Computers and Operations Research 115 (2020) 104739

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Optimum ramp design in open pit mines

Juan L. Yarmuch[a,1,*], Marcus Brazil[b], Hyam Rubinstein[c], Doreen A. Thomas[a]

[a] Department of Mechanical Engineering, The University of Melbourne, Victoria 3010 Australia
[b] Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010 Australia
[c] Department of Mathematics and Statistics, The University of Melbourne, Victoria 3010 Australia

ARTICLE INFO

Article history:
Received 5 April 2017
Revised 8 April 2019
Accepted 24 June 2019
Available online 25 June 2019

Keywords:
Integer programming
Surface mining
Road design
Mine planning

ABSTRACT

The problem studied in this paper is that of designing the optimal open pit haulage ramp that, for a given ramp width and a maximum ramp gradient, connects two points of the mine, minimising construction and operational costs. Because in-pit ramps require the removal of a considerable amount of non-valuable material (stripping), we discuss two different problems: high stripping (or in-pit) ramp design and low stripping (or ex-pit) road design. For the first situation, we present an integer programming model; in the second case, a shortest path approach is undertaken. In both cases, the models can be solved exactly, and include gradient and curvature constraints. The proposed formulations have been tested on real mine data, showing a significant reduction in cost compared to the previous mine design.

1. Introduction

In broad terms, open pit mines are excavations created to extract valuable materials (ore) located below the surface. Ore is usually accompanied by a considerable amount of material with little or no value (waste) which must be extracted before reaching the ore; this process is known as stripping. During the life of the mine the ore is hauled to processing plants or stockpiles, while waste material is transported to waste dumps. Thus an open pit mine requires a road network which connects the extraction sectors inside the pit to all possible destinations outside the pit (Yarmuch et al., 2017). In-pit ramps are roads that connect the working faces to the pit exits; these require the removal of a great amount of material for their construction. The total tonnage of ore and waste and the pit shape can change dramatically after the addition of the in-pit ramps. Ex-pit roads, on the other hand, connect pit exits to processing plants or dumps, and their construction usually requires removing much less material than in-pit ramps.

The ramp design process begins with the determining the pit exits, the ramp width and gradient. Pit exits are usually located close to the processing plants or waste dumps to reduce the total haulage cost, while, the ramp width is defined by considering operational factors, safety conditions and the stripping cost. Examples of operational factors are the size of the haulage equipment, the number of traffic lanes, the traffic density and safety elements (centre line piles, road edge barriers and ditches). A comprehensive description of these factors can be found in Atkinson (1992). As a rule of thumb, the minimum ramp width should be at least 4 times the width of the hauling equipment. The ramp gradient is constrained by a maximum gradient determined by diverse factors such as the type of hauling trucks, the mining operation conditions and the road surface quality. Mining trucks work best when the maximum ramp gradient is between 8% and 10% (Thompson, 2011).

Once the design parameters have been defined, engineers must decide whether to use spiral ramps, switchbacks (180° turns) or a combination of both. Because ramps can be used as catch berms to control local rock failures, a design that includes switchbacks is generally more stable, in geotechnical terms, than spiral ramps. However, a significant number of switchbacks over the same wall increases the amount of stripping of the pit, compromising the profitability of the mine. Also, switchbacks affect the cost and safety of the haulage truck's operation (Thompson, 2011). For instance, switchbacks reduce the truck speed and consequently the truck productivity, they require periodic maintenance increasing the operational cost, and they generate visual problems for drivers increasing the risk of collision.

The problem considered in this paper is the design of the optimal open pit haulage ramp, for a given ramp width and a maximum ramp gradient, which connects two points of the pit, minimising construction and operational costs. Since the stripping cost associated with in-pit ramps can be considerably larger than that of ex-pit roads, we discuss the two different problems: high stripping ramp design and low stripping road design. The first is commonly associated with in-pit ramps, whereas the second is more relevant to ex-pit roads.

* Corresponding author.
E-mail address: jyarmuch@ing.uchile.cl (J.L. Yarmuch).
[1] Now at Department of Mining Engineering, University of Chile, Santiago, Chile

https://doi.org/10.1016/j.cor.2019.06.013
0305-0548/

This paper is organised as follows. In Section 2 we discuss the literature related to the ramp design problem in open pit mines. We also discuss the contribution of this paper. In Section 3 we present a shortest path formulation to solve the low stripping road problem. In Section 4, we give a binary linear programming model for the high stripping ramp design problem. In Section 5 we introduce a heuristic approach to solve the high stripping ramp design problem. In Section 6 we present a case study for both formulations, and in Section 7 present the conclusions.

2. Literature survey and contribution

In this section we review the state of the art of the open pit ramp design problem. Although there are many articles about the open pit ramp design problem, the majority of them focus on practical guidelines instead of mathematical formulations or algorithms. We are aware of only a few publications that analyse the in-pit ramp design problem, and none on the ex-pit road design problem. For that reason, we survey some useful literature on road design in forest operations that can be applied to the ex-pit (low stripping) road design problem. Interested readers can find a more complete review of optimisation techniques in forest road network problems in Akay et al. (2013).

2.1. Low stripping roads

If the stripping associated with the construction of a road is not significant, the problem can be reduced to the problem of finding the minimum cost curvature-constrained path between two directed points. Different approaches considering continuous or discrete optimisation have been studied. For instance, Chang et al. (2012) introduces a continuous space formulation to find a cost curvature-constrained path in a plane where the cost function is anisotropic. However, this technique is only valid when the construction cost is independent of the geographical location. More recently, Casal et al. (2017) have presented a sequential quadratic programming method for the optimisation of horizontal alignment roads (which are composed of circular curves and tangent line segments joined by means of transition curves). This model can be used in road reconstruction projects. Also, Mondal et al. (2015) develops a bi-level programming model for the optimisation of the horizontal alignment in a specified corridor. The proposed method obtains locally optimal horizontal alignment and a vertical alignment that is globally optimal.

On the other hand, discrete optimisation techniques have been widely used to design roads in forest operations. One of the most interesting approach consists in dividing the terrain into cells and then associating with each cell a certain set of nodes to represent the directions in which the cell can be accessed. Thus, the turning radius constraint can be mathematically modelled and discrete optimisation techniques can be applied (for example, see Anderson and Nelson, 2004; Epstein et al., 2001; Liu and Sessions, 1993; Meignan et al., 2012; Najafi and Richards, 2013; Pushak et al., 2016).

Current digital elevation models allow the discretisation of the terrain in cells of size of 10 by 10 m, and the number of cells selected as neighbours to each cell is 8 or 24. A smaller cell size and a higher number of neighbours increases the resolution of the model, but makes it harder to solve (Heinimann et al., 2003) and Stückelberger (2008). Because roads in forest operations are types of low stripping roads, this approach can be implemented to find the minimum cost ex-pit roads.

Interestingly, Burdett et al. (2015) uses three-dimensional blocks to compute a more accurate cost (compared to the method of estimating the cost by the interpolation of 2D vertical sections) for the Earth Allocation Problem. The partition of the problem domain into 3D regular blocks, known as a block model, is often used in mining optimisation problems (see Newman et al., 2010) and, indeed, is a standard tool in mine planning. This approach is relevant to us since we use a block model structure as the basis for the integer programming model introduced in Section 4.

2.2. High stripping ramps

A complete description of the in-pit (high stripping) ramp design problem can be found in Couzens (1979). In addition, Kaufman and Ault (1977), Atkinson (1992) and Tannant and Regensburg (2001) provide a good explanation of hauling operational issues and ramp design guidelines. Ramp parameters, such as width, super-elevation, gradient and construction materials, and operational considerations, such as safe distances between trucks and the use of traffic signals, are explained well in those articles.

To our knowledge, there are only two papers in the literature that discuss algorithms to solve the in-pit ramp design problem. Dowd and Onur (1992) presents a simple algorithm that starts from a specific block at the bottom of the pit and then generates one clockwise ramp and one anti-clockwise ramp from bottom to top. After these ramps are generated, the algorithm iterates to another block in the bottom of the pit until no more blocks can be evaluated. Finally, the algorithm selects the ramp with minimum loss in profit. Depending on the topography surrounding the pit, clockwise and anti-clockwise ramps can differ greatly in the amount of stripping and the pit exit location. However, the algorithm does not consider the addition of switchbacks to the ramp or place a constraint on the pit exit, creating purely spiral maximum gradient roads.

Gill (1999), on the other hand, presents a dynamic programming algorithm to minimise the cost of the ramp construction for a given open pit. Using a graph to represent the pit contour, the algorithm is capable of incorporating gradient constraints as well as minimising the stripping cost. Gill claims that his algorithm can find the optimum ramp under some assumptions, but no details are provided due to commercial confidentiality.

2.3. The open pit ramp design problem

The problem studied in this article is to find a ramp design that minimises the associated construction, haulage and maintenance costs when ramp width and maximum gradient are given. As noted in the introduction, the problem has been divided into two independent sub-problems, the high stripping or in-pit ramp design problem and the low stripping or ex-pit road design problem.

In-pit ramps must connect the bottom of the pit to a given pit exit at the top of the excavation (see Fig. 1). The model should account for the fact that external conditions, such as rock quality, could disqualify some sectors of the pit for ramp construction and that material above the ramp must be extracted so as to satisfy the maximum pit-wall slope conditions. Decision variables of the problem are the ramp starting and final points (from a given set of candidate points), the direction and slope of each section of the ramp, and the number and locations of switchbacks.

The ex-pit road design problem consists of finding the path that connects the pit exit to the mine facilities such as processing plants, stockpiles, dumps and maintenance bays. For this problem, the starting and ending points are given, as well as any forbidden areas. Because the stripping associated with the ex-pit roads is usually not significant, the topography can be considered invariant. Thus, for simplicity, the stripping associated to each segment of the road can be considered as a pre-calculated cost. Even though a change in direction in the path does not have a significant impact

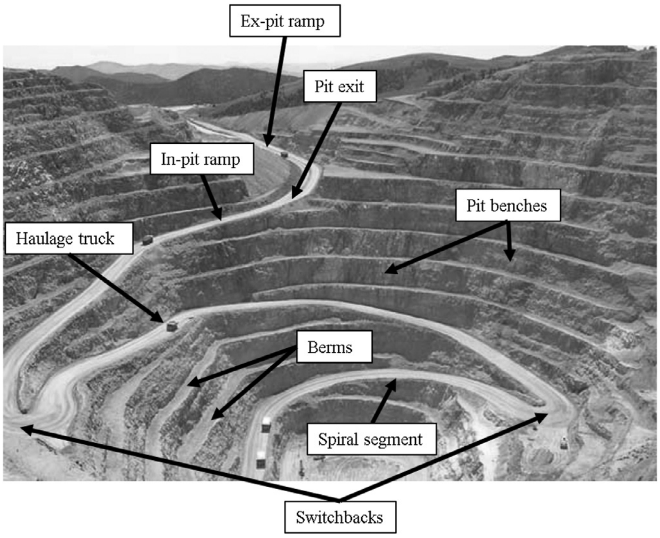

Fig. 1. Open pit ramp.

in stripping cost, it does increase the road construction and maintenance cost. Therefore, the changes of direction along the road should be taken into account in this problem.

Our first contribution is to adapt the ex-pit road design problem as a minimum distance problem that can be efficiently solved. The second contribution is to develop a new mathematical model to solve the in-pit ramp design problem. In order to be able to solve realistically sized problems within a reasonable time scale, we also develop an heuristic approach to solve the high stripping ramp design problem. Additionally, we present computational results based on real mine data.

3. Optimisation of low stripping roads

Techniques used in the forest industry can be applied to define the optimum ramp in open pit mines. One of the problems that needs to be solved in the planning stage for the forest problem is to determine the layout of the roads that connect the forested (production) areas to the timber mills (processing plants). Road systems of several kilometres in length and at least 10m wide are designed as part of the planning stage. Similarly, the ex-pit haulage roads in the open pit mining problem need to connect the pit exit to the processing plant. The layout selected will define the amount of stripping and the haulage cost, which is proportional to the ex-pit road length.

3.1. Shortest path formulation

The problem of finding the minimum path over a weighted graph has been widely studied and efficient algorithms have been developed to find the optimum in polynomial time. According to Bertsimas and Tsitsiklis (1997), Dijkstra's algorithm is the most efficient for solving the shortest path problem in dense graphs. However, different implementations of the Dijkstra algorithm improve theoretical computational bounds or empirical running times for special classes of graphs (see Ahuja et al., 1993 for further details).

As mentioned in Section 2, low stripping roads can be approximated as a road location problem. First, the topography must be projected onto a 2D horizontal plane and divided in regular cells. For each cell we associate a group of nodes that represent the directions in which that cell can be entered and left. The nodes of each cell are then connected to the nodes in the neighbourhood cells, creating a graph that represents possible paths on the topography. Thus, costs for changes in direction and gradient constraints,

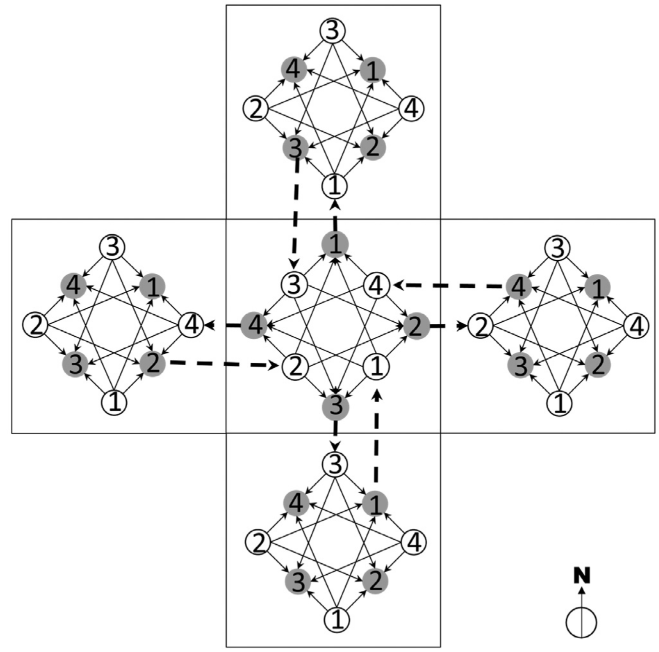

Fig. 2. Representation of a simple 4 choice of directions graph $\mathcal{G}$. 1, 2, 3 and 4 represent North, East, South and West directions, respectively. Vertices in white represent incoming directions, while vertices in grey colour outgoing directions. Solid arcs connect all possible incoming directions to all possible outgoing directions at the same cell. Dashed arcs represent the feasible connections from a cell to a neighbouring cell.

as well as operational costs depending on the geographical location or characteristics of the ramp can be incorporated into the problem as weights on the arcs between nodes. Note that the representation of directions as nodes in which a cell can be accessed was briefly introduced by Epstein et al. (2001) and more described in Epstein et al. (2007). The following section explains in detail the construction of the graph for the ex-pit road case.

3.1.1. Construction of the graph

First, it is necessary to subdivide the given topography, where the road will be defined, into a set of cells ($\mathcal{C}$). Our approach is to use a regular 2D lattice of cells, where each cell, $c \in \mathcal{C}$, represents a rectangular area of the terrain. For each cell c, we define the set of neighbours cells $\hat{c} \in \hat{\mathcal{C}}_c$ to generate the number of directions in which the cells will be accessed. For example, in Fig. 2, the 4 neighbours of the central cell correspond to 4 rectilinear directions. More generally, neighbourhoods of 8 or 24 cells correspond to more directions.

The vertices $\mathcal{V}$ of the graph $\mathcal{G}(\mathcal{V}, \mathcal{A})$ represent the possible incoming and outgoing directions of the ramp at each cell. Let $\mathcal{D}'$ and $\mathcal{D}''$ represent a discretised set of incoming and outgoing directions respectively. Then, the vertex of an outgoing direction $d'' \in \mathcal{D}''$ at cell c is represented as $v_{d'',c}$. Similarly, $v_{d',\hat{c}}$ is the vertex of an incoming direction $d' \in \mathcal{D}'$ at cell $\hat{c}$. Therefore, an arc denoted as $(v_{d'',c}, v_{d',\hat{c}})$ represents the connection of the outgoing direction d'' at cell c to the incoming direction d' at cell $\hat{c}$. In Fig. 2, vertices of incoming directions are illustrated as white nodes, while the vertices of outgoing directions are represented as grey nodes.

Two types of connections are needed to build the graph. To clarify the construction of the graph, we define the disjoint sets of arcs $\mathcal{A}_d$ and $\mathcal{A}_c$, where $\mathcal{A} = \mathcal{A}_d \cup \mathcal{A}_c$. The set $\mathcal{A}_d$ connects all possible incoming directions to all possible outgoing directions at the same cell (solid arcs inside cells in Fig. 2). Thus, $\mathcal{A}_d = \{(v_{d',c}, v_{d'',c}), \quad d' \in \mathcal{D}', \quad d'' \in \mathcal{D}'', \quad c \in \mathcal{C}\}$. The set $\mathcal{A}_c$ represents the set of permitted connections from a cell to a neighbour-

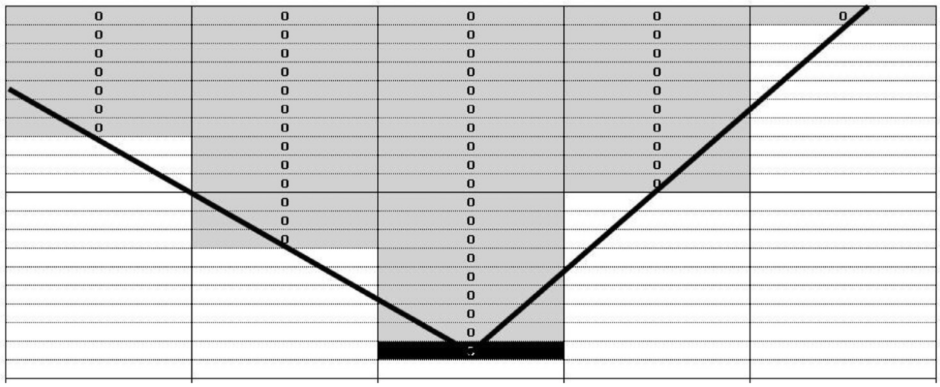

Fig. 3. Illustration of the wall slope precedences. Blocks in grey need to be mined before the block in black to ensure slope constraint represented by the two inclined lines that start at the black block.

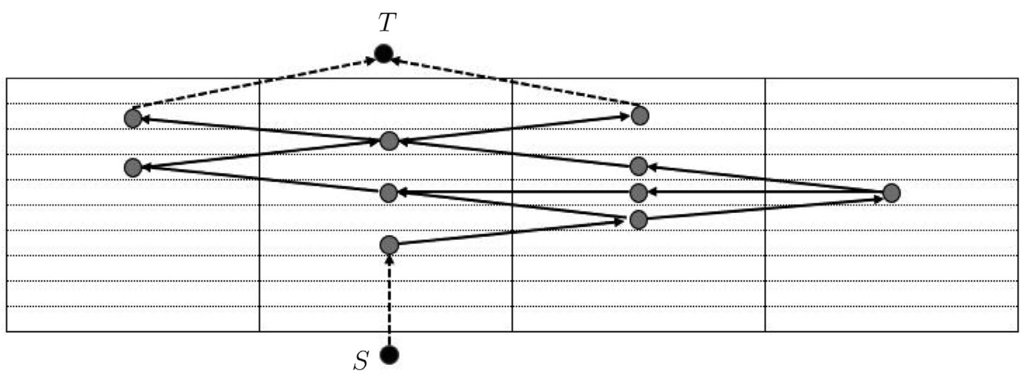

Fig. 4. Example of a simple 2D case of $\mathcal{G}_3$. The vertices in grey illustrate the blocks that can be selected as ramp, the edges in solid black represent the possible connections of the path. The initial (S) and final (T) vertices are represented in black and the edges from and to those vertices are illustrated with dashed lines.

ing cell (dashed arcs in Fig. 2). Formally, $\mathcal{A}_c = \{(v_{d'',c}, v_{d',\hat{c}}), \quad \hat{c} \in \hat{\mathcal{C}}_c, \quad c \in \mathcal{C}\}$.

Curvature constraints can be modelled within this graph definition. Changes of direction constraints can be included by weighting or cutting off some arcs in $\mathcal{A}_d$. Likewise, topographical (or gradient) constraints can be incorporated by adding weights to arcs in $\mathcal{A}_c$.

The weights associated with the elements of $\mathcal{A}_d$ are calculated as the costs of changing the direction d' to d'' at each cell c. These costs may vary depending on the angle of the turn, the super-elevation and the maintenance costs. Likewise, weights related to arcs in $\mathcal{A}_c$ are designed to reflect the cost of building a section of road from c to $\hat{c}$; usually this cost is proportional to the Euclidean distance that separates the nodes, adjusted to account for any difference in height.

Once the weights have been calculated and the start and end cells defined, the problem of finding a minimum cost path can be solved using Dijkstra's algorithm for the weighted graph $\mathcal{G}$ (or any other shortest path algorithm that suits graph $\mathcal{G}$). This approach does not take into account a physical modification in topography as we are dealing with the low stripping case. Note that the graph $\mathcal{G}$ can be also used to solve the problem of finding the minimum cost network when more than two terminals need to be connected through a road network. In this case the problem reduces to the Steiner Tree problem in a graph (Anderson and Nelson, 2004), a well-studied problem, for which good exact and heuristic algorithms exist.

4. Optimisation of high stripping ramps

In this section, we present a binary linear programming (BLP) formulation to find a minimum cost ramp with vertical and horizontal alignment constraints for a given pit, a given set of initial and final points, and a given ramp width. This formulation takes into account the stripping associated with the ramp excavation and the construction and haulage cost.

4.1. Minimum cost ramp formulation

Here a binary linear model to find the minimum cost ramp is formulated. The inputs for the model are divided into two types: the geological inputs, which will include the geological block model, the topography, the contour of the pit, and the dependencies between blocks based on the required pit slopes; and operational inputs, which include various costs, the ramp width, and the set of possible starting and final points for the ramp. The geological model is assumed to be represented as a regularly spaced set of blocks (equal dimensions in coordinates x and y), and must include the contour of the pit shape we are aiming to achieve. To reduce the size of the problem, blocks that are too far from the contour of the pit can be discarded. We assume the ramp width is equal to the block size in x (dx) or y (dy) and the dimension in z (dz) represents the maximum ramp gradient (max. slope $\geq \frac{dz}{dx}$ or $\frac{dz}{dy}$). For a standard block model this may necessitate first dividing every block into several horizontal slices (dotted lines in Figs. 3 and 4).

To consider the stripping associated with the ramp, we define a set of vertical precedences (upward), or blocks above the ramp, that need to be extracted to ensure a minimum wall slope (see Fig. 3). Similarly, to avoid the construction of the ramp above excavated blocks, we consider a set of vertical precedences (downward) to ensure that no block corresponding to the floor of a ramp is extracted.

Formally, the precedence relationships are represented in terms of the digraph $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{A}_1)$ where $\mathcal{V}_1$ is the set of blocks and

$(u, v) \in \mathcal{A}_1$ only if the block v must be extracted before the block u. We assume that digraph $\mathcal{A}_1$ contains only immediate precedence relationships. That is, if u, v and $w \in \mathcal{V}_1$ are such that $(u, v) \in \mathcal{A}_1$ and $(v, w) \in \mathcal{A}_1$, then $(u, w) \notin \mathcal{A}_1$. Similarly, the condition that the ramp cannot be built above excavated blocks is represented by a digraph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{A}_2)$ where $(u, v) \in \mathcal{A}_2$ represents the constraint that if the block u is selected as ramp, then the block v must not be extracted.

To build $\mathcal{G}_1$ we add a vertex to $\mathcal{V}_1$ for each block in the sliced block model, and we add an arc (u, v) to $\mathcal{A}_1$ from each vertex $u \in \mathcal{V}_1$ to all vertices $v \in \mathcal{V}_1$ such that the minimum wall slope is ensured for the block that is represented by vertex u. This is a standard method in pit design problems as it can be seen in Newman et al. (2010). The graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{A}_2)$ has a similar structure to $\mathcal{G}_1$, but in this case for each vertex u there is a unique arc (u, v) in $\mathcal{A}_2$ such that the block represented by v is immediately below the block represented by u.

The ramp is represented as a connected path of adjacent blocks from S to T (two artificial blocks that represent the starting and ending points of the ramp). Formally, the possible adjacencies between blocks is represented in terms of the digraph $\mathcal{G}_3 = (\mathcal{V}_3, \mathcal{A}_3)$, where there is one vertex $v \in \mathcal{V}_3$ for each block that can be selected as part of a ramp, plus the initial vertex S and the final vertex T. The set of arcs $(u, v) \in \mathcal{A}_3$ consists of all pairs of vertices that can be selected as adjacent blocks in any feasible ramp. Additionally, we add one edge from S to every node that can be selected to begin the ramp, and one edge per vertex that can end the ramp to T. Fig. 4 illustrates an example of the graph of connections for a simplified case.

We now present a formal integer programming model for finding the minimum cost in-pit ramp.

4.1.1. Indices, sets and parameters

In addition to the block model and sets $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$ and $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ described above, we have the following sets and given parameters.

- $d \in \mathcal{D}$: types of change of direction (e.g. slight left turn, right sharp turn, switchbacks, etc.).
- $\mathcal{A}'_3$: Set of arcs in $\mathcal{A}_3$ other than the arcs that end at vertex T, $\mathcal{A}'_3 = \mathcal{A}_3 - \{(i, j)$ such that $j = T\}$.
- $\mathcal{A}''_3$: Set of arcs in $\mathcal{A}_3$ other than the arcs that start at vertex S or end at vertex T, $\mathcal{A}''_3 = \mathcal{A}_3 - \{(i, j)$ such that $i = S$ or $j = T\}$.
- $\mathcal{F}$: a mapping function $\mathcal{F} : \{(i, j, k) | (i, j), (j, k) \in \mathcal{A}''_3\} \to \mathcal{D}$.
- C_i : extraction cost of block i.
- $H_{i,j}$: haulage cost associated with arc $(i, j) \in \mathcal{A}''_3$. Note that the haulage cost $H_{i,j}$ of any arc (i, j) should account for the total tonnage that will traverse that arc over the life of the mine.
- P^d_i : change of direction cost at block i, where $d \in \mathcal{D}$ is the type of change of direction.

4.1.2. Variables

- r_i : binary, equal to 1 if the block i is selected as a ramp, 0 otherwise.
- x_i : binary, equal to 1 if the block i is extracted, 0 otherwise.
- v^d_i : binary, equal to 1 if the ramp undergoes a change in direction of type d at block i, 0 otherwise.
- $a_{i,j}$: binary, equal to 1 if the arc $(i, j) \in \mathcal{A}_3$ is selected as a segment of the ramp, 0 otherwise.

4.1.3. Objective function

The objective function is defined as follows, subject to a number of constraints:

$$\text{Min: } \sum_{i \in \mathcal{V}_1} C_i \cdot x_i + \sum_{(i,j) \in \mathcal{A}''_3} H_{i,j} \cdot a_{i,j} + \sum_{i \in \mathcal{V}_3} \sum_{d \in \mathcal{D}} P^d_i \cdot v^d_i$$

The objective function minimises the cost of the blocks extracted due to the ramp construction together with the haulage cost and those costs associated with changes of direction of the ramp.

4.1.4. Geotechnical constraints

$$x_i \le x_j \qquad (i, j) \in \mathcal{A}_1 \tag{1}$$

$$r_i \le x_i \qquad i \in \mathcal{V}_1 \tag{2}$$

$$r_i \le 1 - x_j \qquad (i, j) \in \mathcal{A}_2 \tag{3}$$

Constraint (1) ensures the vertical precedences are honoured in the extraction of any block. Constraint (2) ensures that every block flagged as part of the ramp must be extracted. Constraint (3) avoids building a ramp in the air.

4.1.5. Ramp constraints

$$\sum_{(i,j) \in \mathcal{A}'_3} a_{i,j} \le r_j \tag{4}$$

$$r_i + r_j \le 1 + a_{i,j} \qquad (i, j) \in \mathcal{A}''_3 \tag{5}$$

$$\sum_{(i,j) \in \mathcal{A}_3} a_{i,j} - \sum_{(j,k) \in \mathcal{A}_3} a_{j,k} = \begin{cases} -1, & \text{if } j = S \\ 1, & \text{if } j = T \\ 0, & \text{if } j \ne S \text{ and } j \ne T \end{cases} \tag{6}$$

Constraints (4) to (5) flags the block j as belonging to the ramp if and only if the block i belongs to the ramp and the arc $a_{i,j}$ has been selected. Note that Constraint (4) could be also expressed as $a_{i,j} \le r_j, \forall (i, j) \in \mathcal{A}'_3$, but Constraint (4) gives a stronger formulation. Constraint (6) ensures the connectivity of the ramp.

4.1.6. Change of direction constraints

$$a_{i,j} + a_{j,k} \le v^d_j + 1 \quad (i, j), \quad (j, k) \in \mathcal{A}''_3, \quad d = \mathcal{F}(i, j, k) \tag{7}$$

Constraint (7) identifies the change of direction at each segment of the ramp.

4.1.7. Scope of the variables

$$x_i, r_i, a_{u,v}, v^d_j \in \{0, 1\} \qquad i \in \mathcal{V}_1, \quad d \in \mathcal{D}, \quad (u, v) \in \mathcal{A}_3, \quad j \in \mathcal{V}_3 \tag{8}$$

Finally, Constraint (8) defines the scope of the variables, all of which are binary.

5. Solution method

With the mathematical formulation presented in Section 4.1, the high stripping ramp model can be implemented and solved directly with a commercial MIP-solver. In this section we introduce much faster heuristic solution approach based on backtracking algorithms. A new algorithm to find feasible paths with mutually exclusive ramp segments is introduced in Section 5.1. In Section 5.2 we describe a local search method to improve the solution found by the algorithm presented in Section 5.1. Throughout this section we use the same notation as in Section 4.1.

5.1. Mutually exclusive greedy adaptive path (MEGAP) algorithm

This new algorithm is an adaptation of the depth-first search algorithm presented in Tarjan (1972). The idea of the MEGAP algorithm is to traverse the graph $\mathcal{G}_3$ from the source node S to the sink node T following a greedy procedure. The way in which $\mathcal{G}_3$ is traversed ensures that the nodes in the solution ramp p satisfy the constraints introduced in Section 4.1. In other words, the resultant path will be a feasible solution for the In-pit ramp problem.

Before introducing the MEGAP algorithm, define $\mathcal{G}_{tc} = (\mathcal{V}_{tc}, \mathcal{A}_{tc})$ as the transitive closure of $\mathcal{G}_1 \cup \mathcal{G}_2$. Also, the stripping associated with each node $u \in \mathcal{V}_3$ is calculated as the out-degree of $u \in \mathcal{G}_{tc}$.

The MEGAP algorithm works as follow. Lines 2–6 initialise the variables `p` (ramp), cN (nodes in conflict with the ramp segments already selected), and the node attributes `visited` and `predecessor` for all nodes in $\mathcal{G}_3$. Line 7 begins the exploration of $\mathcal{G}_3$ from node S. Lines 10–12 store the solution ramp in the variable `p`. Line 13 returns the solution ramp `p`.

The function MEGAPvisit (Lines 14–22) consists of a greedy recursive exploration of $\mathcal{G}_3$ from node u taking into account the current list of conflicted nodes cN. Lines 14–15 update the conflicted node list cN. `Line 16` marks node v as visited. Lines 17–21 explore the non-visited and non-conflicted nodes v such that $(u, v) \in \mathcal{A}_3$, following a recursive greedy type procedure until the node T is reached.

The computational complexity of MEGAP is bounded by $O(2 \cdot |\mathcal{V}_3| + \Delta(\mathcal{G}_{tc}) \cdot |\mathcal{A}_3|)$, where $\Delta(\mathcal{G}_{tc})$ is the maximum degree of $\mathcal{G}_{tc}$.

5.2. Local search

The local search (LS) procedure consists of a partial exploration of all possible paths from S to T that prunes the paths that "promise" poor values for the optimisation problem. At each stage of the LS algorithm, the best feasible solution (incumbent) is stored and used to prune the exploration of the paths. More specifically, the LS procedure prunes all paths that have a value less than an incumbent value plus a given tolerance function ($tol(i)$) which decreases proportionally with the number of iterations ($i = 0, 1, \ldots, I$) of LS.

The LS procedure relies on a sub-routine called `allComb2(u)` which finds all feasible combinations of length 2 from node u (denoted by $\mathcal{C}_u^2$) in the graph $\mathcal{G}_3$. For each combination $c_u \in \mathcal{C}_u^2$ a path from S to T is calculated using Algorithm 1, with the condition that the nodes in c_u are in the resultant ramp. The LS procedure begins by running Algorithm 1, the resultant ramp (R_0) is stored as the incumbent and the cutoff value is calculated as $\text{cutoff}_0 = \text{cost}(R_0) \cdot (1 + tol(0))$. Then, a limited branching from node S is calculated using the procedure `allComb2(S)`, which gives at most $|\mathcal{C}_S^2|$ feasible ramps ($\mathcal{R}_s$). All the paths which give a cost greater than cutoff_0 are pruned. The procedure repeats the branching at all nodes in the feasible paths at distance $2 \cdot i$ updating cutoff_i at each iteration and updating the incumbent if a better value is found, until the maximum number of iterations I is reached.

Algorithm 1: MEGAP, Mutually Exclusive Greedy Adaptive Path

```
   Data: G_3 with edges sorted by stripping in increasing order, G_tc,
         starting node S, final node T
   Result: A path p that is a feasible solution for In-pit ramp problem
1  begin
2      p ← ∅;
3      cN ← ∅;
4      foreach u ∈ V_3 do
5          u.visited ← False;
6          u.predecessor ← null;
7      MEGAPvisit(S)
8      p ← p ∪ {T}
9      pred ← T.predecessor
10     while pred ≠ null do
11         p ← p ∪ {pred}
12         pred ← pred.precedessor
13     Output p
14     Function MEGAPvisit (u):
15         foreach v such that (u, v) ∈ A_tc do
16             cN ← cN ∪ {v}
17         u.visited ← True;
18         foreach v such that (u, v) ∈ A_3 do
19             if (v.visited=False) and (v ∉ cN) and (T.visited=False)
                then
20                 v.visited ← True;
21                 v.predecessor ← u;
22                 MEGAPvisit(v)
```

6. A case study

Data from a copper mine was used to test our mathematical models. The mine is located in the Antofagasta province, Chile. The mining reserves within the pit are composed of 84.2 million tonnes of ore and 52.6 million tonnes of waste. For planning purposes, the final pit is subdivided in 4 operational pushbacks, or inner phases. For simplicity, the first pushback was selected to test the performance of the high stripping model.

Two ramp location problems, ex-pit and in-pit, were studied to test our low and high stripping models respectively. For the ex-pit road case, we compared the low stripping to the high stripping model for a given topography. On the other hand, for the in-pit ramp case, we compared the high stripping model to a ramp designed by an experienced mining engineer.

Additionally, we compared the roads obtained by imposing a forbidden region to those with no such region. The model introduced in Section 3 was implemented in Matlab R2016a. The model proposed in Section 4.1 was implemented in Python 2.7 and solved using Gurobi 7.5 MIP-solver. All the pre-processing routines, such as the definition of the sets and variables, as well as the Local search method introduced in Section 5.2 were coded in Python. Execution of the code was done on an Intel Xeon machine with 32GB running Windows 10. This machine has eight processors that run at 2.40 GHz.

6.1. Ex-pit road

Two models, a low stripping and a high stripping formulation, and two scenarios, one with no obstacles and the second avoiding the grey area showed in Fig. 5, were considered to test our approaches on real mine data. The task is to find minimum cost paths from the pit exit A to four possible locations for a processing plant ($L1 - L4$).

In the first case, a nearly flat surface with an extension of 570 m by 600 m and a regular grid of 30 m was used to test the low stripping formulation. Eight incoming and eight outgoing directions were considered for each cell of the grid. The order of the graph $\mathcal{G}$ is $20 \times 21 \times 16 = 6720$. We defined a vertical gradient constraint of 10% and a horizontal change of direction cost proportional to the degree of the change. The horizontal change of direction was divided into three categories: slight turns (changes up to 45°), right angle turns (changes of 90°) and pronounced turns (changes from 90 – 180°). The costs considered for each category were US$14,580, US$21,870 and US$29,160, which correspond to a factor of 2, 3 and 4 times the mining cost for a ramp segment, respectively. Additionally, we assumed a ramp construction cost of US$340.2 per linear meter.

For simplicity, the minimum cost path was solved using a Matlab implementation of the Dijkstra shortest path algorithm. However, other algorithms might be faster than Dijkstra for the graph $\mathcal{G}$. According to Ahuja et al. (1993) Dial's implementation of the

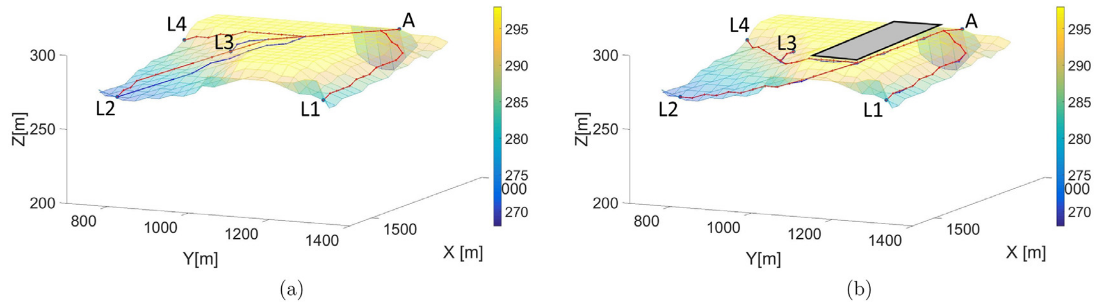

Fig. 5. Illustration of Scenario 1 (a) and Scenario 2 (b). The grey zone in Figure (b) represents a forbidden area for the roads. Roads in red represent paths obtained with the high stripping model, while roads in blue show paths obtained with the low stripping model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Minimal paths for scenario 1.

Case/Path	Length[m]	Cost[US$]	S.T.	R.T.	Solv t[s]
LS/A-L1	573.10	231,421	1	1	0.05
LS/A-L2	685.15	262,250	2	0	0.05
LS/A-L3	399.82	157,890	0	1	0.07
LS/A-L4	512.58	188,960	1	0	0.06
HS/A-L1	572.94	231,365	1	1	200.81
HS/A-L2	685.01	254,910	1	0	175.69
HS/A-L3	382.04	137,263	0	0	89.51
HS/A-L4	512.58	188,960	1	0	147.64

Table 2
Minimal paths for scenario 2.

Case/Path	Length[m]	Cost[US$]	S.T.	R.T.	Solv t[s]
LS/A-L1	573.10	231,421	1	1	0.05
LS/A-L2	878.67	342,663	3	0	0.06
LS/A-L3	770.49	327,732	3	1	0.05
LS/A-L4	964.99	379,321	2	1	0.07
HS/A-L1	572.94	231,365	1	1	102.57
HS/A-L2	878.53	342,618	3	0	126.28
HS/A-L3	770.21	327,637	3	1	132.98
HS/A-L4	964.83	379,267	2	1	146.09

Dijkstra algorithm has excellent empirical results (see Ahuja et al., 1993 pages 113–114 for Dial's implementation details).

The first scenario involved finding four minimum cost paths from point $A = (1035, 1215, 298)$ to locations $L1 = (1545, 1305, 280)$, $L2 = (1545, 795, 271)$, $L3 = (1305, 945, 292)$ and $L4 = (1185, 765, 289)$ independently, and the second scenario involved finding minimum cost paths between the same points but avoiding the grey zone in Fig. 5. Since we are asked to find the paths from A to $L1 - L4$ independently, the shortest path approach gives optimal results. However, if the task was to find the shortest network that connects all terminals A, $L1$, $L2$, $L3$ and $L4$, then a Steiner tree in graph algorithm should be used, as mentioned in Section 3. An initial North–South incoming direction at point A was specified to represent the ending of an in-pit ramp.

In the second case, a grid of 5600 regular blocks of $30\text{m} \times 30\text{m} \times 3\text{m}$ and the same economic parameters as in the first case were used. Also, a mining cost of \$1 per ton was added to account for the stripping cost, and an overall maximum wall slope of 45° was stipulated. Fig. 5 illustrates the graphic results of both cases and scenarios, and Tables 1 and 2 summarize the results for the first and the second scenario respectively. The red path represents the high stripping solution while blue is used to represent the low stripping path (in most cases both solutions overlap and only one colour is visible).

Small differences in length are explained by the different representation of the topography of the formulations, a 3D set of blocks in the case of the high stripping model and a surface in the case of the low stripping formulation. The cost function is composed of the construction cost associated with the length of the road plus the changes of direction cost, where the latter depends on the number of slight (S.T.), right angle (R.T) and pronounced turns (P.T.). The high stripping formulation also includes the stripping cost associated with the modification of the topography.

For scenario 1, the high stripping model (HS) shows better results for routes from point A to Location 2 (A-L2) and from point A to location L3 (A-L3). In both cases, one block was extracted (7290 tonnes), reducing the length of the path and saving a change of direction. The short-cut (in red) shown in route A-L3 in Fig. 5 is not an option in the low stripping formulation due to the gradient constraint and the impossibility of modifying the topography. An improvement of 2.7% and 13% were obtained in path A-L2 and A-L3 using the HS model (see Table 1).

For paths A-L1 and A-L4 of scenario 1 and all paths of scenario 2, the HS and the LS formulations gave the same results. The average computation time for the LS model is 0.06 s versus 140.19 s for the HS model. Finally, the pronounced turns (P.T.) column is not shown in Tables 1 nor 2 since none of the paths contained a pronounced turn.

6.2. In-pit ramp

As mentioned in the previous section, the low stripping formulation is not suitable for finding the minimum cost ramp in steep terrain. Therefore, we compare the performance of the high stripping formulation to a ramp previously designed by a mining engineer. We used the same initial geological and topographical data that was used by the engineer for his design.

The pushback contains a total of 22.2 million tonnes of ore and 42.4 million tonnes of waste that need to be hauled to the top level of the pit. An average haulage cost of $8.046 \cdot 10^{-4}$ [US$/ton/m] was used for the optimisation model. The binary linear programming model (BLP) was solved using the commercial software Gurobi. The resulting BLP has 218,826 constraints, 65,568 binary variables and 702,991 non-zero coefficients. The solver default optimisation parameters remained unchanged except for the time limit, which was set to 24 h (86,500 [s]).

The best solution of the BLP found by the solver within the time limit was US$4,628,154 with an optimality gap, measured against linear relaxation, of 31.7%. However, despite the significant gap this solution is 43% more economical than the engineer's design which has a cost of US$8,101,162.

We ran the local search procedure described in Section 5.2 to find a good solution in a more reasonable time (a few hours). The tolerance function $tol(i)$ was defined as $tol(i) = 0.2 \cdot e^{-0.4 \cdot i}$. The resulting ramp found by our procedure has a value of US$5,430,055,

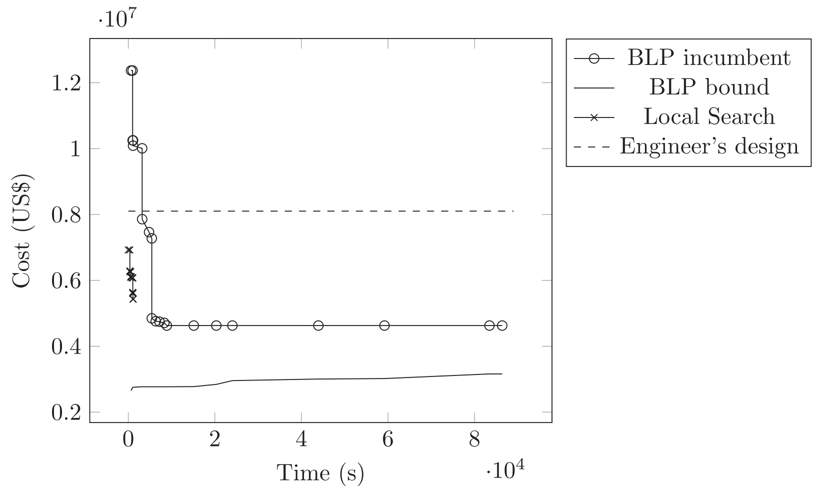

Fig. 6. Comparison of the performance (cost and time) of the binary linear programming (BLP) model and the proposed local search method (Local Search).

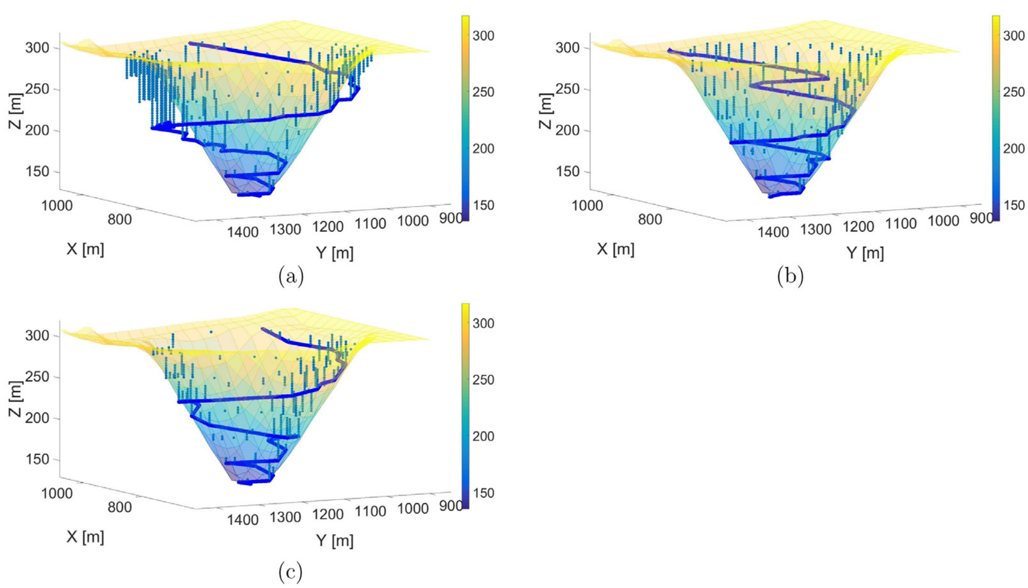

Fig. 7. Illustration of the in-pit ramps. Figures (a), (b) and (c) represent the engineer design, the ramp obtained with the local search method and the ramp obtained solving the BLP model, respectively. Ramps are represented in blue lines, while the stripping associated with the ramps is represented with blue dots. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
In-pit ramp cost comparison between the BLP model, the local search model and the engineer's design.

	Length [m]	Total cost [US$]	Stripping [US$]	Hauling [US$]	Direction [US$]	Solv t. [s]
BLP	2047.4	4,628,154	3,455,460	1,085,215	87,480	86,500
LS	2146.5	5,430,055	4,196,870	1,087,385	145,800	1145
Eng.	2289.3	8,101,162	6,714,090	1,124,632	262,440	N/A

which is 33% better than the engineer's solution. Our solution method took only 1145 [s] (approximately 19 min).

Fig. 6 illustrates the evolution in time of the solutions found using the BLP model and the one found with our local search method. Note that the Local search procedure was entirely coded in native Python which is, in general, much slower than lower level programming languages such as C or C++. Therefore, a considerable speed up in our results can be expected by implementing our proposed algorithms in lower level programming languages.

Fig. 7 shows the engineer's design (a), the ramp obtained with the local search method (b) and the ramp found using the high stripping formulation. Ramps are drawn with blue lines, and the stripping associated with the construction of the ramp is represented with blue dots.

The cost of the high stripping ramp is 43% less than the cost of the engineer's design ramp (see Table 3). Although this result shows a great improvement in cost, it is not entirely surprising because the traditional ramp design process is done manually, so it relies on the engineer's experience. It is worth noting that some of the value gained in the optimisation might be lost in the smoothing process of generating the engineering CAD designs. However, the loss in value due to the smoothing process can be controlled by choosing a suitable block discretisation as well as a proper modelling of the curvature constraints.

7. Conclusion

This article has presented a solution for the open pit ramp design problem. Two situations were studied: the low stripping and the high stripping cases. For the first case, an adaptation of the shortest path method was undertaken. In the second case, a binary linear programming approach was developed. In both cases, our methods give an exact solution to the problem within the model constraints. Additionally, both models allow gradient and curvature constraints to be handled within the given discretisation. They can incorporate construction and haulage costs and the formulations are flexible enough to find an optimal ramp that avoids any given forbidden region.

The low stripping road model can be solved in polynomial time ($O(|A| + |V| \cdot \log(|V|))$ for a graph $G(V, A)$). However, this formulation assumes that the road respects the existing topography, without the need for excavation. On the other hand, the high stripping ramp formulation can handle complex and steep terrains where extensive excavation may be required to satisfy the gradient constraint on the road. However, the high stripping model is much less efficient than the low stripping one, and does not scale well for large problems. To account for this we introduced a new algorithm that produces feasible solutions to the high stripping model (MEGAP algorithm) in linear time. We then developed a local search heuristic based on the MEGAP algorithm that rapidly finds good solutions for the high stripping model (typically within 20 min).

Finally, computational experiments were performed for ex-pit roads and in-pit ramps. When no stripping was required, the low stripping model shows the same results as the high stripping model, but running, on average, 2000 times faster. For the in-pit ramp case, the high stripping model gave a ramp with 43% less cost than the cost of the ramp previously designed by an experienced mining engineer.

Some limitations are worth noting. Our in-pit model assumes that the optimal ramp occurs on the boundary of a precomputed pushback, which might be suboptimal compared to the problem of finding a ramp and a pushback simultaneously. Currently, we are working on a new mathematical model that incorporates the ramp design into the pushback optimisation problem. Also, a combined in-pit ramp design and scheduling optimisation problem, that takes into account the ways in which the properties of the haulage network interact with the mining schedule, would be another worthwhile avenue for future work.

Acknowledgement

We would especially like to thank the anonymous referees who handled this paper for their generous and very useful suggestions. Their insightful comments led to significant improvements in the results presented. Juan L. Yarmuch is supported by the Gilbert Rigg scholarship, the National Commission for Science Research and Technology (CONICYT-Chile) grant no. 72160405, through its program Becas Chile, and FONDEF project IT16M10010.

References

Ahuja, R.K., Magnanti, T., Orlin, J.B., 1993. Network Flows. Prentice-Hall, Upper Saddle River, New Jersey.

Akay, A.E., Aruga, K., Bettinger, P., Sessions, J., 2013. Using optimization techniques in designing forest roads and road networks. J. Fac. For. Istanb. Univ. 15 (1), 49–62.

Anderson, A.E., Nelson, J., 2004. Projecting vector-based road networks with a shortest path algorithm. Can. J. Forest Res. 34 (7), 1444–1457. doi:10.1139/x04-030.

Atkinson, T., 1992. Design and Layout of Haul Roads. In: Hartman, H.L. (Ed.), SME Mining Engineering Handbook. Society for Mining, Metallurgy, and Exploration, Littleton, Colorado, pp. 1334–1342. chapter 13.4

Bertsimas, D., Tsitsiklis, J., 1997. Introduction to Linear Optimization. Athena Scientific, Belmont, Massachusetts.

Burdett, R., Kozan, E., Kenley, R., 2015. Block models for improved earthwork allocation planning in linear infrastructure construction. Eng. Optim. 47 (3), 347–369. doi:10.1080/0305215X.2014.892594.

Casal, G., Santamarina, D., Vázquez-Méndez, M.E., 2017. Optimization of horizontal alignment geometry in road design and reconstruction. Transp. Res. Part C 74, 261–274. doi:10.1016/j.trc.2016.11.019.

Chang, A.J., Brazil, M., Rubinstein, J.H., Thomas, D.A., 2012. Curvature-constrained directional-cost paths in the plane. J. Glob. Optim. 53 (4), 663–681. doi:10.1007/s10898-011-9730-1.

Couzens, T., 1979. Aspects of Production Planning: Operating Layout and Phase Plans. In: Crawford, J., Hustrulid, W.A. (Eds.), SME Mining Engineering Handbook. Society of Mining Engineers of American Institute of Mining, Metallurgical, and Petroleum Engineers, New York, pp. 217–231. chapter 16

Dowd, P., Onur, A., 1992. Optimising Open Pit Design and Sequencing. In: 23rd International Symposium of Application of Computers and Operations Research, pp. 411–422.

Epstein, R., Weintraub, A., Sapunar, P., Nieto, E., Sessions, J., Musante, H., Bustamante, F., Sessions, J.B., 2007. A combinatorial heuristic approach for solving real-Size machinery location and road design problems in forestry planning. Oper. Res. 54 (6), 1017–1027. doi:10.1287/opre.1060.0331.

Epstein, R., Weintraub, A., Sessions, J., Sessions, B., Sapunar, P., Nieto, E., Bustamante, F., Musante, H., 2001. PLANEX: a System to Identify Landing Locations and Access. In: Proc. of the International Mountain Logging and 11th Pacific Northwest Skyline Symposium, pp. 190–193.

Gill, T., 1999. Express Road routing: the Application of an Optimal Haul Road Generator to Real World Data. In: Whittle, J. (Ed.), Optimizing with Whittle. Whittle Programming Pty. Ltd., Perth, Western Australia, pp. 71–80.

Heinimann, H.R., Stückelberger, J.A., Chung, W., 2003. Improving Automatic Grid Cell Based Road Route Location Procedures. In: Stampfer, K. (Ed.), Austro2003: High Tech Forest Operations for Mountainous Terrain. Schlaegl - Austria, pp. 1–10.

Kaufman, W.W., Ault, J.C., 1977. Design of Surface Mine Haulage Roads: a Manual. Technical Report. United States. Department of the Interior, Bureau of Mines., Washington.

Liu, K., Sessions, J., 1993. Preliminary planning of road systems using digital Terrain models. J. Forest Eng. 4 (2), 27–32.

Meignan, D., Frayret, J.-M., Pesant, G., Blouin, M., 2012. A heuristic approach to automated forest road location. Can. J. Forest Res. 42 (12), 2130–2141. doi:10.1139/x2012-140.

Mondal, S., Lucet, Y., Hare, W., 2015. Optimizing horizontal alignment of roads in a specified corridor. Comput. Oper. Res. 64, 130–138. doi:10.1016/j.cor.2015.05.018.

Najafi, A., Richards, E.W., 2013. Designing a forest road network using mixed integer programming. Croat. J. Forest Eng. 34 (1), 17–30.

Newman, A.M., Rubio, E., Caro, R., Weintraub, A., Eurek, K., 2010. A review of operations research in mine planning. Interfaces 40 (3), 222–245. doi:10.1287/inte.1090.0492.

Pushak, Y., Hare, W., Lucet, Y., 2016. Multiple-path selection for new highway alignments using discrete algorithms. Eur. J. Oper. Res. 248 (2), 415–427. doi:10.1016/j.ejor.2015.07.039.

Stückelberger, J.A., 2008. A Weighted-Graph Optimization Approach for Automatic Location of Forest Road Networks. ETH Zurich.

Tannant, D., Regensburg, B., 2001. Guidelines for Mine Haul Road Design, university edition University of Alberta Press, Edmonton.

Tarjan, R., 1972. Depth-First search and linear graph algorithms. SIAM J. Comput. 1 (2), 146–160. doi:10.1137/0201010.

Thompson, R.J., 2011. Design, Construction, and Maintenance of Haul Roads. In: Darling, P. (Ed.), SME Mining Engineering Handbook. Society for Mining, Metallurgy, and Exploration, pp. 957–976. chapter 10.6

Yarmuch, J., Epstein, R., Cancino, R., Peña, J.C., 2017. Evaluating crusher system location in an open pit mine using Markov chains. Int. J. Min. Reclam. Environ. 31 (1), 24–37. doi:10.1080/17480930.2015.1105649.