Combined optimisation of an open-pit mine outline and the transition depth to underground mining

D Whittle\textsuperscript{a}, M Brazil\textsuperscript{b}, P A Grossman\textsuperscript{a}, J H Rubinstein\textsuperscript{c}, D A Thomas\textsuperscript{a}

\textsuperscript{a}Dept. Mechanical Engineering, The University of Melbourne, Parkville, Vic 3010, Australia
\textsuperscript{b}Dept. Electrical & Electronic Engineering, The University of Melbourne
\textsuperscript{c}Dept. Mathematics and Statistics, The University of Melbourne

Abstract

Miners harvest minerals from ore-bodies in the ground by a variety of specialised mining methods, with most falling into the categories of open-pit and underground. Some ore-bodies are harvested by a combination of open-pit and underground methods. In these cases there is often material that could be mined by either method, and an economic choice has to be made. This is referred to as the transition problem and it has received some attention in the mining literature since the 1980s and more recently has had attention in the mathematics literature. The transition problem is complicated by the need in many cases to leave a crown pillar (un-mined rock above the underground mine) and for this crown pillar to have a prescribed shape.

We have developed a method to optimise the design of an open-pit mine, while solving the transition problem and taking into account the need for a crown pillar with a prescribed shape. We base it on an existing method to optimise the design of an open-pit mine, framed as a maximum graph closure problem. Our method introduces non-trivial strongly connected sub-graphs (NSCSs) of the graph, a complication that previous authors on maximum graph closure problems do not appear to have covered. To obviate the need to check every method for compatibility with NSCSs, we reduce the problem to an equivalent problem without them. This has the added advantage of reducing overall processing time in cases where the number of NSCSs is large.

Keywords:
OR in natural resources, Graph theory, Cost benefit analysis, Strongly connected components, Closure problem
1. Introduction

In this introduction we first present some general mining, planning and optimisation background information, then review the literature on the topic of optimising the transition from open-pit to underground.

Mining is a global industry producing around 17 billion tonnes of mineral fuels, metals and industrial minerals annually (2014 [1]). The vast majority of minerals are extracted by methods falling into the categories of open-pit (an open excavation from the surface) and underground (a network of tunnels and/or shafts giving access to the minerals underground). Some ore-bodies are harvested by a combination of open-pit and underground methods, either starting with an open-pit followed by underground mining or less commonly, the reverse. A few example Australian mines are Telfer (A Newcrest owned gold mine that transitioned from open-pit to underground); Sunrise Dam (An AngloGold gold mine that transitioned from open-pit to underground) and Golden Grove (an MMG owned copper and zinc mine that transitioned from underground to open-pit) [2]. Before commencing any such transition a project study is conducted in which strategic mine plans are created. The study may take many months or years. An example is the Grasberg copper mine in Indonesia owned by Freeport-McMoRan. Study work on the transition from open-pit to underground commenced some time before 1995 but was not completed until 2008. Construction commenced in 2011, and first production is forecast for 2018. The expected capital cost is US$6b [3]. With such large sums of money at stake, it is very important to correctly set key design parameters, such as the extents of open-pit and underground mines.

Any kind of mine planning exercise relies on having a model of the ore-body in the ground. In the vast majority of cases, the ore-body is represented in a regular block model. This is a model in which each block record in a regular three-dimensional grid carries information about the rock type and mineral grades. In this paper, we restrict our interest to regular block models.

The application of mathematical optimisation to open-pit mine outlines is widespread, beginning in the 1980s with commercial implementations of a graph algorithm developed by Lerchs & Grossmann [4]. This algorithm, commonly referred to as the LG Algorithm, provides an exact method to optimise the outline of an open-pit mine, in order to maximise its un-discounted cash value. The LG Algorithm is in wide use in the mining industry, being available from many of the largest mine planning software vendors (Dassault Systèmes; Maptek; Hexagon Mining). The problem can also be framed
as a maximum flow problem [5]. Maximum flow methods in industrial use include the push-relabel algorithm [6] (used by Mincom and Minemax); and Hochbaum’s pseudoflow [7] (used by Dassault, Deswik and Muir & Associates Computer Consultants). Some of these maximum flow implementations provide significantly better performance than LG implementations. For example Dray [8] found that Minemax Planner is significantly faster than Dassault Systèmes LG Algorithm implementation, particularly for large block models (e.g. models with more than 5 million blocks). Importantly, Dray also found that the two optimisers yielded the same results.

These pit optimisers decide the outline of the pit in order to maximize un-discounted cash values and they can do so for very large and detailed models (hundreds of thousands or millions of blocks). The computing times vary from seconds to a few hours, depending on the number of blocks and the complexity of the constraints applied to the model. However, miners would rather work with discounted cash-flows and maximize net present value (NPV). Accordingly, pit optimisers are almost always used in a structured process that pursues high NPV solutions for a wide range of planning decisions. The process involves running various optimisers multiple times with different data inputs (For example see [9] and [10]). With long planning processes (for example the multi-year Grasberg case), and with efficient pit optimisers available, pit optimisation is generally not on the critical path. However fast optimisation still offers the advantage of allowing a wider range of alternatives to be tested and for larger, more detailed models to be used.

There are many different types of minerals deposits, including those with valuable materials such as gold, copper, nickel and diamonds, and for bulk materials such as coal. In this paper we are generally interested in mines for valuable materials that are extracted from open pits and/or from underground mines. Most open pit mines for valuable materials are amenable to the open pit optimisation methods discussed in this paper.

In underground mines, we limit our interest to mines that employ stoping or caving methods. We are not concerned with the details of the underground methods as we really only require a model containing underground mine values for each block in the model, representing the benefit of including the block in the underground mine. That value is derived from an underground mine plan. The reason for this limited concern with the details of the underground plan will become apparent later in the paper. However, for the benefit of readers unfamiliar with mining, we provide some general information: In a stoping mine, tunnels are driven into the ore-body, and the ore is blasted with explosives, and removed by mechanical means from constructed drawpoints. The resulting voids may be left open, or they may be backfilled.
with waste and cement in order to avoid weakening of the surrounding rock. In a caving mine, miners rely on gravity to collapse the rock down to the drawpoints where it is extracted. Since the ground collapses in (sometimes all the way up to the Earth’s surface), there should not be any voids left over in a caving mine. In either stoping or caving mines, drawpoints are accessed by networks of shafts and tunnels, which also provide ventilation, and access for exploration and blasting. The three-dimensional shapes that are targeted for mining are called stopes in a stoping mine and caving blocks in a caving mine, however to avoid confusion with regular blocks discussed in the paper, we will refer to them as caving polygons. A collection of blocks in a regular block model can be used to represent a stope or a caving polygon.

The application of optimisation in design of underground mines is less mature than it is for open-pit mines. However a variety of heuristics and exact methods have been developed to separately optimise the outline of a stope (e.g. Alford & Hall [11], Bai, Marcotte & Simon [12]), the outline of a caving polygon (e.g. Diering [13]) and tunnels (e.g. Brazil, Grossman, Rubinstein & Thomas [14], Sandanayake, Topal & Asad [15], and Sirinanda, Brazil, Grossman, Rubinstein & Thomas [16]).

Various authors have tackled the issue of the combined optimisation of open-pit and underground mines and some have focused in particular on the transition problem (optimising the economic decision as to where to stop the open-pit and where to start the underground mine). J Whittle [17] incorporated a method into pit optimisation software that takes into account the value that ore has if mined by an underground method. Consider a case in which some blocks can be mined by either the open-pit or underground methods. For any such block, the value used for pit optimisation should be the difference between its open-pit value and its underground value. The assumption underpinning this is that for a block that can be mined by either method, if it is not mined by the open-pit method, it will be mined by the underground method. Camus [18] independently described an approach that will generate equivalent results. We will henceforth refer to this as the opportunity cost approach to solving the transition problem (“opportunity cost approach” for short), following the terminology used in the field of economics (For example see McTaggart, Findlay and Parkin [19]).

Definition 1 (Opportunity cost). Let \( v_c \in \mathbb{R} \) and \( v_d \in \mathbb{R}_{\geq 0} \) be the net values for mutually exclusive alternatives \( c \) and \( d \). If no other alternative to \( c \) has a higher value than \( d \), then the opportunity cost for \( c \) is \( v_d \).

In our case, the mutually exclusive alternatives with respect to any given block are to mine it by open-pit method (alternative \( c \) in Definition 1), or
by underground method (alternative \(d\)). If we mine a block by open-pit method, we gain its open-pit value \((v^c)\), but we lose the value that would have been gained if it had instead been mined by underground method \((v^d - \text{the opportunity cost for } c)\). When the opportunity cost approach is applied we subtract the underground value from the open-pit value for each block, before doing pit optimisation. When applying this approach, as opposed to optimising a pit without regard for the opportunity cost, the optimised open-pit is almost always smaller. Also, the value of the open pit mine is lower, but the sum of the values of the open-pit and underground mines is maximized. The reason for this is given later in the paper (Section 2.2). There is more than one way to generate a smaller pit using pit optimisation software; for example, it is common to use a technique called \textit{pit parameterization} to generate a family of pits by flexing the commodity price (for example see \[9\]). However, the pit created using the opportunity cost approach may not match the shape and tonnage of any of the pits created using parameterization techniques due to the different ways in which the block values are calculated.

Chen, Gu & Li \[20\] described a method similar to the opportunity cost approach. They did not use exact optimisation, but they did include consideration of a \textit{crown pillar}, which the earlier authors had not. A crown pillar is a body of rock left in place above the shallowest part of an underground mine to ensure stability in the surrounding rock. The need for stability is driven by the land use above the underground mine, which in some cases is an open-pit mine. The crown pillar also acts to reduce or avoid the ingress of water to the underground mine and ensure the stability of the cavity below. When crown pillars are used, their design must take into account the geotechnical characteristics of the native rock and the planned sizes and shapes of the underground stopes \[21\]. Note that the use of crown pillars is not universal, since collapse of the ground above the underground mine is sometimes a desired or acceptable outcome.

Pit optimisation can be incorporated into a wider workflow with other optimisation tools to support a wide range of planning decisions as previously mentioned, and this is the case also when dealing with the transition problem. Finch & Elkington \[22\] advocate for automated scenario analysis in which a number of candidate transition depths are evaluated with schedule optimisation software. Roberts, Elkington, van Orden & Maulen \[23\] provide a case study using a number of different in-house and commercially available optimisation tools. The case study involves an existing underground mine and contemplates an open-pit expansion on ore that would otherwise be mined in a future underground expansion. These authors were able to compare a number of different transition scenarios, each with opti-
mised schedules.

Some authors have applied mixed integer programming methods to the problem, though they are often faced with tractability issues. Chung, Topal & Ghosh [24] formulated an integer programming model to optimise simple open-pit and underground mines with a simple crown pillar. The crown pillar was modelled as a flat exclusion zone with a specified thickness across the full width of the block model. This approach is good for simple cases in which an underground mine begins beneath the deepest part of a pit, but will not cater to a case in which a pit might extend deeper in an area where underground mining is not viable. Chung et al.’s model maximises un-discounted cash-flows. In a case study using a model with 83,000 blocks, the run time was 34.5 hours. Bakhtavar, Shahriar & Mirhassani [25] applied integer programming in the consideration of transition depth, taking into account the crown pillar. They compared a range of transition depths to determine the depth that gave the highest NPV, although only applied to a two-dimensional model. Newman, Yano & Rubio [26] formulated the transition as a large monolithic longest-path network problem. MacNeil & Dimitrakopoulos [27] formulated the open-pit to underground transition as a stochastic optimisation problem. In both cases, the authors incorporated the consideration of a crown pillar, scheduling decisions and mining rate decisions and in order to make the problem tractable, they represented the mines in a set of strata: conceptually horizontal slices of the model. The result is an assignment of strata to open-pit, crown pillar and underground mines respectively. King, Goycoolea & Newman [28] optimise the open-pit and underground mining schedules, the placement of the crown pillar and also the placement of sill pillars (material left in-situ in a particular type of underground mine to allow for a change in mining direction). The open-pit model used in this case is not a regular block model, but a set of open-pit scheduling polygons produced in part by preprocessing a block model with a pit optimiser. This approach has the advantage of reducing a problem with a large number of blocks, to a problem with just hundreds of open-pit scheduling polygons (336 in their case study). This reduction in numbers dramatically improves tractability. However, since the preprocessing steps use regular pit parameterization rather than an opportunity cost approach, the shapes of the open-pit scheduling polygons may be inappropriate. Like Chung, Topal & Ghosh [24], King, Goycoolea & Newman [28] model the crown pillar as a single model-wide exclusion zone.

In summary, there are three general approaches:

1. Use an opportunity cost approach with a pit optimiser.
2. Use a purpose-built integer programming optimisation tool.
3. Use workflows that rely on a combination of optimisation tools and judgement to make a wide range of planning decisions.

None of the approaches can handle the full range of complexities confronting mine planners. The opportunity cost approach can use a very capable and mature pit optimiser that can handle large and detailed models, but cannot model the crown pillar or optimise for NPV directly. The purpose-built optimisation tools can optimise to maximize NPV and can model the crown pillar, but the block model and mining models must be simplified in order to make them tractable. The workflow-based approaches promise to cover the gaps, but would be better served by more capable mathematical optimisation models.

In the next section, we overcome some of these deficiencies by focusing on a new optimisation method for the application of the opportunity cost approach using a pit optimiser that provides for flexible modelling of the physical and economic characteristics of the crown pillar.

2. New Optimisation Method

This section builds on our previous paper [29]. We commence by restating the normal model used for pit optimisation with the LG Algorithm or a maximum flow method and then describe the new model.

2.1. Pit Optimisation Model

We describe two models below: A regular block model is used to represent the material in the ground; an equivalent digraph (a graph with directed edges) is used to frame the optimisation problem.

Recall from the Introduction that a regular block model is a model in which each block record in a regular three-dimensional grid carries information about the rock type and mineral grades. Given price and cost information, it is possible to assign dollar values to each of the blocks. The open-pit value for block $i$ is $m_i = r_i - c_i$ where $r_i$ is the revenue and $c_i$ is the extraction cost. In keeping with the norms of block value calculations for pit optimisation, the cost $c_i$ includes only the cost to mine and remove the rock in the block, to dump and rehabilitate the rock that is classified as waste, and to process the rock that is classified as ore (For further details see [10] or [9]). In the problem at hand we assume that the open-pit value is not time dependent. A block’s open-pit value will be achieved if and only if it is included in the open-pit outline. In an open-pit mine, in order to mine
any block, the blocks above it must also be mined. Moreover, if the walls of the excavation are too steep, they will be unstable and collapse, so a safe maximum pit slope must be maintained. Safe maximum pit slopes typically range from 40 to 55 degrees (measured from the horizontal), depending on the strength of the rock. The open-pit optimisation problem involves finding the pit outline that encloses a subset of blocks such that the sum of the values of the blocks in the subset is maximised, whilst obeying pit slope constraints.

In tackling this optimisation problem, it is convenient to use a digraph (a graph with directed edges, otherwise known as arcs). Blocks in the block model are represented by vertices, and the maximum safe pit slope constraints are modelled with arcs. Let the set $X$ of vertices $x_i$, the set $A$ of arcs $a_k = (x_i, x_j)$ and the set $M_X$ of vertex weights $m_i$ define a digraph $G = (X, A, M_X)$. In this digraph:

- Each vertex $x_i \in X$ corresponds to block $i$ in the block model, and has a weight $m_i \in M_X$, $m_i \in \mathbb{R}$. The weight represents the open-pit value of the corresponding block $i$ in the block model.

- Each arc $a_k = (x_i, x_j), a_k \in A$ represents a mining dependency, specifically relating to the need to uncover a block and to maintain maximum safe pit slopes. If mining the block represented by the vertex $x_i$, is dependent on the block represented by $x_j$ being mined, then the arc $a_k = (x_i, x_j)$ will be included in $A$.

**Definition 2 (Closure).** A closure $G_Y$ of $G$ is a digraph $(Y, A_Y, M_Y)$, defined by a set of vertices $Y \subseteq X$ such that $A_Y$ includes all the arcs in $A$ that have tails in vertices of $Y$. Vertices at the heads of all arcs in $A_Y$ must also be included in $Y$. The vertex weights $m_i \in M_Y, i : x_i \in Y$, are induced by the weights in $G$.

Note that we do not exclude the possibility that a closure $G_Y = \emptyset$. Figure 1 shows an example of a two-dimensional model with arcs (left) and an example of a closure (right). This is a simplistic two-dimensional model for an open-pit. With square blocks, only three arcs per block are required to model 45-degree slopes. To model slopes accurately in a typical 3-dimensional model, thirty or more arcs per block may be needed, including arcs that extend up more than one level.

The above-mentioned pit optimisation problem can now be restated in Graph Theory terms as a **maximum graph closure problem**.
**Definition 3 (Maximum graph closure (MGC)).** A maximum graph closure is a closure $G_Y$ of a digraph $G$ that maximises $\mathcal{M}_Y$:

$$\mathcal{M}_Y = \sum_{i : x_i \in Y} m_i$$

(1)

and such that for each closure $G_X \subset G_Y$, $\mathcal{M}_X < \mathcal{M}_Y$.

Note that this definition for MGC differs from some earlier papers (for example see [5]) by the inclusion of the condition for each closure $G_X \subset G_Y$, $\mathcal{M}_X < \mathcal{M}_Y$. This condition is consistent with the operation of the LG Algorithm, which guarantees to find a maximum graph closure with the smallest number of vertices and this is a desirable feature in pit design. Suppose there were two solutions to the problem, both with the same value but one with more blocks than the other. A miner would much rather see the solution with fewer blocks, implying less work, less time, and less environmental disturbance.

Theorem 1 formalises a proof given by J Whittle in the early 1990s and included in unpublished training material for pit optimisation software.

**Theorem 1.** The maximum graph closure (MGC) is unique.

**Proof.** Suppose a graph $G$ has two distinct non-empty MGCs $G_Y$ and $G_Z$ implying $\mathcal{M}_Y = \mathcal{M}_Z > 0$, and consider these cases:

1. **One MGC is a proper subset of the other:** It follows from Definition 3 that an MGC cannot be a proper subset of another MGC leading to a contradiction.
2. **The two MGCs do not share any vertices:** If $G_Y \cap G_Z = \emptyset$, then $G_Y \cup G_Z$ is a closure that has a value $\mathcal{M}_Y + \mathcal{M}_Z$. Since $\mathcal{M}_Y = \mathcal{M}_Z > 0$, $G_Y \cup G_Z$ is a closure with a higher value than either $G_Y$ or $G_Z$ leading to a contradiction (that is, neither $G_Y$ nor $G_Z$ are MGCs).
3. **The two MGCs share one or more vertices and neither is a proper subset of the other:** Let $\mathcal{M}_{Y \setminus Z}$, $\mathcal{M}_{Z \setminus Y}$ and $\mathcal{M}_{Y \cap Z}$ be the sums of the weights of vertices in $G_Y \setminus G_Z$, $G_Z \setminus G_Y$ and $G_Y \cap G_Z$ respectively. $\mathcal{M}_Y$ and $\mathcal{M}_Z$ are given by Equations 2 and 3.

$$\mathcal{M}_Y = \mathcal{M}_{Y \setminus Z} + \mathcal{M}_{Y \cap Z}$$

(2)

$$\mathcal{M}_Z = \mathcal{M}_{Z \setminus Y} + \mathcal{M}_{Y \cap Z}$$

(3)
Figure 1: A two-dimensional model with arcs representing block dependencies (left) and an example closure (right).

If $G_Y$ and $G_Z$ are closures, then $G_Y \cup G_Z$ and $G_Y \cap G_Z$ are closures and the value of $G_Y \cup G_Z$ is given by Equation 4.

$$\mathcal{M}_{Y \cup Z} = \mathcal{M}_{Y \setminus Z} + \mathcal{M}_{Y \cap Z} + \mathcal{M}_{Z \setminus Y}$$

(4)

Since $G_Y$ and $G_Z$ are both MGCs, $\mathcal{M}_Y = \mathcal{M}_Z$ and given Equations 2 and 3, $\mathcal{M}_{Y/Z} = \mathcal{M}_{Z/Y}$. Hence $\mathcal{M}_{Y \cup Z} = 2\mathcal{M}_{Y \setminus Z} + \mathcal{M}_{Y \cap Z}$.

Since $G_Y \cap Z$ is a proper subset of $G_Y$, $\mathcal{M}_{Y/Z} \neq 0$ (by Case 1). Consider the remaining alternatives:

- If $\mathcal{M}_{Y \setminus Z} > 0$ then $G_Y \cup Z$ is a closure with a higher value (a contradiction).
- If $\mathcal{M}_{Y \setminus Z} < 0$ then $G_Y \cap Z$ is a closure with a higher value (a contradiction).

Definition 4 (Maximum graph closure problem). This is the problem of finding an MGC for a given digraph.

Figure 2 illustrates an optimal pit design. In this simple two-dimensional example, pit slopes are 45 degrees and there are just a few blocks. In practice, models are three-dimensional; have tens of thousands to millions of blocks; with thirty or more arcs per block to represent complex pit slope requirements. Commercial pit optimisers are efficient enough to solve these large problems in practical time-frames.

2.2. New Model

We define a new model as follows. $G = (X, A, M_X)$ is a digraph representing the open-pit optimisation problem with underground option and allowance for a well-formed crown pillar. Note that we illustrate the concepts that follow in figures showing trivial numbers of blocks. This does not imply that the methods only work for small models and we demonstrate an industrial-size application in Section 5.
2.2.1. Vertices and Weights

Vertices are divided into two sets corresponding to two copies of the set of blocks (See Figure 3):

- $x^p_i \in X^p$ are vertices corresponding to blocks $i$ in the block model with weights set to the open-pit value of the block: $m^p_i = r^p_i - c^p_i$ (as in regular pit optimisation model).

- $x^u_j \in X^u$ are vertices corresponding to blocks $j$ in the block model with weights set to the negative underground mining value of the block: $m^u_j = -(r^u_j - c^u_j)$. This is a negative value since the effect of including this vertex in a closure $G_Y$ is that it is no longer available for underground mining, and so its underground mining value is lost. The important assumption here is that if this vertex is not included in a closure $G_Y$, then the block it represents will be included in an underground mine and its underground mining value will be realised. In micro-economic terms, the vertex weight is an opportunity cost (See Definition 1) with respect to the opportunity to mine a corresponding block in $X^p$ represented by a vertex $x^p_i$ and connected to this vertex via an arc $a^\gamma_k = (x^p_i, x^u_j)$ (see below).

The underground mining values for blocks are always positive. They represent the value of including a block in the underground mine. Consequently the corresponding vertex weights are always negative. Note also that we only need values for blocks representing material in stopes and caving polygons, so there may be fewer vertices defined for underground than for open-pit. Further details on calculation of open-pit and underground mine values are included in our earlier paper [29].
The vertices in $X^p$ and $X^u$ correspond to the same blocks in the block model’s three dimensional framework, but are distinct and separate vertices in the digraph with $X = X^p \cup X^u$.

### 2.2.2. Arcs

The arcs $A$ in $G$ are of three different types, $\beta$, $\gamma$ and $\delta$, where $A = \beta \cup \gamma \cup \delta$.

- $\beta$ is the set of arcs $a^\beta_k = (x^p_i, x^p_j)$, $a^\beta_k \in \beta$ with tails and heads both in $X^p$. Each arc represents a mining dependency to maintain maximum safe pit slopes (as in the normal pit optimisation model).

- $\gamma$ is the set of arcs $a^\gamma_k = (x^p_i, x^u_j)$, $a^\gamma_k \in \gamma$ with tails in $X^p$ and heads in $X^u$. These arcs bring the opportunity cost into the closure. Each arc connects vertex $x^p_i$ to vertex $x^u_j$, for every $x^u_j$ that has a weight defined (recall we only require underground values for blocks in stopes or caving polygons, and only the vertices corresponding to these will have weights defined). Observe that block $x^u_j$ has the same physical location as block $x^p_i$. The block $x^u_j$ is under the block $x^p_i$, with the vertical separation being equal to the minimum required thickness of the crown pillar as illustrated in Figure 3. The purpose of these arcs is as follows: if a block in $X^p$ is included in a closure, then the opportunity to mine another block (some levels down - with the number of levels being equivalent to the minimum required thickness of the crown pillar) by underground method will be lost.

Before describing the $\delta$ arcs, we present some definitions.
Definition 5 (Strongly connected subgraph (SCS)). A strongly connected subgraph (SCS) is a subgraph of a digraph in which, for every ordered pair of vertices \((x_i, x_j)\), there is a directed path from \(x_i\) to \(x_j\) and from \(x_j\) to \(x_i\).

Note that the path in Definition 5 need not be between distinct pairs and that a vertex is considered to have a path to itself.

Definition 6 (Trivial SCS). A trivial SCS is an SCS containing one vertex.

Definition 7 (Non-trivial SCS (NSCS)). A non-trivial SCS is an SCS containing more than one vertex.

\(\delta\) is the set of arcs \(a_\delta = (x_i^u, x_j^u), a_\delta \in \delta, x_i^u \neq x_j^u\), with tails and heads both in \(X^u\) forming NSCSs (and we observe that neither \(\beta\) nor \(\gamma\) arcs give rise to NSCS). Each NSCS represents the required shape for the ceiling of the underground mine (the base of the crown pillar) at a given depth as illustrated in Figure 4. In the example on the left, eight NSCSs prescribe a flat ceiling, with each NSCS corresponding to a different level. In the example on the right, the NSCSs corresponding to the three highest levels prescribe a dome shape (shown here in 2D) and at lower levels, a flat ceiling is prescribed. Each NSCS can be constructed by using arcs to form one or more overlapping directed cycles. Note that the \(\delta\) arcs combined with the \(\gamma\) arcs also influence the shape of the bottom of the open-pit mine, but do not fix the shape. There is often an operational requirement to have a minimum flat area at the bottom of a pit, and these NSCSs might contribute to the achievement of this requirement in some cases. However, full consideration of this requirement is beyond the scope of this model.

2.2.3. Problem Definition

The problem is to find a closure \(G_Y\) of \(G\) that maximises \(\mathcal{M}_Y\) (i.e. the maximum closure of this digraph):

\[
\mathcal{M}_Y = \sum_{i:x^p_i \in Y} m^p_i + \sum_{j:x^u_j \in Y} m^u_j
\]

(5)

We now show through derivation how maximisation of \(\mathcal{M}_Y\) is equivalent to maximisation of the total value of the open-pit and underground mines. The value of the open-pit mine is given by \(\sum_{i:x^p_i \in Y} m^p_i\) and the value of
The underground mine is $-\sum_{k: x_k^u \notin Y} m_k^u$. Let $\mathcal{M}_T$ denote the sum of these values.

$$\mathcal{M}_T = \sum_{i: x_i^p \in Y} m_i^p - \sum_{k: x_k^u \notin Y} m_k^u$$ (6)

The total value of material potentially available for underground mining is a constant $-\sum_{i: x_i^u \in X_u} m_i^u$. Let $\kappa = -\sum_{i: x_i^u \in X_u} m_i^u$. Some of the vertices are in the MGC ($x_j^u \in Y$) and the rest are not ($x_j^u \notin Y$) and Equations 7 and 8 follow from this observation.

$$\kappa = -\sum_{j: x_j^u \in Y} m_j^u - \sum_{k: x_k^u \notin Y} m_k^u$$ (7)

$$\Rightarrow \sum_{k: x_k^u \notin Y} m_k^u = -\kappa - \sum_{j: x_j^u \in Y} m_j^u$$ (8)

Substitute from (8) into (6).

$$\mathcal{M}_T = \sum_{i: x_i^p \in Y} m_i^p - (-\kappa - \sum_{j: x_j^u \in Y} m_j^u) = \sum_{i: x_i^p \in Y} m_i^p + \sum_{j: x_j^u \in Y} m_j^u + \kappa$$ (9)

Substitute from (2) into (9).

$$\mathcal{M}_T = \mathcal{M}_Y + \kappa$$ (10)

Since $\kappa$ is a constant we can conclude that maximising $\mathcal{M}_Y$ also maximises $\mathcal{M}_T$. 

Figure 4: The figures depict various NSCSs to model the ceiling of the underground mine (the base of the crown pillar).
Figure 5: Blocks in the optimal pit $x_p^o \in Y$ are shown on the left in green. Blocks in the crown pillar $x_u^c \in X^u \cap Y : x_p^o \notin Y$ are shown in red on the right and blocks available for underground mining $x_u^g \notin Y$ are shown in blue on the right. The green and red blocks on the right are not available for underground mining ($x_u^g \in Y$).

Figure 5 provides an illustration of the interpretation once the MGC is found. The details are as follows:

- $x_p^o \in Y$ represent blocks in the optimal open-pit.
- $x_u^g \in Y$ represent blocks not available for underground mining (including blocks in the crown pillar).
- $x_u^c \in X^u \cap Y : x_p^o \notin Y$ represent blocks in the crown pillar.
- $x_u^g \notin Y$ represent blocks available for underground mining.

3. Reduction of an Opportunities with Dependencies and Opportunity Costs (ODOC) Problem to an MGC Problem

We construct an optimisation problem involving multiple opportunities with dependencies, and with opportunity costs (ODOC) (Definition 8), which is a generalised form of the New Model (Section 2.2) not including the $\delta$ arcs, which are discussed later in Section 4. We then show that this problem can be reduced to an MGC problem (Definition 4).

Recall Definition 1 for Opportunity Cost and recall that in our case, the mutually exclusive alternatives with respect to any given block are to mine it by open-pit method, or by underground method. If we mine it by open-pit
method, we gain its open-pit value, but we lose the value that would have been gained if it had been mined by underground method.

Let $C$ be a set of opportunities. For each opportunity $c_i \in C$, let $v^{c_i} \in \mathbb{R}$ be the value and $v^{d_i} \in \mathbb{R}_{\geq 0}$ be the opportunity cost. If opportunity $c_i$ is selected, then the value $v^{c_i}$ will be gained but the value $v^{d_i}$ will be lost. Let $E$ be a set of dependencies and let $e_j = (c_i, c_k) \in E$ represent a dependency between $c_i$ and $c_k$ such that if $c_i$ is selected, then $c_k$ must be selected.

In our mining problem, an opportunity corresponds to a decision to include a given block in the open pit, and the opportunity cost corresponds to the resultant inability to include some block or blocks in the underground mine.

**Definition 8 (ODOC problem).** Find the set $C' \subseteq C$ such that $\sum_{i : c_i \in C'} (v^{c_i} - v^{d_i})$ is maximised, subject to: if $e_j = (c_i, c_k) \in E$ and if $c_i \in C'$ then $c_k \in C'$.

An example digraph model for an ODOC problem is shown in Figure 6. In this example, opportunities $c_1, c_2, c_3$ and their values $v^{c_1}, v^{c_2}, v^{c_3}$ are represented by vertices $x^{c_1}, x^{c_2}, x^{c_3}$ and weights $m^{c_1}, m^{c_2}, m^{c_3}$ respectively. Dependencies $(c_1, c_2), (c_1, c_3)$ and $(c_2, c_3)$ are represented by arcs $(x^{c_1}, x^{c_2}), (x^{c_1}, x^{c_3})$ and $(x^{c_2}, x^{c_3})$. The opportunity costs are represented by combinations of arcs and weighted vertices. For example arc $(x^{c_1}, x^{d_1})$ and vertex $x^{d_1}$ with weight $m^{d_1} = -v^{d_1}$ represent the opportunity cost $v^{d_1}$ for $c_1$.

**Theorem 2.** For any set of opportunities with values in $\mathbb{R}$, the ODOC Problem reduces to an MGC problem.

**Proof.** We first describe a transformation of the objects and values in the ODOC problem to vertices, vertex weights and arcs in a digraph. We then show that with this transformation, values accrue to the objective functions for ODOC and MGC in exactly the same way. Since both ODOC and MGC aim to maximise an objective function, we can conclude that for any given problem instance, a solution to the MGC problem will be identical to a solution to the ODOC problem.

Let $G$ be a digraph $G = (X, A, M_X)$ with weights for each vertex $x_i \in X$ given by $m_i \in M_X$. Let $X = X^c \cup X^d$, $M_X = M^c \cup M^d$ and let the ODOC problem be represented in the digraph as follows:

- For each $c_i \in C$ in the ODOC problem, there exists a vertex $x^{c_i} \in X^c$ with weight $m^{c_i} = v^{c_i}$, and a vertex $x^{d_i} \in X^d$ with weight $m^{d_i} = -v^{d_i}$. Note that since $v^{d_i} \geq 0$, it follows that $m^{d_i} \leq 0$. 

For each $i$ there exists an arc $(x^{c_i}, x^{d_i})$.

For each $e_j = (c_i, c_k) \in E$ in the ODOC problem there exists an arc $(x^{c_i}, x^{c_k})$.

Given the above representation, in order to show that the ODOC problem reduces to the MGC problem, it suffices to show that for a set $C' \subseteq C$ solving the ODOC problem, there exists a set $G_Y \subseteq G$ solving the MGC problem such that

$$\sum_{i:c_i \in C'} (v^{c_i} - v^{d_i}) = \sum_{i:x_i \in Y} m_i.$$

Recall that $X = X^c \cup X^d$ and $M = M^c \cup M^d$. It follows that $\sum_{i:x_i \in Y} m_i = \sum_{i:x^{c_i} \in Y} m^{c_i} + \sum_{i:x^{d_i} \in Y} m^{d_i}$. It is clear that the arcs $(x^{c_i}, x^{c_k})$ exactly model the dependencies $e_j = (c_i, c_k) \in E$ in the ODOC problem and so the value $v^{c_i}$ accruing to $\sum_{i:c_i \in C'} (v^{c_i} - v^{d_i})$ is equivalent to the value $m^{c_i}$ accruing to $\sum_{i:x^{c_i} \in Y} m^{c_i}$. We will now show that vertices $x^{c_i}$ and $x^{d_i}$ must either be both in the MGC or both not in the MGC. It then follows that the value $-v^{d_i}$ accruing to $\sum_{i:c_i \in C'} (v^{c_i} - v^{d_i})$ is equivalent to $m^{d_i}$ accruing to $\sum_{i:x^{d_i} \in Y} m^{d_i}$.

Any arc with its tail in a closure must also have its head in the closure. Hence, since $(x^{c_i}, x^{d_i}) \in A$, it follows that $x^{c_i}$ can only be in the MGC if $x^{d_i}$ is in the MGC. Vertex $x^{d_i}$ could be in a closure without $x^{c_i}$ but since $m^{d_i} \leq 0$, it cannot be in the MGC unless $m^{c_i} > -m^{d_i}$ in which case $x^{c_i}$ is also in the MGC.
4. Inclusion of Non-Trivial Strongly Connected Subgraphs

4.1. Introduction

Recall Definition 7 for non-trivial SCS (NSCS). In our earlier paper [29], we demonstrated with a realistic data set that a specific application of the LG Algorithm behaves well with respect to NSCSs, created by the $\delta$ arcs which are used to model well-formed crown pillars. Although our one demonstration is encouraging, it is not sufficient to give comfort that all solutions to the MGC problem will operate properly in all cases involving NSCSs.

Lerchs & Grossmann [4] and other authors applying maximum graph closure models in mine optimisation only contemplated the pit optimisation problem, not the transition problem or crown pillars and only described the use of arcs to model pit slope limits. Such arcs do not form NSCSs and the authors did not consider whether their method would be affected by NSCSs. Caccetta & Giannini [30], who restated the LG Algorithm in more formal terms, made no mention of NSCSs either. Furthermore, as discussed in the Introduction, other methods based on maximum flow algorithms have been developed and there is no way to guarantee that all implementations of LG and maximum flow will deal with NSCSs correctly. To avoid this risk, we have developed a procedure that collapses all NSCSs to representative vertices prior to solving the MGC problem, then expands representative vertices in the MGC to their original vertices. We will show that a digraph with NSCSs is equivalent to a digraph containing the aforementioned representative vertices. Furthermore, supposing some method for solving the MGC problem correctly deals with NSCSs, we will show that our procedure will generate identical results, with the added advantage of reducing the size of the MGC problem instance, and so promising to reduce computing time.

Consider the digraph $G = (X, A, M_X)$ and a closure $G_Y = (Y, A_Y, M_Y)$ such that the sum of the weights of the vertices in the closure $M_Y$ is maximised and such that for each closure $G_X$, where $G_X \subset G_Y$, $M_X < M_Y$ (i.e. the MGC as per Definition 4).

4.2. SCSs and their Properties in Closures

**Lemma 3.** Members of an SCS are either all in or all out of a closure.

*Proof.* Consider some vertex that is a member of an SCS. As a property of an SCS, the vertex must have a path to every other vertex in the SCS and every other vertex in the SCS must have a path to the vertex. It follows from the basic properties of a closure (Definition 2) that if one member of
an SCS is in a closure then all members of the SCS must be in the same closure.

4.3. Strongly Connected Components

**Definition 9 (Strongly connected component (SCC)).** A strongly connected component (SCC) is an SCS that is not contained in any other SCS.

We make the following observations:

- A trivial SCS (Definition 6) is an SCC provided it is not contained in any other SCS.
- Every vertex in a digraph must be in exactly one SCC.

Let \( \mathcal{S}_X \) be the set of SCCs in a digraph \( G = (X, A, M_X) \) such that \( \mathcal{S}_X \) contains all the elements of \( X \). Let \( S_i = (X^i, A^i, A^i_R, A^i_T, M^i_X) \in \mathcal{S}_X \) denote the \( i^{th} \) of \( r \) SCCs in \( G \) where:

- \( x_j \in X^i \) is a vertex member of the SCC.
- \( A^i \) is the set of arcs connecting vertices in \( X^i \) to each other.
- \( A^i_R \) is the set of arcs with tails outside \( X^i \) and heads in \( X^i \).
- \( A^i_T \) is the set of arcs with heads not in \( X^i \) and tails in \( X^i \).
- \( m_j \in M^i \) is the weight of vertex \( x_j \in X^i \).

It follows from the properties of an SCS that each vertex at the tail of an arc in \( A^i_R \) can reach all vertices in \( X^i \) and that all vertices at the heads of arcs in \( A^i_T \) can be reached from all vertices in \( X^i \). From Definition 5 (SCS) and Definition 9 (SCC), it is immediately apparent that there is an equivalence relation between any two vertices \( x_j \in X^i \) and \( x_k \in X^i \) (i.e. the relation is reflexive, symmetric and transitive) and that \( \mathcal{S}_X \) forms a partition of \( X \) (i.e. \( X = \bigcup_{i \in \{1,2,...,r\}} X^i \), where \( \bigcup \) signifies a disjoint union).

4.4. Functions

We define a function to collapse all SCCs in a digraph and the inverse of this function, to expand all SCCs in a digraph. We observe that in collapsing and expanding all SCCs, these function collapse and expand all NSCSs. We also reframe MGC as a function using the same notation.

Define \( \Gamma(X, A, M_X) = (X^*, A^*, M_X^*, \mathcal{S}_X) \) [Collapse SCCs] as follows:
1. Identify all $r$ SCCs $S_i = (X^i, A^i, A^i_R, A^i_T, M^i_X)$ and denote the set of sets $\mathcal{S}_X$.
2. Let $X^* = X$, $A^* = A$ and $M^*_X = \emptyset$.
3. For $i \in \{1, 2, \ldots, r\}$:
   (a) In $X^*$: replace all of the vertices in $X^* \cap X^i$ with a single vertex $x_0^i$.
   (b) In $M^*_X$: add $m^i_0 = \sum_{j \in M^i} m^j$.
   (c) In $A^*$:
      i. Remove arcs $A^* \cap A^i$.
      ii. Modify arcs in $A^i_T$ such that vertex $x_0^i$ is at the head of each arc.
      iii. Modify arcs in $A^i_T$ such that vertex $x_0^i$ is at the tail of each arc.
      iv. Replace each set of parallel arcs with a single arc.

Define $\Gamma^{-1}(X^*, A^*, M^*, \mathcal{S}_X) = (X, A, M_X)$ [Expand SCCs] as follows:
1. Let $X = \emptyset$, $A = \emptyset$ and $M_X = \emptyset$.
2. For $i \in \{1, 2, \ldots, r\}$, $i : S_i \in \mathcal{S}_X$:
   (a) In $X$: If $x_0^i \in X^*$ add vertices $X^i$.
   (b) In $M_X$: If $x_0^i \in X^*$ add weights $M^i_X$.
   (c) In $A$: Add any arcs in $A^i \cup A^i_R \cup A^i_S$ incident to any vertex in $X^i$.

Define $\Pi(X, A, M_X) = (Y, A_Y, M_Y)$ as finding the MGC $G_Y = (Y, A_Y, M_Y)$ of $G = (X, A, M_X)$.

Define $\Pi^*(X, A, M_X, \mathcal{S}_X) = (Y, A_Y, M_Y, \mathcal{S}_Y)$, differing from $\Pi$ only in the addition of $\mathcal{S}_X$ and $\mathcal{S}_Y$ to the domain and co-domain respectively, where $S_i \in \mathcal{S}_Y$, $i : X_i \subseteq Y$.

We make the following observations:
- $\Gamma^{-1} (\Gamma(X, A, M_X)) = (X, A, M_X)$.
- Functions $\Gamma$ and $\Gamma^{-1}$ map closures to closures (by the basic properties of a closure (Definition 2) and Lemma 3) and preserve the value of any closure.
- Since weights $m_j$ for vertices in $X$ each accrue to exactly one $m_0^i$ in $\Gamma$ it follows that $\sum_{j : x_j \in X} m_j = \sum_{i : x_0^i \in X^*} m_0^i$. 

20
4.5. Procedures

Define Procedure A as \( \Pi \) with notation as shown in Section 4.4.

Define Procedure B as a composition of three functions \( \Gamma^{-1}(\Pi^*(\Gamma(X, A, M_X))) = (X', A', M_{X'}) \), and with notation as follows:

1. \( \Gamma(X, A, M_X) = (X^*, A^*, M_{X^*}, \mathcal{I}_{X^*}) \)
2. \( \Pi^*(X^*, A^*, M_{X^*}, \mathcal{I}_{X^*}) = (Y^*, A_{Y^*}, M_{Y^*}, \mathcal{I}_{Y^*}) \)
3. \( \Gamma^{-1}(Y^*, A_{Y^*}, M_{Y^*}, \mathcal{I}_{Y^*}) = (X', A', M_{X'}) \)

We will show the equivalence of Procedures A and B as illustrated in Figure 7.

Observe that \( Y^* \) and \( X' \) are the vertex outputs of \( \Pi^* \) and \( \Gamma^{-1} \) respectively in Procedure B.

**Lemma 4.** The vertex output of \( \Pi^* \) will contain the representative vertex of an SCC, if and only if the vertex output of \( \Gamma^{-1} \) contains the vertices of the SCC. That is: \( x_i^0 \in Y^* \iff X_i \subseteq X' \).

**Proof.** The proof follows from the function \( \Gamma^{-1} \), in which we replace \( x_i^0 \) with the vertices \( X_i \).

Denote the elements of the domains for \( \Pi \) in Procedure A and \( \Pi^* \) in Procedure B as \( G \) and \( G^* \) respectively. That is, \( G = (X, A, M_X) \) and \( G^* = (X^*, A^*, M_{X^*}, \mathcal{I}_{X^*}) \).

**Lemma 5.** The MGC from \( \Pi \) in Procedure A will contain the vertices of an SCC if and only if the MGC from \( \Pi^* \) in Procedure B contains the representative vertex of the SCC. That is, \( X^i \subseteq Y \iff x_i^0 \in Y^* \).
Proof. Through the operation of $\Gamma$, collectively the vertices in $X^i \in X$ have the same weight and the same dependencies as $x_0^i \in X^*$. The effect of including $X^i$ in a closure of $G$ is identical to the effect of including $x_0^i$ in a closure of $G^*$. It follows that for any closure of $G$ containing some selection of $X^i$’s there is a closure of $G^*$ containing the representative vertices $x_0^i$ for the same selection, such that the sum of the weights of the vertices in the first closure is equal to the sum of the weights of the vertices in the second closure.

We observe that $Y$ and $Y^*$ are the vertices of the MGCs of $G$ and $G^*$ respectively and by Theorem 1, each is unique.

Consider these two cases, one of which must be true if Lemma 5 is false:

- Suppose $X^i \subseteq Y$ and $x_0^i \notin Y^*$: Then there is an MGC on $G^*$ that does not contain $x_0^i$ and a closure on $G^*$ that does contain $x_0^i$, but which must be an MGC because it is induced by the MGC on $G$ that contains $X^i$.

- Suppose $X^i \not\subseteq Y$ and $x_0^i \in Y^*$: Then there is an MGC on $G^*$ that contains $x_0^i$ and a closure on $G^*$ that does not contain $x_0^i$, but which must be an MGC because it is induced by the MGC on $G$ that does not contain $X^i$.

Both of the above cases lead to contradictions of Theorem 1.

\[\square\]

**Theorem 6.** A maximum graph closure (MGC) induces an MGC in which SCCs have been collapsed to representative vertices. Furthermore, when the representative vertices in the MGC are expanded to their original vertices, this set of vertices equals the set of vertices in the first-mentioned MGC. That is, $\Gamma^{-1}(\Pi^*(\Gamma(X,A,M_X))) = \Pi(X,A,M_X)$.

Proof. It is sufficient to show that the vertices in $Y$ (the vertex output of Procedure A) are the same as the vertices in $X'$ (the vertex output of $\Gamma^{-1}$ in Procedure B). In other words, $x_j^i \in Y \iff x_j^i \in X'$, which follows immediately as a consequence of Lemmas 4 and 5:

\[x_j^i \in Y \iff x_0^i \in Y^* \iff x_j^i \in X'\] (11)

\[\square\]
The implication of Theorem 6 is that we can use Procedure B to avoid the risk of NSCSs being dealt with incorrectly when solving the MGC problem.

4.6. Computational Complexity

Procedure A (Π) solves the MGC problem. Caccetta & Giannini [30] state that MGC can be solved in $O(n^2 \log n)$ time, where $n$ is the number of vertices. For Procedure B:

1. Γ: Recall that NSCSs are created with the $\delta$ arcs in the new optimisation model (Section 2.2), so we know in advance which NSCS each vertex belongs to. Accordingly, Γ can be completed in $O(n)$ time.

2. Π*: This can be solved in the same time as Π in Procedure A, however the input size $m$ may be smaller. That is, it can be completed in $O(m^2 \log m)$, where $m \leq n$ and $n - m$ is the reduction in the number of vertices as a result of Γ.

3. Γ−1: Following similar reasoning to that given above, Γ−1 can be completed in $O(n)$ time.

Overall the computational complexity of Procedure B is $O(n + m^2 \log m)$, which compares favourably with Procedure A:

- If $n < m^2 \log m$, then Procedure B has the same or less complexity than Procedure A. That is, $O(n + m^2 \log m) = O(m^2 \log m) \leq O(n^2 \log n)$.
- If $n \geq m^2 \log m$ then Procedure B is less complex than Procedure A. That is, $O(n + m^2 \log m) = O(n) < O(n^2 \log n)$.

Supposing that there is a method to solve MGC that deals properly with NSCSs and it is used in both procedures:

- If there are no NSCSs in the digraph $G$, Procedure B will take more processing time than Procedure A. This is because Π* in Procedure B will take the same time as Procedure A, and Γ and Γ−1 in Procedure B will take additional time.

- As the number of NSCSs increases, the time taken by Procedure B will decrease relative to the time taken by Procedure A. For a large number of NSCSs, we expect Procedure B will take considerably less time than Procedure A.
5. Experimental Results

In Section 2 we illustrated our new method with small, simple two-dimensional examples. In this section, we demonstrate that our new method works with a large, complex, three-dimensional model.

To obtain a suitable model set, we started with a well-known block model Marvin, constructed in the 1990s by Australian geologist Norm Hanson. It represents a 20 mT copper and gold deposit, with sulphide and oxide ores. The model was initially designed to test open pit optimisation software and is still used for that purpose [31]. We needed a much deeper model than the original Marvin, so we duplicated the lowest bench to make 17 lower benches. We then developed an underground version of the model as well as various sets of arcs. Our full set of models and arcs:

- An open-pit model \((X^p)\) with 124,440 blocks. This is the vertically extended Marvin model.
- An underground model \((X^u)\) with 124,440 blocks. As above, but each block is assigned a negative underground value representing the opportunity cost.
- A set of 6,733,168 \(\beta\) arcs to model pit slopes in \(X^p\).
- For a zero thickness crown pillar: A set of 13,534 \(\gamma\) arcs to model the dependencies between blocks in \(X^p\) and \(X^u\).
- For a 120 metre thick crown pillar: A second set of 13,534 \(\gamma\) arcs to model the dependencies between blocks in \(X^p\) and \(X^u\).
- For a well-formed crown pillar: A set of 12,075 \(\delta\) arcs to form NSCSs in \(X^u\). Note that the pit optimiser that we used (see below) behaved well with respect to NSCSs so we did not need to run a function to collapse them (see Section 4.4).

In an earlier paper [29] we developed five cases and optimised them using a widely used commercial pit optimisation system. We summarise the results of those five cases below, and then describe a new sixth case:

- Case 1 (Normal pit optimisation): Using \(X^p\) blocks and \(\beta\) arcs.
- Case 2 (Opportunity cost): We used the software’s inbuilt facilities to effect the opportunity cost approach (the method described by J Whittle [17] and Camus [18]) with \(X^p\) blocks and \(\beta\) arcs. As expected the
open pit was smaller than for case 1 due to the effect of the opportunity cost, but the combined value of the open pit and underground mine was greater than the value for case 1. We observed that there were some isolated underground mine blocks beside the open pit. They contributed to the value calculation, but in practice these isolated blocks would not be economic to mine.

- Case 3 (Opportunity cost with new model): We used $X^p$ and $X^u$ blocks, and $\beta$ and $\gamma$ arcs for a zero thickness crown pillar. The result was identical to that for case 2 as expected.

- Case 4 (120 metre thick crown pillar): As for case 3, but with $\gamma$ arcs modelling a 120 metre thick crown pillar. There was a reduction in value compared to case 3. This result was expected, since we had added further constraints to the optimisation.

- Case 5 (Well-formed 120 metre thick crown pillar): As for case 4, but with the addition of $\delta$ arcs to model well-formed crown pillars. The value in this case was lower than for case 4, since we had added a new constraint. The pit depth increased, which was unexpected, but turned out to be correct for this data set.

The processing time for our new method to model a well-formed crown pillar (case 5) was 14 seconds, compared to 8 seconds for case 1. The extra 6 seconds is immaterial. Recall from the Introduction section, these optimisation runs are conducted many times, but in the context of a strategic planning process that may take months or years. Recall also from the Introduction section that Chung et al. [24] solved a similar (but not identical) problem to our case 5 using an integer programming approach: Their 83,000 block model took 34.5 hours to process.

Since the publication of our earlier paper [29] we have developed a sixth case. The purpose of this sixth case is to provide a reasonable benchmark against which to compare the dollar value of the plan produced by our new method (case 5). Case 6 is based on case 3 and on a hypothesis as to how a mine engineer might manually adjust it to add a well-formed crown pillar. A mine engineer would also remove the aforementioned isolated underground blocks, since they are not economic to mine.

Removal of the isolated underground blocks is straightforward. As to the addition of the crown pillar, there are several options, including the removal of blocks from the underground mine, or the open pit mine, or both. We chose the removal of blocks from the underground mine for several
reasons. Firstly, removal of blocks from a mine reduces its value, and on the
assumption that the open pit precedes the underground mine, removal of
blocks from the latter is preferred. Secondly, this approach is the only one
we found explicitly described in the literature [25]. Finally, it is the simplest
and most easily reproduced method. The result was a value of $1,617m
for case 6, as compared to $1,858m for case 5. This means that our new
method (case 5) provides a $241m or 15% increase in value compared to the
benchmark (case 6). This improvement in value is an example of the efficacy
of our new method. The general case for the efficacy of our new method is
provided by its mathematical integrity, the main topic of this paper.

6. Discussion

In the Introduction and New Model sections, we state that we require
an underground mine plan as input (specifically underground mining values
for blocks) and that we are generally unconcerned with the details of this
plan. However, when we optimise the transition depth, we assign blocks to
the open-pit mine that may have been included in the initial underground
mine plan. This reduction in blocks in the underground mine may or may
not materially change key assumptions. For example, suppose the initial
underground mine plan was made on the assumption that there would be 5
million tonnes of ore and then the transition optimisation reassigns 0.5 mT
of this to the open pit. Then a mine engineer may determine that the unit
costs of mining (including development) used in the initial underground mine
plan still hold, and we can accept the result. Now suppose the transition
optimisation cuts the underground mine down to 2mT of the deepest ore.
This is a substantially different underground mine, with more development
required to reach the first ore, and much less ore across which to spread
development costs. In this case, a mine engineer may determine that unit
costs for the underground mine must be increased and a new underground
plan developed.

In the Introduction we mentioned the work of King, Goycoolea & New-
man [28]. They optimise the open-pit and underground mining schedules,
the placement of the crown pillar and also the placement of sill pillars (recall
a sill pillar is material left in-situ in a particular type of underground mine
to allow for a change in mining direction). Their method cannot handle very
large models but they overcome that by producing a small number of open-
pit scheduling blocks. Our method can handle very large models, but does
not optimise directly for NPV. We believe there is scope for the combination
of our method and King et al.’s. Firstly, use our method as part of the generation of the open-pit scheduling blocks. This could be done by running our method several times with different open-pit prices to generate a family of pits, and by using that family of pits to produce open-pit scheduling blocks, with methods well-known in the mining industry. These open-pit scheduling blocks would have shapes that respond to the opportunity costs. Secondly, modify the King et al. method slightly to only require a crown pillar above the underground mine. Using this combination, we believe mine planners could access the full benefits of both of these very different approaches.

7. Conclusions

In this paper we establish the mathematical integrity of a new method to solve the transition problem. The new method includes three modifications to the normal model used for pit optimisation (solving for the maximum graph closure problem). The modifications allow the algorithm to take account of the underground mining value of a block; to take account of the requirement for a crown pillar with a specified thickness; and to impose a shape to the crown pillar above the underground mine.

The modifications include:

- The use of an additional set of vertices \( X^u \), with each additional vertex representing the opportunity cost of mining a block by open-pit method (the opportunity cost being the lost opportunity to mine a block by underground method).

- The use of an additional set of \( \gamma \) arcs to connect vertices representing open-pit values to vertices representing the above-mentioned opportunity costs.

- The use of a further set of \( \delta \) arcs to create NSCSs, each of which defines a required shape for the crown pillar above the underground mine at a given level.

In Theorem 2 we prove that the general form of our new model (not including the NSCSs) can be reduced to an MGC problem. In Theorem 6 we prove that NSCSs can be removed from the MGC problem, by collapsing each SCC down to a single representative vertex, reducing the size of the input to the MGC problem and obviating the need to prove that any given approach to solving the MGC problem can correctly deal with NSCSs.
We finish with some comments on the commercial relevance of this work. Mining companies can improve the value of their combined open pit and underground mines by solving the transition problem. Whilst there are methods in the literature to solve the transition problem directly for maximum NPV, they generally suffer from an inability to use commercial-scale models and they treat the crown pillar in a very simplistic way. Of all the methods for solving the transition problem, only the cash-maximising opportunity cost approach is currently widely available to mine planners through regular pit optimisation software. Our new method allows mine planners to extend the opportunity cost model to include detailed geometric and economic consideration of crown pillars with specific shapes. In our test case we achieved a 15% increase in cash value.

Acknowledgement

David Whittle is the grateful recipient of the 2015 Gilbert Rigg Scholarship, which supports his graduate research in underground mine plan optimisation.

We wish to thank Dassault Systèmes for providing the software used in Section 5.

We also wish to thank our four reviewers, whose thoughtful and detailed comments enabled us to make significant improvements to this paper.

References


