Drilling Costs** The drilling costs can possibly be lowered if the resource model can be continuously updated and the amount of mined ore from lesser drilled areas is controlled to prevent large uncertainties within a period. Less infill drilling is then needed to achieve the same accuracy level. Further research should investigate the potential of the combination of real time updating of the resource model and blending GCC's to control the deviations from production targets.
- Missclassifications** To determine the effects of an optimal blend of GCC's on the misclassification of ore and waste, the amount of missclassified blocks can be determined for schedules with different blends of GCC's. This can be combined with an evaluation of the costs of misclassifications during the short-term planning phase to determine an optimal blend of GCC's for a maximum accepted cost figure.



## Nomenclature

<b>ChromePop</b> Container for all the candidate schedules in the population for the GA [N,T,P]	<i>O</i> Number of objective functions
<b>Chrome</b> Individual schedule for the GA [N,T]	<i>A</i> Number of ants
<b>AntRoutes</b> Container for all the candidate schedules for the ACO [N,T,A]	<i>I<sub>gslib,n</sub></i> Inverse gslib index of block <i>n</i>
<b>BB</b> Ultimate pit boundary to indicate whether block <i>n</i> can be mined (1) or not (0) [N]	<i>X<sub>n,t</sub></i> Binary decision variable to indicate whether block <i>n</i> is mined in period <i>t</i> (1) or not (0)
<b>OB</b> The objective values of the candidate schedules [P,O]	<i>O<sub>x</sub></i> The x-coordinate of the origin of the blockmodel
<b>LB</b> Matrix containing all the block indexes that have to be mined in order to mine block <i>n</i> [N,N]	<i>O<sub>y</sub></i> The y-coordinate of the origin of the blockmodel
<b>NLB</b> Matrix containing the number of blocks that have to be mined in order to mine block <i>n</i> - when INF, the block cannot be mined [N]	<i>E<sub>z</sub></i> The z-coordinate of the outer extend of the blockmodel
<b>BD</b> Block matrix containing the densities [N]	<i>S<sub>x</sub></i> The block size in the x direction ( <i>m</i> )
<b>BC</b> Block matrix containing the confidence levels [N]	<i>S<sub>y</sub></i> The block size in the y direction ( <i>m</i> )
<b>BG</b> Block matrix containing the grades [N]	<i>S<sub>z</sub></i> The block size in the z direction ( <i>m</i> )
<b>BOW</b> Block matrix indicating whether block <i>n</i> is ore (1) or waste (0) [N]	<i>N<sub>x</sub></i> Number of blocks in the x-direction
<b>BAR</b> Block Matrix indicating whether block <i>n</i> is air (0) or rock (1) [N]	<i>N<sub>y</sub></i> Number of blocks in the y-direction
<b>BX</b> Block matrix containing the x-coordinates [N]	<i>D<sub>cone</sub><sup>t</sup></i> Depth of the cone in period <i>t</i> ( <i>m</i> )
<b>BY</b> Block matrix containing the y-coordinates [N]	<i>β</i> Angle of the cone
<b>BZ</b> Block matrix containing the z-coordinates [N]	<i>C<sub>mine,t</sub></i> Maximum allowed mining capacity in period <i>t</i>
<b>Ph</b> Pheromones of the objectives [N,T,O]	<i>C<sub>proc,t</sub></i> Maximum allowed processing capacity in period <i>t</i>
<i>N</i> Number of blocks	<i>F<sub>reduc</sub></i> Reduction factor for the maximum capacity
<i>T</i> Number of periods	<i>g<sub>cutoff</sub></i> Cutoff grade ( <i>oz/t</i> )
<i>P</i> Population for the GA	<i>V<sub>B</sub></i> Block volume ( <i>m</i> <sup>3</sup> )
	<i>R</i> Recovery factor
	<i>Y<sub>smoothing</sub><sup>k,t</sup></i> The smoothing penalty for block <i>k</i> in period <i>t</i>
	<i>A</i> The penalty weights of the blocks in region <i>nb1</i>

$B$	The penalty weights of the blocks in region $nb2$	$TO_{max}$	Maximum tour size
$nb1$	The blocks directly adjacent to the mined block	$TO_{start}$	Moment when the tour size is increased
$nb2$	The blocks in the outer region of the smoothing cube	$TO_{quit}$	Moment when the tour size is at its maximum
$N_{nb1}$	The number of blocks in region $nb1$	$DA_{GAL}$	The damp factor at loop GAL
$N_{nb2}$	The number of blocks in region $nb2$	$DA_{start}$	Moment when the damp factor is decreased
$d_{geo}$	The geological discount factor	$DA_{quit}$	Moment when the damp factor is 0
$Y_{confidence}$	Penalty for not meeting the geological confidence ratio	$M_{B,max}$	The maximum number of brute force mutations per period
$Y_{mine}$	Penalty for not meeting the mining target	$M_{chance}$	The chance of a child being mutated
$Y_{proc}$	Penalty for not meeting the processing target	$CR_{move}$	The maximum allowed number of block moves for the heuristic crossover
$R_{measured}$	Required ratio for the measured blocks	$CMX_t$	x-coordinate of the Centre of Mass in period $t$
$R_{indicated}$	Required ratio for the indicated blocks	$CMY_t$	y-coordinate of the Centre of Mass in period $t$
$R_{inferred}$	Required ratio for the inferred blocks	$CMZ_t$	z-coordinate of the Centre of Mass in period $t$
$D_{euc}^n$	Euclidean distance of block $n$ to the nearest borehole	$CP$	Compaction measurement
$D_{euc}^{mea}$	Maximum euclidean distance for the classification of a measured block	$NQ_1$	Number of blocks in quadrant 1, inversely scaled with the period the block is mined in
$D_{euc}^{ind}$	Maximum euclidean distance for the classification of an indicated block	$NQ_2$	Number of blocks in quadrant 2, inversely scaled with the period the block is mined in
$PO$	Power factor for the confidence ratio penalties	$NQ_3$	Number of blocks in quadrant 3, inversely scaled with the period the block is mined in
$GAL$	Number of proceeded loops for the optimization	$NQ_4$	Number of blocks in quadrant 4, inversely scaled with the period the block is mined in
$TO_{min}$	Minimum tour size		

# Bibliography

- Alaya, I. (2007). Ant colony optimization for multi-objective optimization problems. *19th IEEE International Conference on Tools with Artificial Intelligence*, 1st:450–457.
- Albuquerque, M., Antunes, I., Seco, M., Roque, N., and Sanz, G. (2014). Sequential gaussian simulation of uranium spatial distribution – transboundary watershed case study. *Procedia Earth and Planetary Science*, 8:2–6.
- Askari-Nasab, H., Badiozamani, M., and H., E. (2010). Hierarchical clustering algorithms for block aggregation in open pit mines. *Proceedings Mine Planning and Equipment Selection*, pages 469–480.
- Atlas-Copco (2012). *Blasthole Drilling in Open Pit Mining*. Taylor and Francis., 3rd edition.
- AusIMM (2012). Australian code for reporting of mineral resources and ore reserves. *Mineral Resource and Ore Reserve Estimation-The AusIMM Guide to Good Practice*.
- Benndorf, J. and Dimitrakopoulos, R. (2009). Stochastic long-term production scheduling of iron ore deposits – integrating joint multi-elemental geological uncertainty. *Orebody Modelling and Strategic Mine Planning*, pages 155–163.
- Benndorf, J. and Dimitrakopoulos, R. (2013). Stochastic long-term production scheduling of iron ore deposits: Integrating joint multi-elemental geological uncertainty. *Journal of Mining Science*, 49:68–81.
- Bohling, G. (2005). Introduction to geostatistics and variogram analysis. *C and PE*, 940.
- Cowan, E. (2012). Overview – geological interpretation and geological modelling. *Mineral Resource and Ore Reserve Estimation-The AusIMM Guide to Good Practice*, pages 137–144.
- Creech, M. (2012). Reserves, reserves and not a tonne to mine – a study of reserves reported prior to mine closure. *Mineral Resource and Ore Reserve Estimation-The AusIMM Guide to Good Practice*, pages 627–634.
- CRIRSCO (2013). *International Reporting Template*.
- Dagdelen, K. (2004). Open pit optimisation – strategies for improving economics of mining projects through mine planning. *Orebody Modelling and Strategic Mine Planning*, pages 21–24.
- Dejonghe, L. and Boni, M. (2006). Combining genetic algorithms and topological sort to optimize open-pit mine plans. *Proceedings 15th International Symposium of Mine Planning Equipment Selection*, 15:1234–1239.
- Dimitrakopoulos, R. and Godoy, M. (2012). Grade control based on economic ore/waste classification functions and stochastic simulations – examples, comparisons and applications. *Mineral Resource and Ore Reserve Estimation-The AusIMM Guide to Good Practice*, pages 685–699.
- Dowd, P., Xu, C., and Coward, S. (2014). Strategies for mine planning and design. *Orebody Modelling and Strategic Mine Planning*, pages 41–50.
- Fleurissen, J.-A. (2012). Slope design and implementation in open pit mines: Geological and geomechanical approach. *Procedia Engineering*, 46:27–38.
- Froyland, G., Meabde, M., Stone, P., and Hodson, D. (2004). The value of additional drilling to open pit mining projects. *Orebody Modelling and Strategic Mine Planning*, pages 169–176.
- Gendreau, M. and Potvin, J.-Y. (2010). *Handbook of Metaheuristics*. Springer, 2nd edition.
- Glacken, I. and Trueman, A. (2012). Overview – mineral resource estimation. *Mineral Resource and Ore Reserve Estimation-The AusIMM Guide to Good Practice*, pages 263–276.

- Godoy, M. (2003). *The effective management of geological risk*. PhD thesis, University of Queensland.
- Godoy, M. and Dimitrakopoulos, R. (2004). A multi-stage approach to profitable risk management for strategic planning in open pit mines. *Orebody Modelling and Strategic Mine Planning*, pages 339–346.
- Goodfellow, R. and Dimitrakopoulos, R. (2014). Stochastic optimisation of mineral value chains – developments and applications for the global optimisation of mining complexes with uncertainty. *Orebody Modelling and Strategic Mine Planning*, pages 13–21.
- Hall, G. (2009). The need for validation of optimisation solutions in mine design. *Orebody Modelling and Strategic Mine Planning*, pages 119–123.
- Hennessy, H. (2005). Gold Mine Sparks Battle in Peru. [Online; accessed 23-07-2016].
- Jara, R., Couble, A., Emery, X., Magri, E., and Ortiz, J. (2006). Block size selection and its impact on open pit design and mine planning. *South African Institute of Mining and Metallurgy*, 106:205–212.
- Jewbali, A. and Dimitrakopoulos, R. (2009). Stochastic mine planning – example and value from integrating long- and short-term mine planning through simulated grade control. *Orebody Modelling and Strategic Mine Planning*, pages 327–333.
- Kalinli, A. and Sarikoc, F. (2007). A parallel ant colony optimization algorithm based on crossover operation. *Advances in Metaheuristics for Hard Optimization*, 1st:87–110.
- King, B. (2014). Modelling open pit and underground production scheduling in concert. *Orebody Modelling and Strategic Mine Planning*, pages 369–373.
- Knowles, J. and Corne, D. (2003). Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation*, 7:100–116.
- Lamghari, A. and Dimitrakopoulos, R. (2012). A diversified tabu search approach for the open-pit mine production scheduling problem with metal uncertainty. *Journal of Operational Research*, 22:642–662.
- Leite, A. and Dimitrakopoulos, R. (2009). Production scheduling under metal uncertainty – application of stochastic mathematical programming at an open pit copper mine and comparison to conventional scheduling. *Orebody Modelling and Strategic Mine Planning*, pages 27–31.
- Leuangthong, O., McLennan, J., and Deutsch, C. (2004). Minimum acceptance criteria for geostatistical realizations. *Natural Resources Research*, 13:131–141.
- Li, J. and Heap, A. (2008). A review of spatial interpolation methods for environmental scientists. *Geoscience Australia*, 32:137.
- Luke, S. (2015). *Essentials of Metaheuristics*. Lulu, 2nd edition.
- Manchuk, J. and Deutsch, C. (2012). Implementation aspects of sequential gaussian simulation on irregular points. *Computational Geoscience*, 16:625–637.
- Menabde, M., Froyland, G., Stone, P., and Yeates, G. (2004). Mining schedule optimisation for conditionally simulated orebodies. *Orebody Modelling and Strategic Mine Planning*, pages 347–351.
- Newman, A., Rubio, E., R., C., Weintraub, A., and Eurek, K. (2010). A review of operations research in mine planning. *Interfaces*, 40:222–245.
- Noppe, M. (2012). Reporting and converting resources to reserves – how confident are we. *Mineral Resource and Ore Reserve Estimation-The AusIMM Guide to Good Practice*, pages 805–814.
- Noppe, M. (2014). Communicating confidence in mineral resources and mineral reserves. *The Journal of the Southern African Institute of Mining and Metallurgy*, 114:213–222.
- Nowak, M. and Verly, G. (2004). The practice of sequential gaussian simulations. *Orebody Modelling and Strategic Mine Planning*.



- Pearl, J. (1983). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1st edition.
- Ramazan, S. and Dimitrakopoulos, R. (2004). Stochastics optimisation for long-term production scheduling for open pit mines with a new integer programming formulation. *Orebody Modelling and Strategic Mine Planning*, pages 353–359.
- Reid, R. (2015). The importance of boundary analysis and how to do it in micromine. Technical report, Harmony Gold SE Asia Pty Ltd.
- Remy, N., Boucher, A., and Wu, J. (2011). *Applied Geostatistics with SGeMS*. Cambridge University Press, 1st edition.
- Render, J.-M. (2009). Cut-off grade estimation –old principles revisited –application to optimisation of net present value and internal rate of return. *Orebody Modelling and Strategic Mine Planning*, pages 165–169.
- Ribeiro, M. (2013). *Multi-Objective Pareto-Efficient Algorithms for Recommender Systems*. PhD thesis, Universidade Federal de Minas Gerais.
- Safikhani, M., Asghari, O., and Emery, X. (2016). Assessing the accuracy of sequential gaussian simulation through statistical testing. *Stochastic Environmental Research and Risk Assessment*, 1.
- Soltani, F., Afzal, P., and Asghari, O. (2004). Delineation of alteration zones based on sequential gaussian simulation and concentration–volume fractal modeling in the hypogene zone of sungun copper deposit, NW Iran. *Journal of Geochemical Exploration*, 140:64–76.
- Spleit, M. (2014). Production scheduling optimisation meeting targets by hedging against geological risk while addressing environmental and equipment concerns. *Orebody Modelling and Strategic Mine Planning*, pages 397–404.
- Stephenson, P., Allman, A., Carville, D., Stoker, P., Mokos, P., Tyrrell, J., and Burrows, T. (2012). Estimation and reporting potential mineralisation at bhp biliton – the unconstrained view. *Mineral Resource and Ore Reserve Estimation-The AusIMM Guide to Good Practice*, pages 799–804.
- Stone, P., Froyland, G., Meabde, M., Law, B., Pasyar, R., and Monkhouse, P. (2004). Blasor – blended iron ore mine planning optimisation at yandi. *Orebody Modelling and Strategic Mine Planning*, pages 285–288.
- Tatman, C. (2001). Production rate selection for steeply dipping tabular deposits. *Mining Engineering*, 53:62–64.
- Tavchandijan, O., Proulx, A., and M., A. (2004). Application of conditional simulations to capital decisions for ni-sulfide and ni-laterite deposits. *Orebody Modelling and Strategic Mine Planning*, pages 289–299.
- Teunissen, P., Simons, D., and Tiberius, C. (2009). Probability and observation theory. Technical report, TU Delft. lecture notes.
- Vallee, M. (2000). Mineral resource + engineering, economic and legal feasibility = ore reserve. *Canadian Institute of Mining, Metallurgy and Petroleum*, 90:53–61.
- Vann, J., Jackson, S., Bye, A., Coward, S., Moayer, S., Nicholas, G., and Wolff, R. (2012). Scenario thinking – a powerful tool for strategic planning and evaluation of mining projects and operations. *Mineral Resource and Ore Reserve Estimation-The AusIMM Guide to Good Practice*, pages 593–603.
- Weise, T. (2009). *Global Optimization Algorithms - Theory and Application* -. self-published, 2nd edition.
- Yamamoto, J. K. (2005). Correcting the smoothing effect of ordinary kriging estimates. *Mathematical Geology*, 37(1):69–94.