

Mining sequencing to control blend quality

Winston L. Sweatman¹ Kevin White² Amie Albrecht³
Martin Peron⁴ Peter Pudney⁵ David Whittle⁶

(Received 1 October 2017; revised 28 February 2018)

Abstract

The extraction of ore from a mine must be scheduled to meet specific order production targets or ‘builds’. A number of physical, logical, and capacity constraints affect this planning. There is uncertainty in the process due to the imprecision of our knowledge of the mine’s content until extracted from the ground. At the 2016 mathematics-in-industry study group workshop in Australia, Schneider Electric presented a project to consider mine scheduling. This paper reports on the assorted modelling approaches: exploration of sample data; considerations of the physically feasible mining sequences; the construction of a mixed integer program; a general heuristic strategy for dealing with different levels of uncertainty; and a build simulation. These provide promising avenues for further research on mine sequencing and related problems.

[DOI:10.21914/anziamj.v58i0.12475](https://doi.org/10.21914/anziamj.v58i0.12475), © Austral. Mathematical Soc. 2018. Published 2018-03-23, as part of the Proceedings of the 2016 Mathematics and Statistics in Industry Study Group. ISSN 1445-8810. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to the DOI for this article.

Contents

1	Introduction	M34
2	Block model and sample data	M35
2.1	Mine slices	M38
3	Finding valid paths	M38
4	Deterministic mixed integer programming model	M41
4.1	Constants and variables	M42
4.2	Objective	M44
4.3	Constraints	M45
4.4	Potential enhancements	M47
5	Evaluation of a heuristic to select blocks in an iron ore mine	M51
5.1	Criterion for build feasibility.	M55
5.2	How likely is it that the blocks excavated so far belong to the feasibility cone?	M56
5.3	Discussion	M59
6	Simulating the build construction process	M61
6.1	Future work	M63
7	Conclusion	M65
	References	M66

1 Introduction

A number of factors play a role in the management of an open pit mine. Mine scheduling is the task of determining the best way to utilise the resource

considering these factors. The whole process is complicated by uncertainty in the exact contents of the mine until material has been extracted from the ground. There exist commercial programs to manage mine operations and such operations have been the subject of previous investigations [1, 2, 3].

This 2016 Mathematics-in-Industry Study Group (MISG-2016) project explored two aspects of management: initial planning before removing material from the ground; and handling material after excavation. In particular the project was concerned with the uncertainty that arises due to imprecise knowledge of the ore's grade until it has been extracted.

The contents of a mine are considered to be an ensemble of rock cuboids in the 'block model' (Section 2). Sample data of this form is displayed in Subsection 2.1. During production, the extraction of a block of material by diggers (also known as excavators) is limited by the requirement to extract certain blocks above it. This constrains the number of feasible extraction sequences as considered in Section 3.

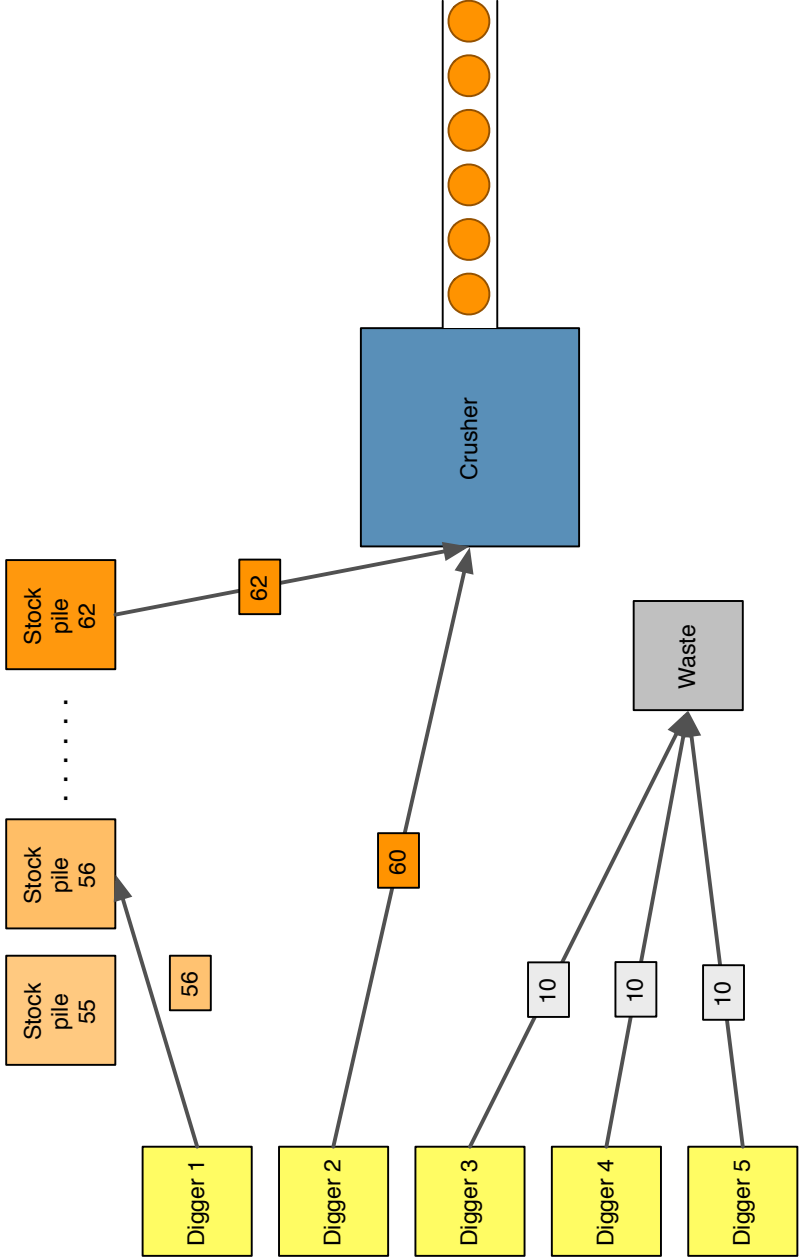
Production targets consist of 'builds', which are individual orders for ore of specified quantity (tonnage) and quality (grade). Once material is above ground it is either immediately added to a build, or added to a waste pile, or temporarily stored in a stockpile for later use. Figure 1 shows a schematic of the mining process.

For the initial determination of the excavation sequence, a mixed integer program is formulated (Section 4). Section 5 considers a more general strategy with sequencing to enable adaptation to meet targets. Section 6 describes a simulation for the build construction process.

2 Block model and sample data

The sample data supplied by the industry representative details a mine partitioned into many thousands of blocks of material. This block model is a

Figure 1: Schematic mining process: diggers remove rock from the ground; this goes to waste, a stockpile or a crusher. Stockpile material is also eventually sent to the crusher. Crushed material makes builds to meet demand. (Numbers on arcs indicate the grade of the ore.)



fundamental feature; it permeates all subsequent analysis. In the sample, the blocks are cuboids, all of the same dimensions: 25 metres wide by 25 metres long, and 5 metres deep. These are typical measurements. A record for each block in the sample includes its position in three-dimensional space and a description of its contents.

In a block model for a mineral resource, each block record contains information such as the rock type, tonnage, product quantities or percentages and other variables that are used to determine mining or processing costs, yields and revenues. For the sample block model there is information for each block's tonnage and grade. The grade is defined by the percentages of the block's mass composed of oxides of iron (Fe) and of various deleterious elements (e.g., silicates (Si)) whose presence degrades the value of the ore. The block record includes predictions of these values as they occur in the ground, as well as predictions of the values after a defined metallurgical process. The in-ground estimates are made by geologists and geostatisticians.

During surveying, special purpose drill rigs are used to collect samples from the area of interest. To limit costs, drilling is carried out on as wide a spacing as possible (say, on a 100 m grid). Values between the samples must be interpolated and miners use a range of geostatistical methods to achieve this interpolation. The further a block is from a drill hole sample, the more uncertain is its predicted grade. Geostatistical methods provide information about this uncertainty. However, this information is rarely passed on to a mine planner.

Blocks are mined by the open pit method, which means that if a block is to be mined, then a set of blocks above it must be mined to uncover it. In our case, we make the assumption that for block i to be mined, the block immediately above block i must also be mined, as well as the eight immediate horizontally adjacent neighbours of the block above block i .

For the excavation itself, diggers must be assigned to mine blocks. The destination of material from those blocks needs to be determined: it could be one of several stockpiles, or waste dumps, or the processing plant. Material

is also reclaimed from the stockpiles and sent to the processing plant, in order that the combined material sent to the processing plant meets the grade and tonnage requirements of a build. The planning method must be scalable to work with tens of thousands of distinct blocks of material in the geological model.

2.1 Mine slices

Diagrams of horizontal slices through the mine, shown in [Figure 2](#), help us to visualise the data. Each image represents two slices—the dots are used to indicate one layer, and the squares indicate the layer above it. The colours indicate the quality of the ore: dark blue is poor quality and shades go through to light red for high quality. Grey is waste. The sample data z-position values for the blocks are at vertical intervals of 5 m and these are labelled beginning from the lowest point of the mine. The sequence of slices are shown from the highest level of the mine (75 m) to the lowest level (5 m).

3 Finding valid paths

Blocks must be uncovered before they can be mined. [Figure 3](#) shows precedence constraints for an example two-dimensional mine, where the depth of the mine increases down the page. The block at the top of each arrow must be removed before the block at the bottom can be mined. For example, blocks 4, 5 and 6 must be removed before block 10 can be mined.

We use topological sorting of such precedence graphs to determine feasible sequences. Suppose we wish to mine from left to right. If we order the blocks from left to right and subsequently do a stable topological sort using the precedence constraints, then we get the mining sequence shown in [Figure 4](#). On the other hand, if we order the blocks from right to left and subsequently

Figure 2: Horizontal mine slices, $z \in \{75, 70, \dots, 5\}$ m. These are shown in the order top left, top centre, top right, \dots , bottom left, bottom centre, bottom right beginning from the highest point (75 m) and going to the lowest point (5 m). The colour shade indicates the ore quality: dark blue is poor quality, light red is high quality, and grey is waste.

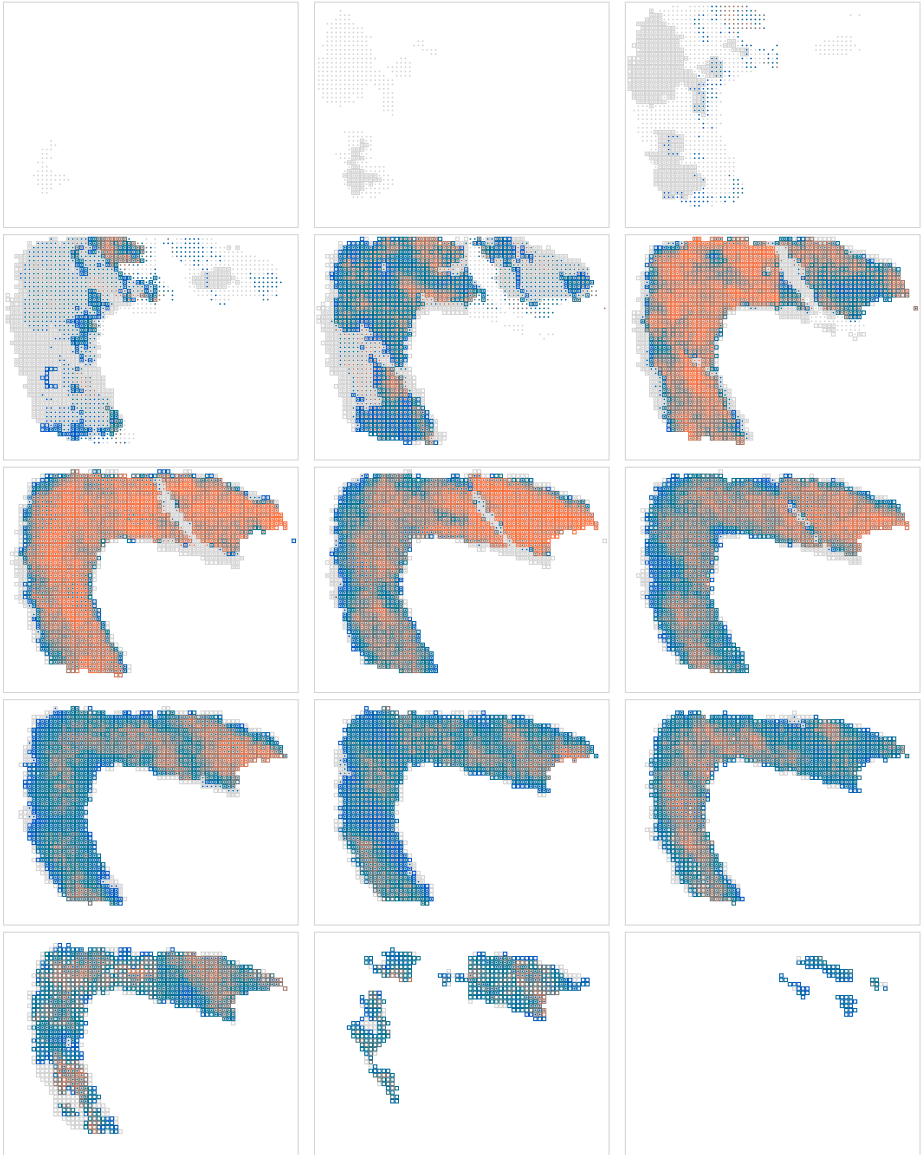


Figure 3: Block precedence constraints.

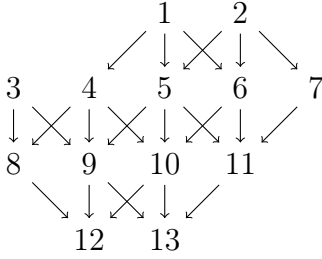


Figure 4: Left to right mining sequence.

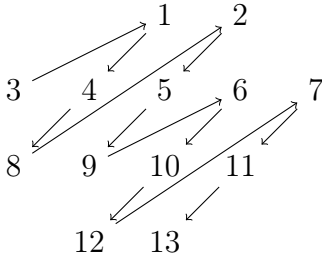
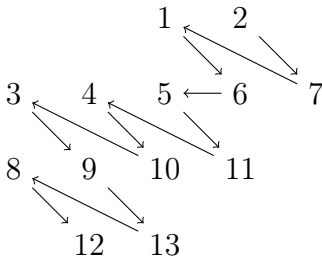


Figure 5: Right to left mining sequence.



do a stable topological sort, then we get the mining sequence shown in Figure 5.

4 Deterministic mixed integer programming model

We present a mixed integer programming (MIP) model for a short-term mine scheduling problem. This MIP model is nonlinear under its current formulation, but we suggest a number of ways to linearise it. Our MIP model works with the user-provided block model.

Mining involves drilling, blasting, excavating and loading the rock onto a truck. The truck is then sent to some predetermined location to dump its load. In our modelling, we focus on the excavators, which typically constitute the production bottleneck in a mine. That is, we assume there are ample resources for drilling, blasting and trucking, and concentrate on the efficient utilisation of the excavators (nominally five in this model).

Our optimisation model seeks to minimise the time required to make a build, or customer order. The time is a function of the capacity of the five excavators, the tonnages of the blocks assigned to be mined by the excavators, and an additional time penalty for blocks depending on their distance from the excavator. This penalty can be increased or decreased to change the propensity of excavators to favour mining blocks near their nominated locations.

We present the features of the MIP in the following order: the constants (or coefficients) that will be fed to the MIP solver and the variables representing decisions; the objective function motivating our decisions; and the constraints linking these components together.

4.1 Constants and variables

Blocks We supplement block records of the recoverable (after processing) contents of a block with a calculation of the block's distance from the initial location of each of the excavators. The tonnage and grade are used to determine a block's contribution to a stockpile or directly to a build. In this example formulation we model only two grades: for block i , F_i (Fe) is the product grade percentage and R_i (Si) is a deleterious element percentage. The formulation can be expanded to include additional grades as required, simply by specifying additional values and constraints.

The distance D_{ij} is used to calculate the time penalty for the excavator j to mine block i . The greater the distance, the greater the time penalty. The calculation of distance could be by any metric, for example Euclidean, rectilinear or following defined pathways. However, we do not model travel directly; we use D_{ij} to apply a time penalty that is factored with another user-defined variable A_j (next paragraph). Accordingly, there appears to be little value in taking a very precise approach to modelling distance.

M_i	Mass (tonnes) for block i
F_i	Fe grade (%) for block i
R_i	Si grade (%) for block i
D_{ij}	Distance (metres) of block i from the initial position of excavator j

Excavators Excavators are each given an initial position, typically where the excavator was at the end of the previous build. Alternatively, the position may be chosen by an engineer, should they have reason to target a particular area. The distance time penalty A_j is used to increase or decrease the propensity of an excavator to move long distances to mine blocks with desirable grade characteristics. If A_j is set to a higher value, then the excavator is less inclined to move large distances. The units of A_j are hours/metre. However, the penalty A_j need not be directly related to the actual speed of the excavator;

it can be a more general penalty factor.

- P_j Production maximum (tonnes/hour) for excavator j
- A_j Distance time penalty (hours/metre) to apply to D_{ij}

Destinations Destinations are stockpiles, waste dump(s) and the processing plant. In the case of each stockpile, nine parameters are used by our MIP model to specify the allowable and initial sizes, and the grade entry criteria. In the case of waste dump locations, no constraints are envisaged and no values are required as input. In the case of the processing plant, only a tonnage/hour capacity limit is required.

Sources and Destinations:

- Stockpiles indexed by $k \in \{1, 2, 3, 4, 5\}$,
- Waste dump $k = 6$,
- Processing plant $k = 7$.

m_k^{\max} , m_k^{\min} , m_k^{init}	Tonnage maximum, minimum and initial value for stockpile k
f_k^L , f_k^U , f_k^{init}	Grade minimum, maximum and initial value for Fe in stockpile k
r_k^L , r_k^U , r_k^{init}	Grade minimum, maximum and initial value for Si in stockpile k
C	Production maximum (tonnes/hour) for processing plant

Build A build is a customer order, specified by its tonnage and grade. There is just one build considered in each optimisation.

T^{\min}	Minimum tonnage for build
T^{\max}	Maximum tonnage for build
F^{\min}	Minimum Fe grade for build
R^{\max}	Maximum Si grade for build

Variables Two sets of decision variables determine the material to be mined and the material which is to be reclaimed from stockpiles in order to complete a build.

$x_{ijk} \in [0, 1]$	Proportion (by weight) of block i excavated by excavator j and sent to destination k
$x'_i \in \{0, 1\}$	$x'_i = 1$ indicates that block i is excavated entirely; $x'_i = 0$ if the block is excavated partially or not at all.
$y_k \geq 0$	Tonnes of ore from stockpile $k \in \{1, 2, 3, 4, 5\}$ sent to the processing plant.
f_k	The Fe grade of stockpile k prior to reclamation.
r_k	The Si grade of stockpile k prior to reclamation.
m_k	The tonnage of stockpile k prior to reclamation.
t	The time necessary to complete the build.

Binary variables x'_i ensure that the precedence constraints are satisfied. Variables f_k , r_k , m_k are functions of the x_{ijk} and in turn constrain y_k . We assume that each stockpile is operated in two stages; all material sent to a stockpile arrives before any material is reclaimed to send to the processing plant.

[Subsection 4.3](#) provides further details.

4.2 Objective

The objective is to minimise the time to complete a build. This is a simple objective, and quite appropriate for an operational time frame:

minimise t .

Many will ask: what about costs? In very short time frames on a mine site, the vast majority of costs are fixed rather than variable (that is, they vary with time, not with production). So in the short term, the simplest way to contribute to profitability is by producing the maximum amount with the available production resources. Each tonne of additional production in a time period adds revenue, at very little additional cost.

4.3 Constraints

Excavator An excavator has a defined tonnage extraction rate P_j and each block has a defined tonnage M_i . Each block also has a distance from each excavator D_{ij} and each excavator has a time penalty parameter A_j . These parameters are used to determine how many hours of excavator time are consumed as a function of the blocks mined by the excavator. Since the variables x_{ijk} are continuous, the time penalty is proportional to the amount excavated. It is possible to enforce that the excavator is fully penalised, at the cost of additional binary variables (Subsection 4.4). The time t that is to be minimised in the objective function must be greater than or equal to the time consumed for each excavator:

$$\sum_k \sum_i x_{ijk} \left(\frac{M_i}{P_j} + D_{ij}A_j \right) \leq t, \quad \text{for all } j.$$

Block precedence Operationally, predecessors of block i must be mined before i is mined. In this model we seek to determine which blocks are to be mined in order to complete a particular build. Thus if block i is to be included as part of this build, then its predecessors must be mined also. If a block is to be mined even partially, then any other blocks that are identified as predecessor blocks should be excavated entirely, so the binary variable x'_h should equal one for each predecessor h :

$$\sum_k \sum_j x_{ijk} \leq x'_h, \quad \text{for all } i, \quad \text{for all } h \in \{\text{predecessors of } i\}.$$

If the binary variable x'_h equals one, then the block should be excavated entirely:

$$\sum_k \sum_j x_{hjk} \geq x'_h, \quad \text{for all } h.$$

Grades in stockpile Each stockpile only accepts ore of appropriate grade. This constraint is expressed in terms of lower and upper bounds on the iron and silica content of blocks sent to the stockpile:

$$\begin{aligned} x_{ijk} &= 0, & \text{for all } i, j, k \text{ such that } F_i \notin [f_k^L, f_k^U], & \quad k \in \{1, 2, 3, 4, 5\}; \\ x_{ijk} &= 0, & \text{for all } i, j, k \text{ such that } R_i \notin [r_k^L, r_k^U], & \quad k \in \{1, 2, 3, 4, 5\}. \end{aligned}$$

Quantity and grade of each stockpile In this model it is assumed that all material sent to each stockpile k through the actions relating to variables x_{ijk} arrives on the stockpile before any tonnes are reclaimed through the action relating to variable y_k . In that context we define m_k as the tonnage of stockpile k before reclamation. Its value is given by the initial tonnage on the stockpile (m_k^{init}), plus the tonnage added by excavators:

$$m_k^{\text{init}} + \sum_i \sum_j x_{ijk} M_i = m_k, \quad k \in \{1, 2, 3, 4, 5\}.$$

Similarly, we define the grades for stockpiles prior to reclamation (f_k and r_k) as functions of each stockpile's initial grades (f_k^{init} and r_k^{init}), and the tonnes (with grades) added by excavators:

$$\begin{aligned} m_k^{\text{init}} f_k^{\text{init}} + \sum_i \sum_j x_{ijk} M_i F_i &= m_k f_k, & k \in \{1, 2, 3, 4, 5\}; \\ m_k^{\text{init}} r_k^{\text{init}} + \sum_i \sum_j x_{ijk} M_i R_i &= m_k r_k, & k \in \{1, 2, 3, 4, 5\}. \end{aligned}$$

Reclaiming from stockpiles The amount that can be reclaimed from a stockpile is limited to the tonnes m_k on a stockpile k before reclamation, less the minimum tonnage allowed for the stockpile m_k^{min} :

$$y_k \leq m_k - m_k^{\text{min}}, \quad k \in \{1, 2, 3, 4, 5\}.$$

Processing plant capacity Although the focus is on the excavators as the bottleneck for mining production, we also allow a processing constraint to be defined. The time t for the build must be greater than or equal to the tonnes of material required for the build, divided by the production capacity of the processing plant:

$$t \geq \frac{T^{\min}}{C}.$$

Build constraints on quantity and grade The tonnage of ore reclaimed from the stockpiles combined with the tonnage of the blocks mined and sent to the processing plant must be within the given bounds for the quantity required. Recalling that the processing plant is denoted as destination $k = 7$, then

$$T^{\min} \leq \sum_{k \in \{1,2,3,4,5\}} y_k + \sum_i \sum_j x_{ij7} M_i \leq T^{\max}.$$

The weighted average grades of ore reclaimed from stockpiles and ore mined and sent directly to the processing plant will meet the build grade constraints F^{\min} (minimum Fe), and R^{\max} (maximum Si):

$$\begin{aligned} \sum_{k \in \{1,2,3,4,5\}} y_k f_k + \sum_i \sum_j x_{ij7} F_i M_i &\geq F^{\min} T^{\min}, \\ \sum_{k \in \{1,2,3,4,5\}} y_k r_k + \sum_i \sum_j x_{ij7} R_i M_i &\leq R^{\max} T^{\max}. \end{aligned}$$

4.4 Potential enhancements

Bounding the input The sample data is a block model containing tens of thousands of blocks, but each build will require only a few dozen blocks to be mined. It may be possible to reduce the size of the input block model, without compromising the optimality of the solution. For example, some blocks cannot become part of a build because they are buried too deeply beneath other

blocks that would be predecessors. Similarly, there may be many blocks that are just too far away from the initial positions of the excavators to be realistically considered. Identifying and removing these blocks from the input model would improve the tractability of the problem.

Additional user-defined constraints on domain or heuristics involving pit optimisation and pit parameterisation could be used to reduce the input size, although such approaches carry with them the risk of high-value solutions being missed.

Removal of nonlinearity Nonlinearities arise in the stockpile quantity and grade constraints due to the quantity variables m_k and y_k being multiplied by grade variable proportions f_k and r_k . There are ways to address this issue.

- Variables f_k and r_k could be replaced by assumed values F_k and R_k . These assumed values would be set at (say) the mid-point of the input grade range for each stockpile. After the MIP is solved, the actual f_k and r_k values (weighted averages of the grades arriving in stockpile k) are calculated and fed to the next solving stage as the new assumed values.
- It is possible to approximate virtually any non-linear constraint by adding new binary variables. For each stockpile $k \in \{1, 2, 3, 4, 5\}$, introduce a discretised finite set V_k of possible values for m_k and one binary variable b_{vk} for each element v of V_k . The value of m_k is approximated by one value v for which $b_{vk} = 1$. The binary variables comprise a Special Ordered Set of type 1. Exactly one of the variables b_{vk} will be non-zero in any solution, which is enforced by the constraint $\sum_{v \in V_k} b_{vk} = 1$. Then the grade constraint

$$m_k^{\text{init}} f_k^{\text{init}} + \sum_i \sum_j x_{ijk} M_i F_i = m_k f_k, \quad (1)$$

is replaced by a pair of ‘big M’ constraints:

$$m_k^{\text{init}} f_k^{\text{init}} + \sum_i \sum_j x_{ijk} M_i F_i \geq v f_k - M(1 - b_{vk}), \quad \text{for all } v \in V_k, \quad (2)$$

$$m_k^{\text{init}} f_k^{\text{init}} + \sum_i \sum_j x_{ijk} M_i F_i \leq v f_k + M(1 - b_{vk}), \quad \text{for all } v \in V_k. \quad (3)$$

These latter constraints are always satisfied for $b_{vk} = 0$, provided that a large enough value for the positive parameter M is chosen. For $b_{vk} = 1$, the pair of constraints is equivalent to the constraint (1) with $m_k = v$.

It remains to find a good set V_k , which should be small enough to be tractable but large enough to be a good approximation of m_k . Each $v \in V_k$ generates a binary variable and a pair of inequality constraints. A similar process needs to be applied to each constraint having a product of variables, including those that involve the y_k variables. It may be impossible to achieve exactly one of the values in V_k , and a small tolerance may need to be added to the right sides of the constraints (2) and (3).

Converting the mixed integer program to a linear program Subject to removal of non-linear constraints, this problem could be relaxed to a linear program (LP) by making each binary variable $x'_i \in \{0, 1\}$ continuous (that is, $x'_i \in [0, 1]$). Although this would yield a more tractable problem with only continuous variables, it would also prevent the precedence constraints from being effective. In the new formulation, excavating say 50% of a block would require excavating 50% of each of the predecessor blocks. However, relaxations are useful for obtaining bounds on the values of optimal MIP solutions, and heuristics are used to subsequently obtain good solutions to the MIP.

Time penalty Each excavator has a time penalty parameter A_j to prompt excavators to mine blocks near their nominated locations. Since the vari-

ables x_{ijk} are continuous, the time penalty $x_{ijk}D_{ij}A_j$ is proportional to how much of the block is excavated. A more realistic model would penalise the excavation of even a small amount of a remote block. However, this would require adding new binary variables for each (excavator, block) pair. Again, it would be up to the practitioner to trade off precision against tractability.

Another issue with the current formulation is that an excavator would be penalised several times when excavating a cluster of remote blocks. Ideally, the time penalty would be directly related to the distance travelled by the excavator. This comes at the cost of one extra binary variable $t_{ii'j}$ per pair of blocks (i, i') and excavator j , with $t_{ii'j} = 1$ meaning that the excavator j travels to block i' immediately after block i . The minimum time constraints would be replaced by

$$\sum_k \sum_i x_{ijk} \frac{M_i}{P_j} + \sum_i \sum_{i'} (D_{ii'}A_j)t_{ii'j} \leq t, \quad \text{for all } j,$$

where $D_{ii'}$ is a measure of the distance between blocks i and i' . Additional constraints would be needed to ensure that the variables $t_{ii'j}$ are allocated. If the number of binary variables becomes too large for the model to be tractable, then a potential simplification consists of partitioning the blocks into several areas and penalising the excavator when it changes areas. This approximation would reduce the number of binary variables.

Multiple sequential builds The model could be extended to (say) three sequential builds. This would ensure that the most immediate build is not achieved in a manner that is inappropriately prejudicial against future builds. In general terms the model would become very much larger:

- there are variables for each of the three builds;
- there are constraints to ensure that blocks mined in earlier builds are not mined in later builds (and precedences are removed as required);

- starting tonnages for stockpiles for a build are taken as the ending tonnages from the previous build;
- initial positions of excavators remain the same throughout the three builds, or alternatively the user must provide data as to the initial positions for each build.

Exposed ore It is often desirable to have some exposed ore in the pit. This allows the mining engineer to access additional ore quickly should it be required, for example, to cope with equipment breakdowns or unexpected grade results caused by the inherent grade uncertainty. On the other hand, it is expensive to keep a large amount of ore uncovered, so there should be just enough exposed ore for these foreseeable emergencies.

The objective function in our model tends to minimise the amount of exposed ore, since mining extra waste can only increase t . The practical solution to this problem is for the engineer to reserve some mining capacity outside of this optimisation, to be applied to pre-stripping waste in accordance with their professional judgement.

5 Evaluation of a heuristic to select blocks in an iron ore mine

This section is a preliminary evaluation of a heuristic for dealing with the uncertainty in the grade of blocks prior to extraction. We assume that we have a set of blocks available for mining with a range of predicted grades and a range of uncertainties for those predictions. We explore the intuitive idea that blocks with high uncertainty should be excavated first, and that the later blocks in a build should have a low uncertainty. With this choice, we hope to be able to compensate precisely for a potential deviation from target grade by having ‘reliable’ low-variance blocks to add towards the end

Table 1: Types of block in the case study

uncertainty	standard deviation σ	medium grade (60%)	high grade (63%)	low grade (57%)
low	1%	type I	type II	type III
high	3%	type IV	type V	type VI

of a build. In contrast, keeping the uncertain blocks until the end may risk failing to meet the build target.

As an illustration of this idea, consider a trivial example with three types of block available of equal mass. One type of block, A, has uncertain grade: it is equally likely to be 50% or 100%. We are certain that the grades of the other two types of block, B and C, are exactly 50% and 100% respectively. Now suppose that we are to produce builds that have exactly the mass of two blocks with grade 75%. If we start by mining a block of type A, then when we discover its actual grade we add a block of type B or type C as necessary to meet the grade requirement. If instead we start by mining a block of type B or type C, then it is not certain that mining a block of type A will enable us to meet the build target. Unless we mine the blocks high uncertainty (type A) first, we can never mine them without risking missing the build target.

To further explore the approach, we take a simplified case study. Consider 120 blocks of equal weight (8,000 tonnes). We assume that there are no block precedence constraints: each block can be excavated independently of the others. There are two levels of uncertainty in our knowledge of the grades of the blocks: low and high and these have standard deviations of 1% and 3%, respectively. There are three grade levels, low, medium and high (57%, 60% and 63%, respectively). Thus there are six different types of block (Table 1).

We assume there is an equal number (20) of each type of block. The build target corresponds to 100 blocks (800,000 tonnes) with average grade 60% (medium grade). We assume that a small error of 0.1% is tolerated for the

grade, that is the final grade should belong to the interval $[59.9, 60.1]\%$. In reality the tolerance would be likely to be asymmetric as undershooting the grade is worse than overshooting.

To make a valid comparison of different heuristics we require them to use the same number of blocks of each level of uncertainty. For the present case both our heuristics use exactly half blocks with low uncertainty and half blocks with high uncertainty.

Our suggested heuristic ('decreasing uncertainty') involves first mining blocks with high uncertainty (20 each of types IV, V, VI). This is followed by the extraction of 20 medium grade blocks with low uncertainty. Then, among the 40 blocks of types II and III, 20 blocks are selected so as to meet the grade target of 60%. If the average grade is too low after 80 blocks, then the heuristic selects mostly (potentially exclusively) blocks of type II to increase the average grade up to the target. Conversely, if the average grade is too high after the first 80 blocks, then type III blocks are favoured subsequently.

We compare this heuristic with another 'extreme' approach ('increasing uncertainty'). This starts with the blocks with low uncertainty (20 each of types I, II, III), followed by 20 medium grade blocks with high uncertainty (IV). Then, among the 40 blocks of types V and VI, 20 blocks are selected so as to meet the grade target of 60%.

So, the leeway given to each heuristic lies in the last 20 blocks (to choose from among 40). This choice might not be enough to meet the target, in which case we call the build 'infeasible'. For example, regarding the heuristic 'decreasing uncertainty', the average grade after excavating blocks IV, V, VI, and I might be so low (say, 57%) that even adding only the high-graded blocks of type II is not enough to reach the target of 60%. Conversely, the grade might be so high that the average cannot be readjusted with low-grade blocks (III).

Since the remaining blocks have uncertain grades, we may not be able to guarantee that the build is 'feasible' (that is, the target is met). However, we would like to ensure that the build is feasible with a certain probability,

Table 2: Notation and values for the heuristic method case study.

Symbol	Meaning	Value
n	The number of blocks excavated so far	$\{81, 82, \dots, 100\}$
g_n	The average grade of the first n blocks (%)	
s_n	The standard deviation of the grade of the first n blocks (%)	
N	The number of blocks of the build target	100
T	The grade of the build target	60%
ϵ	The tolerance of the build target	0.1%
M_i	The tonnage of block i	8000 tonnes
F_i	The average grade of block i	57, 60 or 63%
σ_i	The standard deviation of the grade of block i	1 or 3%
g^H	The highest average grade for the remaining blocks	63%
g^L	The lowest average grade for the remaining blocks	57%
z_p	The p -quantile of the standard normal distribution	$z_{0.01} = -z_{0.99} = -2.32$

say 0.99. Two questions then arise.

- Given a certain number of blocks already excavated, what property should the current average grade satisfy to ensure that the build is feasible with probability 0.99?
- Given a certain number of blocks already excavated, how likely is it that the build is feasible with probability 0.99? Does one heuristic outperform the other in this regard?

To answer these questions, we introduce the notation in [Table 2](#).

5.1 Criterion for build feasibility.

Suppose that n ($\leq N$) blocks have been selected, excavated and included in a build. In order for the build to be feasible, we must ensure that:

- by selecting only high-grade blocks as last blocks of the build, the final average grade is higher than the target grade (minus the tolerance) with probability at least 99%;
- by selecting only low-grade blocks as last blocks of the build, the final average grade is lower than the target grade (plus the tolerance) with probability at least 99%.

These conditions correspond to the following inequalities:

$$\frac{g_n \sum_{i=1}^n M_i + g^H \sum_{i=n+1}^N M_i}{\sum_{i=1}^N M_i} + z_{0.01} \frac{\sqrt{\sum_{i=n+1}^N (M_i \sigma_i)^2}}{\sum_{i=1}^N M_i} \geq T - \epsilon; \quad (4)$$

$$\frac{g_n \sum_{i=1}^n M_i + g^L \sum_{i=n+1}^N M_i}{\sum_{i=1}^N M_i} + z_{0.99} \frac{\sqrt{\sum_{i=n+1}^N (M_i \sigma_i)^2}}{\sum_{i=1}^N M_i} \leq T + \epsilon. \quad (5)$$

We explain Inequality (4), which concerns undershooting the build target. The first term on the left-hand side is the expected final grade if all the remaining ($n + 1, \dots, N$) blocks are chosen to be high grade. The second term on the left-hand side is the adjustment required if, within the variation for high grade blocks, the grades of these selected blocks are at the small value, first percentile. (A random standard normal variable will be greater than $z_{0.01} = -2.32$ with probability 99%.) The right-hand side is the lowest build grade accepted (59.9%). By satisfying this inequality, we ensure that, by selecting high-grade blocks, the final average grade will be high enough with probability at least 99%. The origins of Inequality (5), which concerns overshooting the build target, are similar.

Rearranging and combining the two inequalities, and substituting in the

p-quantiles (assuming normality),

$$g_n^{\min} \leq g_n \leq g_n^{\max} \quad (6)$$

where

$$g_n^{\min} = \frac{(T - \epsilon) \sum_{i=1}^N M_i + 2.32 \sqrt{\sum_{i=n+1}^N (M_i \sigma_i)^2} - g^H \sum_{i=n+1}^N M_i}{\sum_{i=1}^n M_i}, \quad (7)$$

and

$$g_n^{\max} = \frac{(T + \epsilon) \sum_{i=1}^N M_i - 2.32 \sqrt{\sum_{i=n+1}^N (M_i \sigma_i)^2} - g^L \sum_{i=n+1}^N M_i}{\sum_{i=1}^n M_i}. \quad (8)$$

So, g_n must belong to a certain interval $[g_n^{\min}, g_n^{\max}]$ for the build to be feasible.

In the case study example, we calculate the feasibility interval with n varying from 81 to 100, for both the ‘decreasing uncertainty’ and ‘increasing uncertainty’ heuristics (Figure 6). The interval $[g_n^{\min}, g_n^{\max}]$ is wider for ‘decreasing uncertainty’ than that for ‘increasing uncertainty’ because the standard deviations for the last 20 blocks are lower, resulting in a smaller g_n^{\min} (Equation (7)) and a larger g_n^{\max} (Equation (8)). Also, the interval tends to decrease in size as n grows, and equals $[T - \epsilon, T + \epsilon]$ when $n = N$, giving it a cone shape. We call the set of couples (n, g_n) for which the build is feasible the ‘feasibility cone’.

5.2 How likely is it that the blocks excavated so far belong to the feasibility cone?

We now focus on the blocks accumulated between 1 and n . Since the grade in each block i is assumed to be sampled from a normal distribution of mean F_i and standard deviation σ_i , the expected value of the average grade after

n blocks have been excavated is a weighted average of the grades of each block:

$$E[g_n] = \frac{\sum_{i=1}^n M_i F_i}{\sum_{i=1}^n M_i}. \quad (9)$$

At 80 blocks ($n = 80$), this expected grade is the same for both heuristics; it is the weighted average of 60% (20 blocks of 8,000 tonnes), 63% (20 blocks of 8,000 tonnes), 57% (20 blocks of 8,000 tonnes), and 60% (20 blocks of 8,000 tonnes). This value is 60%.

The standard deviation of the average grade of the first n blocks is

$$s_n = \frac{\sqrt{\sum_{i=1}^n (M_i \sigma_i)^2}}{\sum_{i=1}^n M_i}. \quad (10)$$

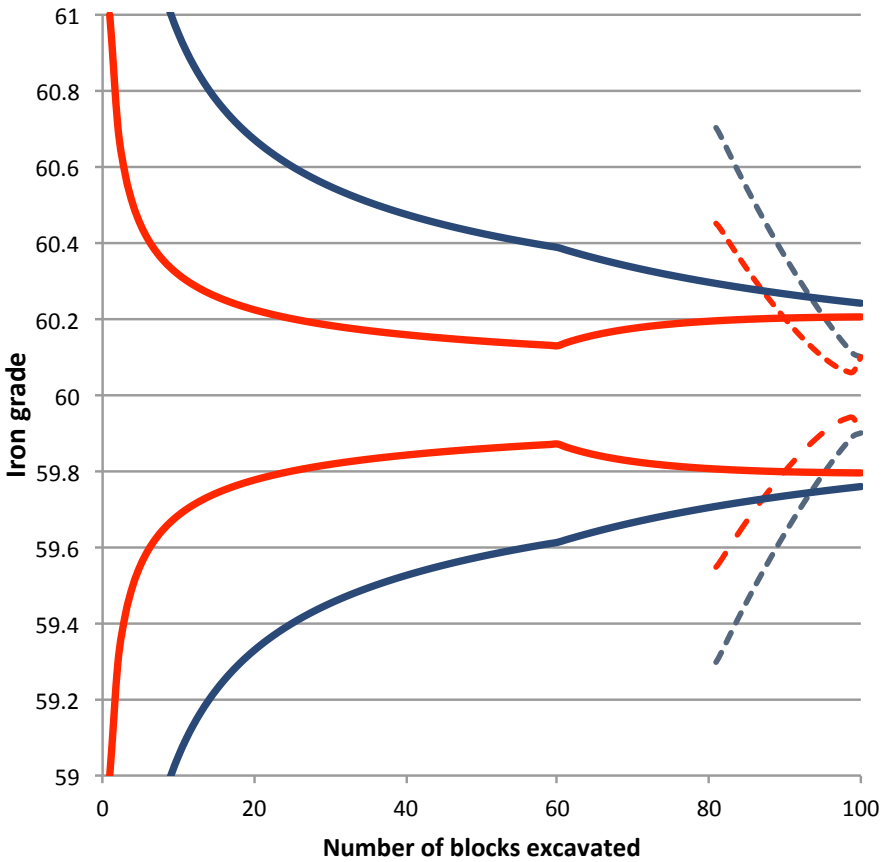
In addition to the feasibility cone described in the previous [Subsection 5.1](#), we find a ‘standard deviation cone’ (for both heuristics and all values of $n \in 1, 2, \dots, 100$). This contains grades within the target grade (60%) plus or minus one standard deviation. [Figure 6](#) shows the standard deviation and feasibility cones of both heuristics for the case study example. The standard deviation cone of heuristic ‘increasing uncertainty’ is contained in that of ‘decreasing uncertainty’ because at any time during construction of the build, the set of blocks ‘increasing uncertainty’ has used has a lower uncertainty than those of ‘decreasing uncertainty’. Countervailing this, the heuristic ‘decreasing uncertainty’ with lower-variance later blocks, has a wider feasibility cone than ‘increasing uncertainty’.

Now we calculate how likely the average grade is to fall in the feasibility cone. The half-width of the feasibility cone is

$$\frac{g_n^{\max} - g_n^{\min}}{2} = \frac{\epsilon \sum_{i=1}^N M_i - 2.32 \sqrt{\sum_{i=n+1}^N (M_i \sigma_i)^2} + \frac{g^H - g^L}{2} \sum_{i=n+1}^N M_i}{\sum_{i=1}^n M_i}. \quad (11)$$

The feasibility and the standard deviation cones are both centred on the target grade (60%), which simplifies the calculation of the probability of

Figure 6: Standard deviation and feasibility cones for the case study example with heuristic ‘decreasing uncertainty’ (blue) and ‘increasing uncertainty’ (red). The solid lines represent the target grade (60) plus or minus one standard deviation. The dashed lines depict the feasibility cones, from 81 to 100 blocks excavated.



- Feasibility cone - decreasing uncertainty
- Feasibility cone - increasing uncertainty
- Standard deviation cone - decreasing uncertainty
- Standard deviation cone - increasing uncertainty

feasibility $\mathbf{p} = \Pr(\mathbf{g}_n \in [\mathbf{g}_n^{\min}, \mathbf{g}_n^{\max}])$. This probability is obtained using the error function and depends on the half-width and standard deviation,

$$\begin{aligned} \mathbf{p} &= \operatorname{erf}\left(\frac{\frac{1}{2}(\mathbf{g}_n^{\max} - \mathbf{g}_n^{\min})}{\sqrt{2}\mathbf{s}_n}\right) \\ &= \operatorname{erf}\left(\frac{\epsilon \sum_{i=1}^N M_i - 2.32\sqrt{\sum_{i=n+1}^N (M_i\sigma_i)^2} + \frac{1}{2}(\mathbf{g}^H - \mathbf{g}^L) \sum_{i=n+1}^N M_i}{\sqrt{2}\sqrt{\sum_{i=1}^n (M_i\sigma_i)^2}}\right). \end{aligned} \quad (12)$$

In our case study, for the final 20 blocks, this probability simplifies to

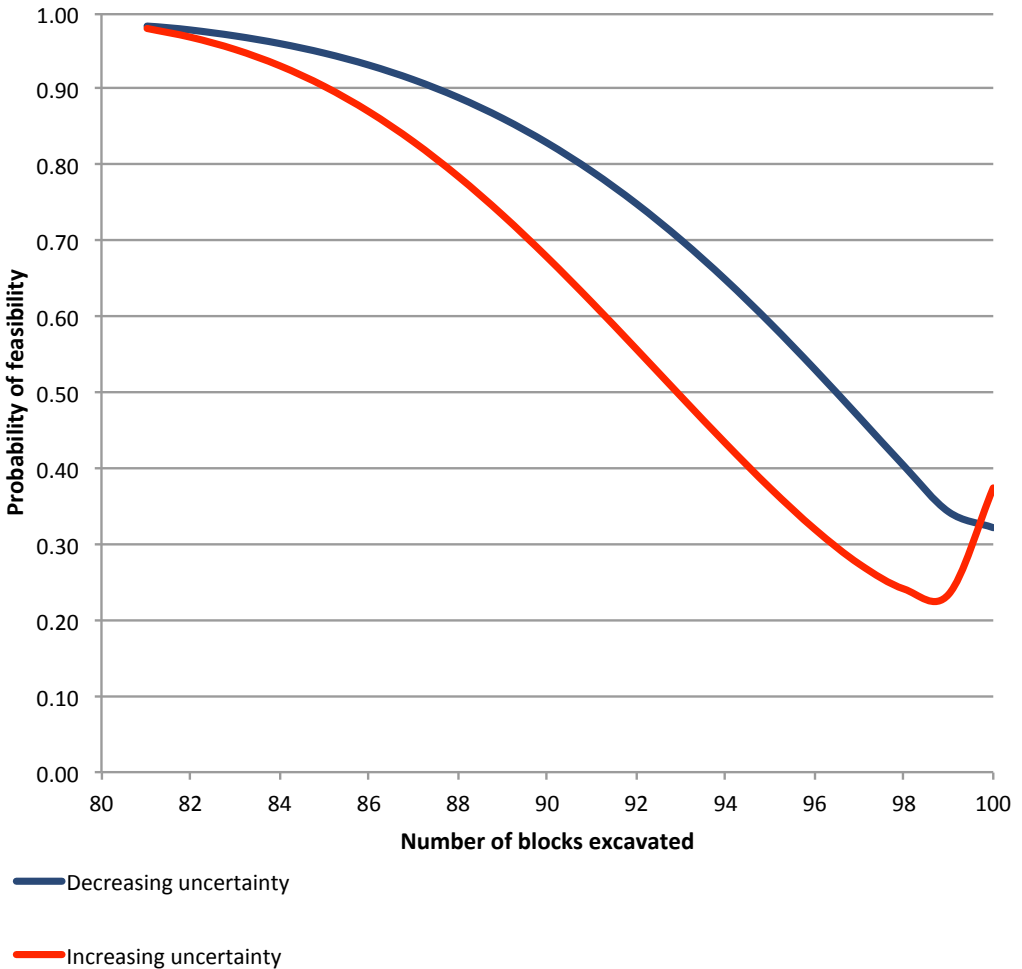
$$\mathbf{p} = \operatorname{erf}\left(\frac{20 - 4.64\sqrt{(100 - \mathbf{n})} \sigma_1 + 6(100 - \mathbf{n})}{\sqrt{2} \sqrt{(\mathbf{n} - 60)\sigma_1^2 + 60\sigma_2^2}}\right) \quad (13)$$

where σ_1 is 1% for the decreasing uncertainty heuristic and 3% for the increasing uncertainty heuristic and σ_2 is 3% for the decreasing uncertainty heuristic and 1% for the increasing uncertainty heuristic. [Figure 7](#) shows the probability of feasibility, \mathbf{p} , for the final 20 blocks; that is, for $\mathbf{n} \in \{81, 82, \dots, 100\}$. The probability of feasibility with the heuristic ‘decreasing uncertainty’ is consistently higher than that of ‘increasing uncertainty’, for example 70% against 50% with 93 blocks excavated. This means that the necessity to adjust the average grade at the end of a build is less likely (and happens later) when starting with the high variance blocks than when starting with the low variance ones.

5.3 Discussion

In this [Section 5](#) we designed a simple case study modelling the optimisation of an iron ore mine. [Sections 5.1](#) and [5.2](#) explored the hypothesis that the blocks with high uncertainty should be used when starting a new build, following the intuitive idea that the low uncertainty blocks are important at the end of a build to adjust a possibly deviant average grade.

Figure 7: Probability of feasibility of the build for the heuristics for a number of blocks ranging from 81 to 100: decreasing uncertainty (blue) and increasing uncertainty (red).



Subsection 5.2 introduced the concept of ‘feasibility cone’ to describe the intervals which the average grades should belong to in order to meet the target with high probability. We then show how to calculate the probability that the average grade falls in this interval, for the two heuristics: ‘increasing uncertainty’ and ‘decreasing uncertainty’. We found that starting with high uncertainty blocks leads to a higher probability of meeting the target.

One aspect disregarded so far is that the heuristic ‘decreasing uncertainty’ was naturally disadvantaged. Out of 100 blocks, 60 are of high uncertainty whereas 40 are of low uncertainty. By contrast, the heuristic ‘increasing uncertainty’ uses 40 high uncertainty blocks and 60 low-variance blocks. So, not only does ‘decreasing uncertainty’ achieve better results, but it does so with higher-variance blocks and leaves low uncertainty blocks for the next build.

This preliminary test can be extended in a number of ways. In our case study, all blocks weigh the same, and only three levels of grades and two levels of standard deviation are used. Because this is unlikely in real-world problems, this study does not claim to prove that blocks should always be excavated in a strict order solely based on the uncertainty. Rather, it suggests that the grade uncertainties are an important parameter to consider when searching for the optimal schedule. In a real-world problem, the optimal solution is probably a non-trivial trade-off between choosing blocks with decreasing uncertainty, and choosing the ones that actually meet the target.

We also disregarded the precedence constraint, which should require another type of heuristic or optimisation technique to be accounted for.

6 Simulating the build construction process

A MATLAB program was written to simulate the build construction process. The focus here is on managing activities in order to meet build orders. The

process involves a number of factors such as diggers, stockpiles, wastepiles and the crusher, as illustrated in [Figure 1](#).

We assume that there are two diggers. Each digger unearths one block of ore per time period. Each block is numerically graded according to the quality of the ore. For example, the lowest acceptable grade could have a value of 55% and the highest acceptable grade a value of 62%. Once unearthed, a block may be sent either to the crusher or to a stockpile. Waste blocks (with a grade below 55%) are discarded. In our simulation, one digger unearths blocks with grades uniformly distributed in $[50, 62]\%$; the other in $[46, 58]\%$.

We create one stockpile for each acceptable grade. In our example there are eight stockpiles for each integer grade from 55% to 62%. Each stockpile is initialised with ten blocks and has unlimited capacity. The single crusher processes two blocks per time period. The crusher also works on just one build at a time. A build is comprised of twenty blocks and so it is constructed in exactly ten time steps.

The grade of the completed build is the average grade of the blocks in the build. Each build has a target grade; we set this to 58% in our simulation. The blocks that make up a build come either straight from the diggers or from one of the stockpiles. Thus, in any single time period there is a choice about which two blocks the crusher should add to the current build. This decision is guided by the *grade envelope*.

The grade envelope at each time step defines the lowest and highest block grades that can be added to the current build without making it impossible to achieve the target. For example, suppose that 15 blocks have already been added to the build and that it currently has an average grade of 57%. To define the highest grade that we are willing to accept for block 16, we suppose that the rest of the build will be made up entirely of blocks from the lowest grade of 55%. This means that the upper grade limit for the block under consideration is $20 \times 58 - 15 \times 57 - 4 \times 55 = 85\%$; if the grade of the block under consideration is greater than this, then we produce a build with a final grade greater than the target. On the other hand, if the rest of the build is made

up entirely of blocks from the highest grade of 62%, then the lowest grade limit for the block under consideration is $20 \times 58 - 15 \times 57 - 4 \times 62 = 57\%$; if the grade of the block under consideration is lower than this, then we will not be able to achieve the target. Thus, the grade envelope for the current block under consideration is $[57, 85]\%$; we only add it to the build if its grade falls within this envelope. At the start of the build, the grade envelope is large. The envelope progressively narrows as the build is constructed.

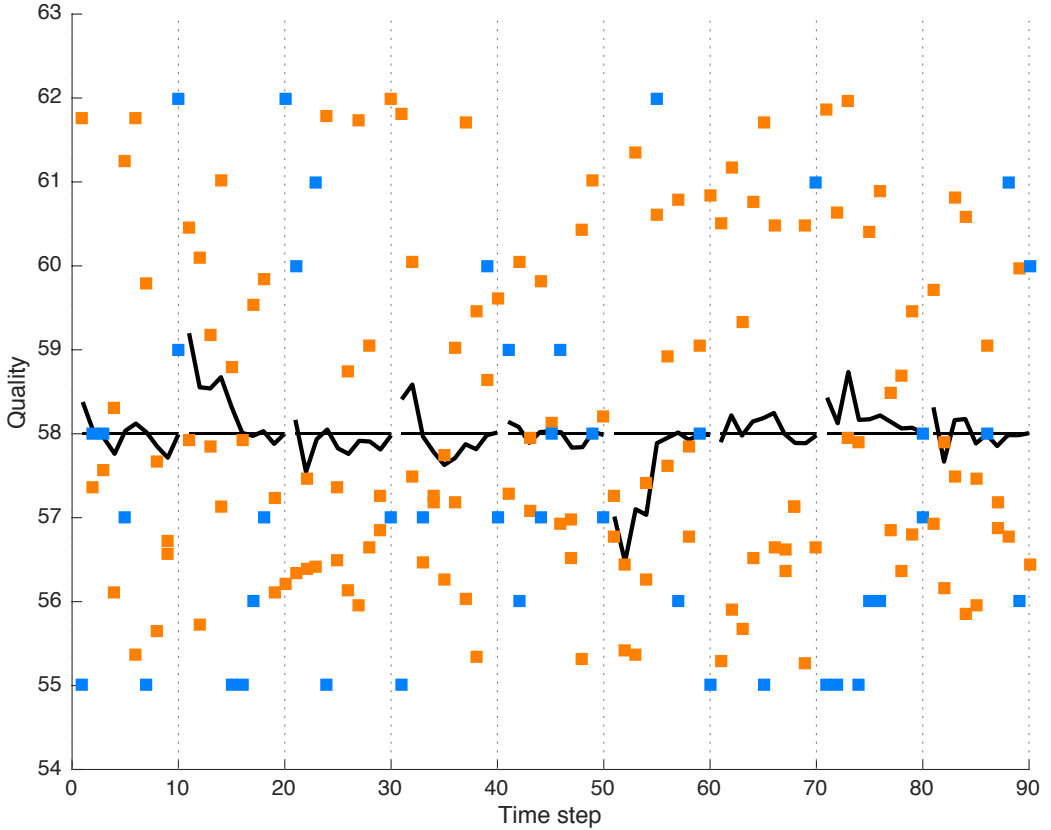
At each time step of the simulation, a block from a digger is sent to the crusher if it is within the current grade envelope. Otherwise it is sent to a stockpile. If fewer than two blocks are sent to the crusher in a time step, then blocks are taken from the stockpile with grade closest to that required to achieve the final target grade of the build.

Figure 8 shows the simulated construction of nine builds. Each build takes ten time periods to complete. In each time step, two blocks are added to the build: an orange block is from a digger, a blue block is from a stockpile. The black line shows the average grade of the current build.

6.1 Future work

The grade envelope calculation assumes that there are enough blocks available of the lowest and highest grade; this is not always the case and should be taken into account in a future version of the simulation. If in each time step the number of blocks from diggers exceeds the crusher rate, then we need to decide which digger blocks to send to the crusher. This has not been implemented in the simulation.

Figure 8: Simulation of nine builds. The black line indicates the quality (grade) of the current build. Coloured squares are blocks of indicated quality being added to the build from a digger (orange) or from a stockpile (blue).



7 Conclusion

We addressed the problem of sequencing excavation in an open pit mine from a number of different angles.

Sequencing of excavation of the necessary blocks may be facilitated by the use of a mixed integer programming model, as described in [Section 4](#), that takes into account block precedence and distances of blocks from the starting location of the excavator. Valuable insights could be gained from running an implementation of such a model. There was not time to do this during the period of the Study Group.

The main issue is the uncertainty in the grade of each block, and, in [Section 5](#), we advanced the hypothesis that blocks of high uncertainty should be included in a build in its early stages, with lower uncertainty blocks used toward the end of the build to achieve the target average grade. The build simulation ([Section 6](#)), shows how recovering material from stockpiles can effect builds of required quality.

Studies employing realistic data could refine and prove these observations.

Acknowledgement We are grateful to Schneider Electric and the industry representative Adam Ghandar for bringing this problem to MISG-2016 and for his valuable input. We also acknowledge and thank other team members who worked on the problem: Erika Belchamber, Elizabeth Bradford, Ajini Galapitige, Judith Shand, Markus Stumptner, Graeme Wake, Brian Webby, Youngho Woo, Peng Zhou. The hospitality of our hosts at University of South Australia was appreciated.

References

- [1] M. Menabde, G. Froyland, P. Stone, G. Yeates (2005), “Mining schedule optimisation for conditionally simulated orebodies”, in Dimitrakopoulos R. (ed.) *Orebody modelling and strategic mine planning: uncertainty and risk management models*, Australasian Inst. Mining and Metallurgy, pp. 343–357, ISBN:1-920806-42-3 [M35](#)
- [2] M. Ibrahimov, A. Mohais, S. Schellenberg, Z. Michalewicz (2014). “Scheduling in iron ore open-pit mining”, *The International Journal of Advanced Manufacturing Technology*, 72(5–8), 1021–1037. doi:[10.1007/s00170-014-5619-8](https://doi.org/10.1007/s00170-014-5619-8) [M35](#)
- [3] L. Caccetta, S. P. Hill (2003) “An Application of Branch and Cut to Open Pit Mine Scheduling” *Journal of Global Optimization*, 27, 349–365. doi:[10.1023/A:1024835022186](https://doi.org/10.1023/A:1024835022186) [M35](#)

Author addresses

1. **Winston L. Sweatman**, Massey University, Auckland, NEW ZEALAND.
<mailto:W.Sweatman@massey.ac.nz>
2. **Kevin White**, University of South Australia, AUSTRALIA.
<mailto:Kevin.White@unisa.edu.au>
3. **Amie Albrecht**, University of South Australia, AUSTRALIA.
4. **Martin Peron**, Queensland University of Technology, and CSIRO, Dutton Park, AUSTRALIA.
5. **Peter Pudney**, University of South Australia, AUSTRALIA.
6. **David Whittle**, University of Melbourne, AUSTRALIA.