

Development and analysis of a methodology to generate operational open-pit mine ramp designs automatically

Nelson Morales¹ · Pierre Nancel-Penard² · Nelson Espejo²

Received: date / Accepted: date

Abstract A critical step in planning an open-pit operation corresponds to the design of ramps required to access the different sectors and levels. This design is very complicated because the ramps affect the excavation's shape, therefore, its economic value. Thus, planners generate contours to be used as reference for the design. These contours (or equivalently the volume they contain) are generated by mathematical models that aim to optimize the economic value that is contained in them. Then, through Computer-Aided Design software, planners manually draw the operational design that, hopefully, retrieves as much value as possible from the reference volumes and is operationally feasible. Unfortunately, this manual process does not ensure the quality of the design, leading to results that are not optimal and are highly dependent on the engineer's expertise.

This article presents a methodology that starts from the same reference volume that the planner uses to generate a mine design automatically. It works in two steps. Firstly, it uses integer programming to generate a new discretized contour but containing enough space for ramps. Secondly, it utilizes a computer algorithm to transform the discretized profile into an operational pit design that complies with the mine design's geometrical constraints.

To study the proposed methodology's applicability, we considered three cases with their corresponding reference pushbacks and used our approach to create 15 different operational designs (in total). In two of the three cases, the methodology generated profiles that were, at worst, within less than 2% deviation in value and tonnage. In the third case, the loss in economic value was more than 11%; however, this performance was equivalent to a manual design produced by an engineer.

Keywords Open pit design · mine planning · linear programming

1 Introduction

Mining is an important economic activity. For example, copper exports represent 13% of Chile's nominal gross domestic product between 2008 and 2014. Since 2007, Chile has produced more copper tons than any other country in the world (Cochilco, 2017).

✉ Nelson Morales
E-mail: nelson.morales@polytmlt.ca

Pierre Nancel-Penard
E-mail: pierre.nancel@amtc.cl

Nelson Espejo
E-mail: nespejo@delphoslab.cl

¹ Département de génies civil, géologique et des mines, Polytechnique Montréal

² Delphos Mine Planning Laboratory, Department of Mining Engineering, Faculty of Physical and Mathematics Sciences, University of Chile, Beauchef 850, Santiago, Chile; Advanced Mining Technology Center, University of Chile, Avda. Tupper 2007, Santiago, Chile

One of the most common mining methods is open-pit mining, which consists of excavating the material from the surface of the deposit and then transporting it to its destination for processing or storage. In this method, the cost per ton is small enough (compared to underground operations) to allow the extraction of profitable material (ore) together with large amounts of waste, which can lead to the extraction of hundreds of millions of tons and excavations that are hundreds of meters deep (Altimi *et al.*, 2021). However, as the excavation of the pit progresses and the pit gets deeper, the mining operation must ensure that the pit walls are stable. Therefore, the planning team will design the pit shape with the aim to maximize the stope walls, in order to minimize waste extraction, while maintaining reasonably a low likelihood of a slope failure Hartman and Mutmansky (2002).

Due to the relevance and impact on the economic value of a mine, the design of a pit takes place in the strategic stages of the planning of the mine. At this level, in order to optimize the value of the material extracted, mining engineers utilize a discretization of the deposit into a 3-D array of regular blocks, called the *block model*. Very roughly, the block model is a table where each row contains the relevant attributes of a block, for example, its location, tonnage, grades, among others. Notice that the block model is an input for the planning of a mine.

The discretization of the terrain into blocks facilitates the use of computer software and mathematical models for optimizing the design of the pit. For example, a first step in the planning process corresponds to the computation of the *ultimate pit* or *final pit limit*, which defines what is economically mineable from a given deposit, i.e., the ultimate pit identifies which blocks should be mined and which ones should be left in the ground Dagdelen (2001). The only constraint in the ultimate pit problem is to comply with the overall slope angles of the walls, which in terms of the block model are encoded as *precedences*. These precedences enforce that when an individual block is extracted, others located on top of it and within certain slope angles must also be extracted to keep the pit walls stable.

The input of the ultimate pit problem is given a block model with economic values defined for each block, and overall slope angles encoded as precedence arcs. A solution to the problem is a set of blocks of maximum value respecting said precedences. (Lerchs and Grossman, 1965).

Because of its simple mathematical structure, which does not consider constraints like capacity of extracted or processed material, the ultimate pit problem can be solved very efficiently. Indeed, Lerchs and Grossman (1965) proposed a polynomial algorithm to solve the final pit problem.; however, Picard (1976) demonstrated that the final pit problem is equivalent to the maximum closure problem, which in turn can be formulated as a minimum-cut problem. This permitted Hochbaum (Hochbaum and Chen (2000); Hochbaum (2008); Hochbaum and Orlin (2012)) to propose a more efficient polynomial algorithm based on a pseudo-flow approach. Indeed, these algorithms allow solving large instances of the minimum cut problems very quickly and, consequently, can be used to solve large instances of the final pit problem.

Lerchs and Grossman (1965) also showed that nested pits could be generated by varying the price of the metals being extracted; therefore, partitioning the deposit into many nested pits, i.e., a sequence of pits such that each one is contained (as set of blocks) in the next pit. Using different criteria like mineral content or spatial considerations, mine planners select some of these nested pits to be the basis for the design of mining phases. This method is commonly used in commercial software that is the standard in the industry. See Gemcom (2018) for an example.

The selection of the nested pits allows partitioning the ultimate pit into several volumes, namely *pushbacks* as follows: The first pushback corresponds to the first nested pit, and the i -th pushback ($i > 1$) corresponds to the difference (as set of blocks) between the i -th nested pit and the $(i-1)$ -th nested pit. The idea of these pushbacks is that each of them corresponds to a manageable portion of the deposit inside the ultimate pit limit that may be mined, processed, and refined in several years/periods (Asad and Topal, 2011). For reasons that will be clear soon, we will refer to the pushbacks computed with block support as *discretized pushbacks*.

An alternative approach to generate discretized pushbacks is to rely on direct block scheduling (DBS). Mathematically, DBS is a generalization of the ultimate pit problem that considers multiple periods and potential destinations for the blocks, and capacity and blending constraints. Introduced by Johnson (1968, 1969), this approach can be implemented as a large optimization

problem that schedules the blocks' extraction over a given number of planning periods. Contrarily to the ultimate pit problem, DBS considers extraction capacities and blending constraints can be enforced for each period and destination of blocks. The objective of the optimization problem is to maximize the discounted value of extracted blocks. Even though DBS is more focused on determining the best extraction period of blocks rather than defining pushbacks for extraction, the technique can be used to define pushbacks, for example, by considering one per planning period.

Many researchers have studied DBS, because of its interest in industry and its computational complexity, making recent literature about the problem abundant. For example, the reader can refer to Lambert et al. (2014) for a tutorial that presents various formulations for the constrained pit problem with extraction capacities constraints and fixed block destination and MineLib (Espinoza et al., 2013) for a library of public instances with known solutions. In terms of algorithmic approaches, Bienstock and Zuckerberg (2009, 2010) proposed an algorithm based on the relaxation of capacity and blending constraints that improved significantly the time to obtain a solution for the integer relaxation of the problem. Lamghari et al. (2014); Liu and Kozan (2016); Samavati et al. (2017); Jélvez et al. (2016) propose different heuristic methods to address the problem which are tested on the public set of instances provided in MineLib. Finally, improving on Bienstock and Zuckerberg (2010) ideas and proposing new algorithms, Letelier et al. (2020) show that they can solve very large instances (5 millions blocks, 50 planning periods) within a few hours and an integer gap smaller than 1%.

The use of blocks provides planners with a powerful tool for optimizing the economic value contained in the discretized pushbacks while keeping a reasonable approximation of its shape. Unfortunately, the mine's operation cannot follow discretized pushback perfectly. Indeed, mining engineers utilize the discretized pushbacks as a guide and then, through Computer-Aided Design (CAD) software, create actual designs of the mine; therefore, transforming the discretized pushbacks into *operational pushbacks*, which aim to resemble the discretized ones. The rationale of this is that because discretized pushbacks are optimized for value, an operational pushback that resembles the contour of the discretized pushbacks as closely as possible should, in theory, retrieve a similar optimal economic value and contain about the same amounts of ore and waste.

Unfortunately, the design of operational pushbacks needs to comply with many considerations that are not included in the optimization models presented before. This makes the transformation of discretized pushbacks into operational ones a challenging task that is not supported by optimization models or algorithms; however, it requires determining several volumes (or equivalently, contours) that are each geomechanically stable, and that can be constructed by mining machinery. In this regard, in this work we consider that stability and constructability of the pit walls follow from the compliance with several design parameters that include the height of the mining levels or *benches*, the slope and width of the ramps that will allow equipment to move between different levels, the overall and the inter-ramp angles. For a more detailed description of the wall design parameters and stability issues related to the pit walls, we refer to Read and Stacey (2009) and Hustrulid et al. (2013).

Figure 1 presents some of the pit wall design parameters and concepts, and a section of a wall of a discretized pushback represented by the blocks in gray with an overall slope angle of 45° as well as several operational pushbacks. Since all of them comply with the design parameters and the same overall slope angle, this also illustrates that the discretized pushback provides only a guide for the operational design and that different feasible designs may differ significantly from each other. All this makes the process of operational design that retrieves as much value as possible from the discretized pushback an arduous task, considering that in the actual problem there may be several pushbacks to design consistently.

It follows that having a tool to assist the process of generating operational pushbacks would simplify and shorten the design task and allow planners to evaluate several alternatives before doing the actual ramp design, hence enabling them to find the best possible designs. For example, the model could be used to find the best starting points for the ramps and their impact on the design if these parameters are modified.

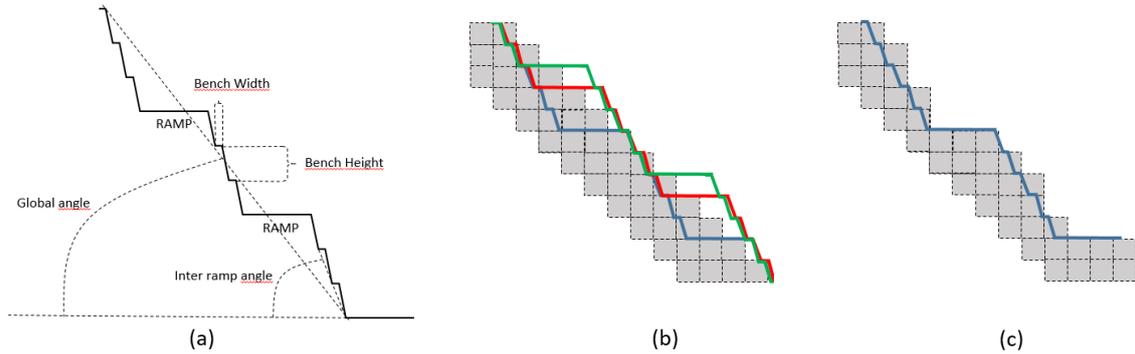


Fig. 1 Section view with a definition of pit wall design concepts, block pit profiles, and pit profiles. (a) Main concepts for pit wall design. (b) Discretized pushback profile (defined by greyed blocks) and several compatible operational profiles (red, blue, and green). (c) A discretized pushback profile (defined by gray blocks) with space to accommodate the ramp and an operational profile that approximates it.

Because of the above, in this paper, we propose a methodology that can automatically generate an operational pushback using as an input the design parameters and the discretized pushback that is used as a reference for its volume and contour. The methodology consists of two steps:

1. Starting from a discretized pushback, a tolerance range for the contour, and ramp parameters (width and decline) an optimization model generates another discretized pushback that has space to accommodate the ramp, but with maximum economic value (evaluated as the sum of economic values of its blocks).
2. A computer algorithm utilizes the discretized pushback generated in the first step plus more detailed design parameters and outputs an operational design of a pushback similar in shape to the input and incorporating the design parameters. We name this procedure the *pushback design algorithm*.

Figure 1(c) shows an example of a wall profile of the discretized pushback generated by the mathematical model, with the blocks outside the pushback colored in grey. The same figure also shows an operational profile generated by the pushback design algorithm obtained from the discretized pushback used as a guide.

To show the methodology's applicability, we tested it on three different block models and considered different design parameters to obtain 15 possible designs (in total). In these instances, we applied the mathematical model and the pushback design algorithm to compare the results with the initial pushback and with manual designs elaborated by engineers to show the consistency of the approach.

The rest of the paper is organized as follows. After reviewing the literature in Section 2, the paper continues by introducing the problem in Section 3 and then describing the methodology in Section 4, which contains the mathematical model and a description of the pushback design algorithm. Then, the numerical experiments are described and analyzed in Section 5. Finally, Section 6 is dedicated to the conclusions and recommendations.

2 Related work

Several other authors have recognized the drawbacks of the current approaches (based on nested pits and DBS); hence, they have addressed the issue through different models and techniques, as we review next. Figure 2 sketches different aspects of strategic mine planning and design, considering existing approaches, where the proposed methodology is located, and how they relate to each other, which are currently used by planners or have been developed in academic publications.

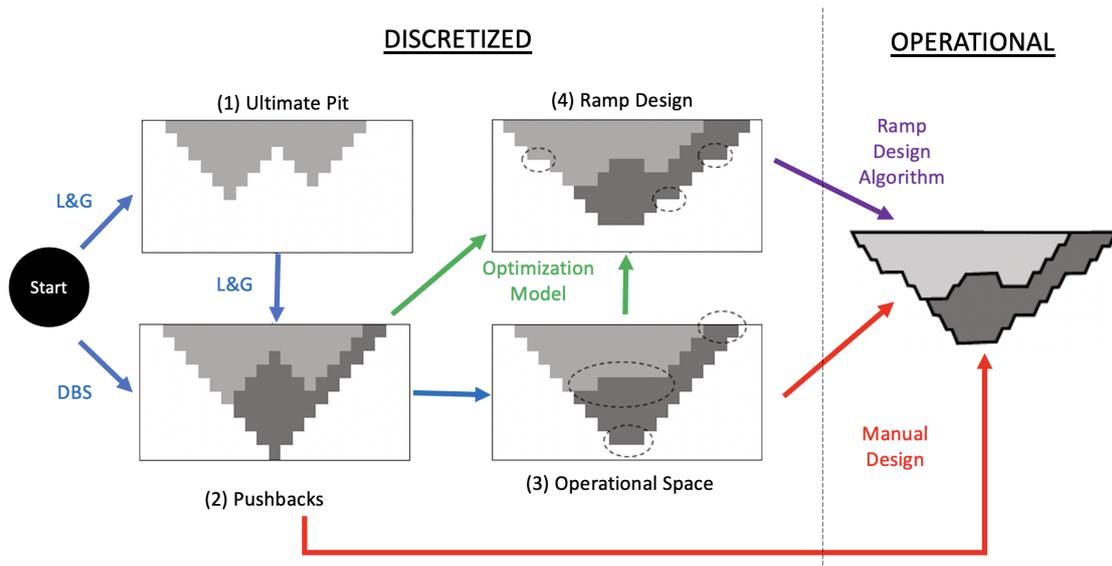


Fig. 2 Schematic of open pit design, from block model to a designed mine consisting of several operational pushbacks. (1)-(4) represent different aspects of the design of a pushback. Blue and red arrows are current methods. Green and purple are stages where the proposed methodology contributes.

The work by Cullenbine et al. (2011) proposes a small variation of DBS, but such that it aims to be more operational by adding extra constraints that look to have more space for excavating at the pit's bottom. For this, they require that each extracted block must have at least one of its four adjacent neighbors also extracted. The resulting model is computationally hard to be solved; thus, the authors proposed a rolling horizon heuristic to find nearly optimal solutions. In Figure 2, this paper could be considered as one generating pushbacks with some operational space (2), without going through the ultimate pit (1) or pushbacks (2).

Tabesh et al. (2014) develop a multi-step pushback design algorithm based on mathematical programming and clustering to generate pushbacks composed of mining polygons while also complying with capacity constraints, for which they assign clusters to pushbacks. They use a hierarchical clustering that generates many small shapes and a post-processing procedure to obtain more practicable forms that are considered for the scheduling. In the schema of Figure 2, their approach aims to generate pushbacks with operational space (3) directly.

Bai et al. (2018) address the problem of generating a sequence of nested pushbacks but considering enough operational space for equipment operation. In their methodology, they utilize discretized pushbacks generated by the nested pit approach and propose certain geometric operations that can be applied to individual pushbacks to produce new ones, also discretized, but respecting minimum bench and bottom widths. That is, in Figure 2, their methodology takes pushbacks (2) and produces pushbacks with operational space (3). Using their approach and an approximation of the Net Present Value (NPV), they show an impact between 0.5% and 7% in the NPV compared with the standard using the nested pit approach.

Nancel-Penard and Morales (2021) introduce a mathematical model to compute a series of nested pits that comply with capacity and geometrical constraints related to minimum operational spaces at the bottom of the pit. They develop a pre-processing algorithm to make the problem tractable and apply it in several instances to show that the methodology is able to produce discretized pushbacks with high NPV values. This paper fits in section (3) of Figure 2.

In another research line, some authors have studied the accuracy of using precedences for pit optimization. For example, Khalokakaie et al. (2000) criticize the use of a set of few fixed prece-

dences to model slope angles and introduce an extension of the Lerchs and Grossman algorithm that can use slope definitions that can change for different regions of the orebody. They consider four principal slope angles (North, South, East, and West) and approximate slope angles in between using linear interpolation in each region. Then other authors (Sattarvand and Shisvan (2012); Gilani and Sattarvand (2015)) improve the interpolation approach by using non-linear interpolation methods, which they show to work better.

We observe that the previous works focus on the generation of discretized pushbacks, which comply with overall slope angles and may or not include space for equipment operation. However, they do not consider the more specific aspect of the design ramps to ensure equipment movement between different levels. Indeed, we envision that our methodology as complementary to these works. We depict this in Figure 2, where our methodology could use the results of any of these methods as input. Furthermore, the figure also emphasizes that all previous works rely on a manual stage to generate the operational design, while our methodology automates this step by using a pushback design algorithm.

The topic of ramp design in mining has been addressed before. For example, Brazil *et al.* (2005, 2007), Brazil and Thomas (2007) study the problem in underground mining, for which they develop and apply a methodology on the three-dimensional space with a fixed gradient, a theorem presented in Dubins (1957) that characterizes the shapes of the shortest admissible path between two oriented points in the plane. By assimilating a ramp as a curve consisting of segment and circumference sections, they can optimize the ramp's shape to a fixed curvature and gradient, end-points, and prohibited zones. Following their work, Yardimci and Karpuz (2017) added minimum curvature constraints and applied a genetic algorithm to minimize the overall underground ramp's cost. Montané *et al.* (2019) use two optimization models: one for estimating the location of a ramp and another for scheduling construction and production) in a Bench and Fill project to show that their approach can reduce development costs and increase NPV.

Unfortunately, the problem of designing ramps in open-pit mining is very different from the underground case. For example, because the material on top of the ramps has to be removed, it is impossible to have sections of a ramp on top of others; also, the ramps' starting and ending points could be part of the design process. Conversely, underground ramps are generally constructed from specific locations on the surface to reach preset extraction zones at given points, and because they are tunnels, they can pass under other portions of the same ramp. Open-pit ramp design has to respect overall and inter-ramp slope angles, which are linked to each other and strongly depend on the ramp itself, while in the case of underground ramps, the constraints are more local to the ramp.

A complete description of ramp design in open pit mining can be found in Couzens (1979) and in Hustrulid *et al.* (2013). Dowd and Onur (1992) present an algorithm that takes a block at the bottom of the pit and generates two ramps (one clockwise and one anti-clockwise), by selecting blocks from bottom to top iteratively, level by level. The method does not allow changing having switchbacks (i.e. 180° turns), thus, the ramp with maximum profit is then selected. Gill (1999) proposed a dynamic programming algorithm that minimizes the cost of the ramp for a given pit. They indicate that they use a graph and consider gradient constraints but unfortunately, do not provide further details. Sussmann (1995) addresses the problem by modeling it as a minimum path in three dimensions, but does not consider gradient constraints. Li and Klette (2007) proposed an algorithm that works on a discretized three-dimensional, then solves some Euclidean Shortest Path problems, like the short path inside polygons, also without gradient constraints. Finally, Thompson and Visser (1999) studied the relation between the quality of ramps and hauling costs.

Some recent works address the problem of optimizing open pit ramp design directly.

Morales *et al.* (2017) introduce an integer linear program to produce a discretized pushback such that its border is contained within a given range and the economic value of the pushback is maximized, i.e., they model the same problem, but with a different set of variables. Their formulation is more complex, less flexible, and harder to implement. Applying that model, Nancel-Penard *et al.* (2019) present cases studies, discuss the practical aspects of its utilization, and generate an operational design that uses the model's output as a guide. They conclude that the

guide of the model helps generating a designed pushback that is very similar in quality compared to a manual design developed independently, but in a fraction of the time and effort.

Espejo et al. (2019) apply the methodology proposed in this paper to five block models available in MineLib (Espinoza et al., 2013), showing that the technique can generate designs that are between 1 and 20% different (in value and tonnage) from the initial pushback. However, their work provides no details about the implementation of the model or design algorithm, and does not compare the results with manual designs elaborated by planners. Moreno et al. (2020) propose to use a statistical method (orthogonal array) to produce a gross estimation of the economic value of a design from few parameters (like starting point, ramp direction, presence/absence of switchbacks). They consider two block models in MineLib and generate 27 designs for each of them using the methodology proposed in this paper to show that their statistical model estimates the economic value with an error between 9% and 12%.

Yarmuch et al. (2020) address the problem of ramp design inside and outside a discretized pit, separately. For the out-pit ramp, the problem is reduced to an application of the shortest path problem. For the in-pit ramp, their approach is similar to Nancel-Penard et al. (2019); however, their goal is to minimize the ramp’s construction and hauling costs and requires the blocks to be aggregated in the X and Y directions so that their size corresponds to the width of the ramp. Contrarily to this, the model presented in this paper decides what blocks need to be excavated to shape the ramp and in order to maximize the economic value of the resulting discretized pushback. As such, the approach is more general because it does not require aggregating blocks (which reduces the fidelity of the resulting volume), but also because it can consider the total economic value of the resulting pushback, including construction and hauling cost, if needed.

The in-pit ramp design in previous work is extended in Yarmuch et al. (2021), which addresses the problem of a single mineable discretized pushbacks for maximum profit. They consider several path patterns which can be assembled to create the ramp, and utilize two penalization parameters “closeness” and “compactness” to control the geometry of the resulting pushback. Using their model, they do a case study where they generate multiple pushbacks that are then scheduled for production, showing an increase in NPV of 6.7% with regards to a schedule based on nested pits. Their approach is limited by the fact that the number of variables is quadratic in the number of blocks, and that results strongly depend on the selection of the penalization parameters, which have to be calibrated by the user.

Finally, another relevant work corresponds to solutions existing in commercial software. Indeed, it is important to observe that current software provide practitioners with powerful tools for the design of ramps, however the approach is very different and more limited:

- The proposed method first determines approximate location of the ramp for each bench in the mine, then generates a ramp and finally it produces the contour of the pushback. Commercial software first projects the contour of the pushback for given pit bottom, then projects a ramp so that it fits in the contour.
- Software requires the user to indicate the direction of the ramp (clockwise or counterclockwise), the starting point of the ramp, and benches where switchbacks will be located. In our methodology this can be automatic or controlled by the user.
- In the proposed solution, the ramp is computed with the aim that the value of the volume is optimized. For example, the methodology may suggest the slope of the ramp to be less steep than the actual constraint. Ramps projected by commercial software do not do this unless it is required manually.
- In commercial solutions, there is no guarantee that the global slope angle of the pit will be satisfied. The proposed pushback design algorithm, however, makes adjustments to achieve this as close as possible.

3 Modeling aspects

In this section, we provide a detailed description of the modeling of the problem addressed in the paper, which is the design of an operational pushback to achieve a high economic value. The

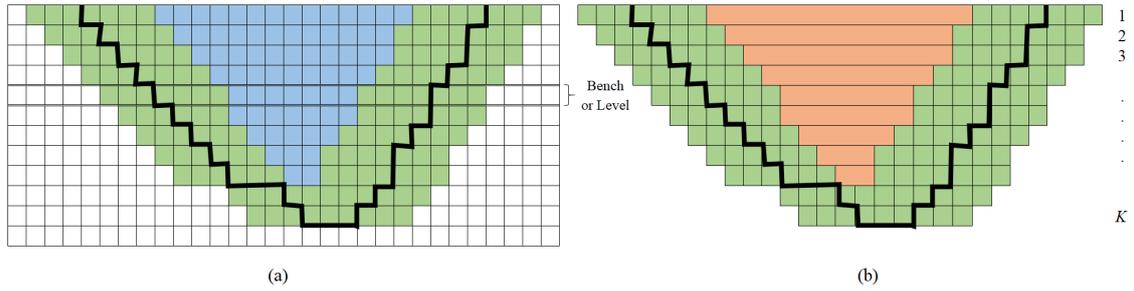


Fig. 3 A 2-D example of a block model. (a) The original pushback (contour in black line) within the whole block model, a bench (or level), the reduced pit (blue), and pit boundary (green blocks). (b) Nodes in the digraph to model the ramps, represented as squares and block aggregations of the reduced pit (orange).

concepts introduced here will be used in the next sections to introduce the proposed two-stage methodology, i.e., an optimization model and an pushback design algorithm.

The main input for the optimization model is a block model and the contour of a discretized pushback (for example, the contour of the ultimate pit), which we name the *original pushback*.

The model's solution is another discretized pushback that we call the *optimized pushback*, which has enough space to accommodate the ramp and maximum economic value. This value is calculated as the addition of the blocks' economic values, for which we will assume that each block already has an economic value provided in advance. We also assume that the global slope angles for the block are known, and encoded as a set of predecessors.

We will look for an optimized pushback similar in shape (and therefore tonnage) to the original pushback. For this, we consider two tolerance parameters δ^- , δ^+ , so that the contour of the optimized pit must lie at most δ^- blocks within the original pushback and at most δ^+ blocks outside the contour of the original pushback. Notice that even though no particular condition is required for the contour of the original pushback, for practical reasons the volume should not be too small, and it would be recommended that it complies with overall slope angles, so an optimized pushback can be computed within the tolerance ranges. However, these conditions are most likely to hold in practical scenarios.

Figure 3(a) depicts an artificial 2-D example, where squares represent blocks. The contour of the original pushback is shown as a thick black line. In this example, we have that $\delta^- = 4$ and $\delta^+ = 3$. The figure also presents the *reduced pit* (colored in blue), which is the set of blocks that are within the original pushback and at a distance greater than δ^- from its boundary, and the *pit boundary* (colored in green), which is the set of blocks within the tolerance parameters.

Notice that because the pit boundary constraints the shape of the optimized pushback, no white blocks in Figure 3(a) can be extracted. Conversely, all blocks in a given level of the reduced pit (blue blocks) must be extracted if the ramp reaches that level. It turns out that only the green and blue blocks take part in the optimization process, but also that the blocks in the reduced pit could be aggregated level by level. Figure 3(b) presents the blocks that participate in the optimization problem, i.e., the blocks in the pit boundary in green and the *aggregation of blocks* (one per level) in orange. In particular, the optimization problem's objective function will be the sum of the economic values of green blocks and orange benches that are selected as part of the solution.

3.1 The ramp as a path in a directed graph

We formulate the problem of finding the best ramp as the search for a path of blocks in a directed graph. In our modeling, the blocks are considered to be smaller than the ramp dimensions, and therefore the ramp may be several blocks wide. Thus, we will consider that the path in the graph

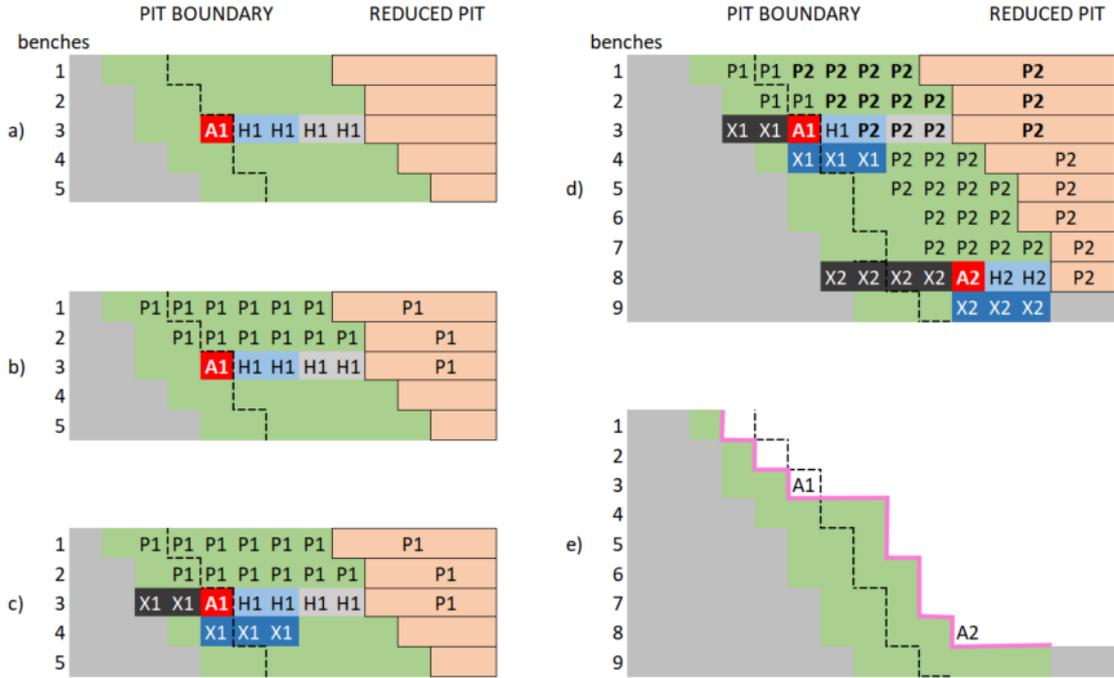


Fig. 4 Sections of a conceptual design of a feasible ramp and a 3-block wide ramp. (a) Pit boundary (in green), contour of original pushback (segmented line) and aggregated blocks of the reduced pit (orange). Interpolation block $A1$ (in red) is selected as part of the ramp, $H1$ are extracted to make room for the ramp (b) $P1$ blocks and aggregated blocks extracted due to slope precedences. (c) Blocks cannot be extracted to ensure that the ramp is not floating. (d) A second interpolation block ($A2$) selected to be part of the ramp, $H2$ blocks removed to make space for the ramp, $X2$ blocks cannot be removed, $P2$ blocks and aggregations to be extracted due to slope precedences. (e) Final resulting profile (in magenta).

corresponds to blocks located in the outer part of the ramp, that is, the one that is adjacent to the wall of the optimized pushback. We call them *interpolation blocks* and observe that they define the ramp completely.

Figure 3(b) depicts the vertices in the digraph, which correspond to the set B of blocks in the boundary, which are represented as green squares.

An arc between nodes b and b' exists if and only if $b, b' \in B$ and it is possible to connect b and b' using a ramp, that is if blocks b and b' are located at two consecutive levels, and they are far enough such that the ramp slope is within the accepted range. We denote the set of arcs as \mathcal{A} . It is worth emphasizing that two adjacent blocks in the graph (i.e., such that $(b, b') \in \mathcal{A}$) are not necessarily next to each other in the block model. For example, if the blocks are cubes with an edge of 10 meters and if the ramp's decline is 10%, then blocks b and b' are ten blocks apart and at different levels. The blocks in between b and b' are named the *elementary path* between b and b' . We provide a precise mathematical definition of the elementary path in Section 4.1.

Using the construction described in this section, a path in the digraph $\mathcal{D} = (B, \mathcal{A})$ that starts on the surface determines a unique ramp by selecting the vertices (interpolation blocks) as the locations where the ramp goes through.

3.2 Modeling of the excavation due to the ramp

As a path in \mathcal{D} defines a ramp, it also determines that some blocks have to be extracted to make room for the ramp and satisfy slope angles, while others remain in place to make sure that the ramp is not floating. Also, if the ramp reaches a certain level, all the blocks in the reduced pit's aggregated node corresponding to that level and above have to be extracted.

Figure 4 illustrates the sets of blocks to be removed or left on the ground. The figure presents several sections of the same profile in the case of a three blocks wide ramp. For the sake of simplicity, the figure introduces the sets incrementally, as follows:

- a) Presents a partial section of 5 benches and assumes that an arc $(\cdot, A1)$ was selected as part of the ramp. Because of this, interpolation block $A1$ and the two light blue blocks $H1$ are extracted to make room for the ramp (which is of width 3), but also two light grey blocks labeled as $H1$ are also extracted to prevent the ramp from separating from the reduced pit.
- b) Shows what blocks and aggregations need to be extracted due to slope precedences by labeling them as $P1$.
- c) Presents blocks that must not be extracted: dark blue blocks labeled as $X1$ are fixed because they become the ramp's surface, while dark grey blocks labeled $X1$ also must remain unextracted, so $A1$ is the outer part of the ramp at that level.
- d) Extends the example to consider 9 benches a second arc $(\cdot, A2) \in \mathcal{A}$ selected as part of the ramp path: Interpolation block $A2$ and those labeled $H2$ must be extracted to make space for the ramp (there are no grey blocks because in this case the ramp is close enough to the reduced pit); $X2$ blocks must remain in place; and blocks and aggregations marked with $P2$ need to be extracted due to slope precedences. Notice that some blocks labeled as $P2$ were already forced for extraction due to $A1$. We have used a bold font to mark them.
- e) Shows the resulting discretized operational profile using a magenta line. The figure also displays the original pushback profile (as a segmented line), and the labels of interpolation blocks $A1, A2$ for reference.

Notice that, because of the precedence constraints, preventing the excavation of the blocks X_i imposes a significant constraint on the ramp's location for more profound levels. For example, in Figure 4, $A2$ is constrained by the existence of a ramp going through $A1$ as no blocks labeled $X2$ could be part of the ramp because slope constraints imply that blocks $X1$ should be extracted, however they cannot be mined due to $A1$.

3.3 Remarks

We observe that modeling the ramp in terms of arcs between consecutive levels implies that the number of arcs leaving a block b is small; thus, the size of the set \mathcal{A} is linear in the number of blocks in the pit boundary. Overall, the number of variables and constraints in the optimization problem is linear in the size of the pit boundary.

It is also worth noting that the nested pits or DBS models can estimate the depth of a ramp, but not its length, because the models do not have information about the location of the ramp. In our modeling, the selected arcs for construction provide an estimation of the location of the ramp; therefore, an approximation of its length but also of the distance between blocks in the bench and the ramp. This can be used to provide a finer estimation of the costs of in-pit ramp construction and material hauling, which in turn, lead to a more accurate estimation of the value contained in the resulting discretized pit. In fact, this is an important contribution with regards to the modeling in Morales *et al.* (2017), which used variables defined on the sets of blocks and not arcs that prevented these finer estimation.

In terms of the pit boundary, it is relevant to mention that its size and location can be controlled through parameters δ^- and δ^+ . On the one hand, this has the drawback that it may bias the obtained solution if they are not chosen correctly. On the other hand, it has two significant advantages. First, it limits the number of blocks in the problem, reducing the total size of the

optimization problem, and, in turn, its computation time. Second, from a methodological point of view, they provide the user with more control of the solution. This is useful, for example, if the method is applied to a pit contained in the ultimate pit. In this case, because the model aims to maximize contained value, a solution will tend to extract blocks outside the original pit; thus, selecting small values for δ^+ may prevent this. It is also worth noting that these parameters provide a simple way to implement a more general methodology. For example, a more general approach would allow these parameters to vary depending on the location in the mine.

3.4 Paper contribution

The contributions of this work are the following:

- A mathematical model that improves on previous versions, to generate a discretized pushback which is optimized for value, but such that it is contained within a reference volume and provides enough space for the design of a ramp.
- A pushback design algorithm that utilizes the output of the optimization model plus design parameters, to generate an operational design of the mine.
- The application of the optimization model and algorithm over 15 different instances, and the comparison to designs produced by mine planners, to show that the combined methodology performs as well as the human counterparts.

4 Methodology for automated pushback design

In this section, we formalize with detail the mathematical program utilized to generate the optimized pushback, as well as the pushback design algorithm to produce transform this optimized pushback into an operational pushback.

4.1 A linear program for ramp design in open pit

As indicated in Section 3, a linear program is set on the *pit boundary* to compute the path that represents the ramp and the designed pushback. The input and output of the mathematical model are discretized pushbacks, i.e., they approximate the actual design utilizing blocks from the block models.

The optimization problem will look for a designed pushback of maximum value, for which we assume that each block already has some defined economic value. We also assume that the deepest possible level of the ramp is known in advance. This is the most common case in practice; however, it is not a real constraint, as the model could be used to try different options.

We are given the set of blocks \mathcal{B} with p_b being the net profit of block $b \in \mathcal{B}$ and assume that the levels of the block model and enumerated as $k = 1, 2, \dots, K$, where K is known and corresponds to the bottom and $\mathcal{B}_k \subset \mathcal{B}$ are the blocks at level k . We are also given $B \subset \mathcal{B}$, which corresponds to the blocks in the pit boundary and $R = \{r_1, r_2, \dots, r_K\}$ the set of aggregated blocks per level of the reduced pit, with profits $\hat{p}_k, k = 1, \dots, K$. For convenience, we define $B_k = \mathcal{B}_k \cap B$ (i.e. the blocks in the pit boundary at level k), $\bar{B} = B \cup \{r_k\}_{k=1}^K$.

We consider the following sets of blocks, which are used to model the excavation in terms of the ramp (see Figure 4 for an introduction of these concepts). For each block $b \in B$, we have

1. $P_b \subset B \times \bar{B}$, the set of slope predecessors. To compute this set, we consider, for a slope angle α_b and the 3-D coordinates of block b , (x_b, y_b, z_b) . the *slope precedence cone* of b as $S_b = \{(x, y, z) : z > z_b \wedge \sqrt{(x - x_b)^2 + (y - y_b)^2} / (z - z_b) \leq \tan(\alpha_b)\}$. Thus, we have that
 - For $b' \in B$, $(b, b') \in P_b$ if the coordinates of b' are in S_b , and
 - For r_k (an aggregation), (b, r_k) is an element of P_b if $b \in B_k$ or if there exists a block $b' \in r_k$ and the coordinates of b' belong to S_b .

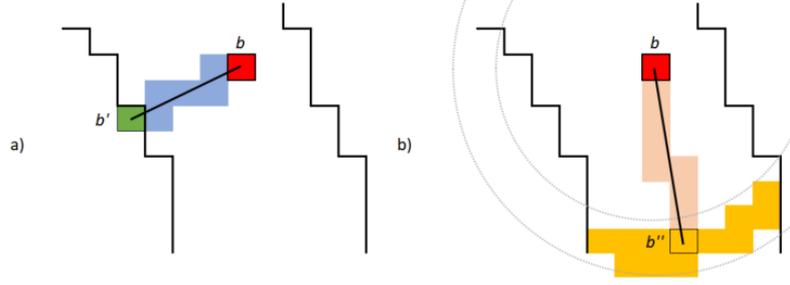


Fig. 5 Plan view of blocks at level k , with pit boundary presented as lines a) The block $b' \in \mathcal{B}_k - B_k$ that is closest to $b \in B_k$, linear segment joining their centers and the set H_b with blocks colored in blue. b) In yellow the set \mathcal{A}_b of blocks that can be reached from b by constructing a ramp and the elementary path $E_{b,b''}$.

2. $H_b \subset B$ the set of predecessors corresponding to the ramp, plus those to prevent railing. If $b \in B_k$, we observe that there is one block $b' \in \mathcal{B}_k - B_k$ which is the closest to b (using Euclidean distance). Then H_b is the set of blocks such that they intersect the segment going from b to b' . For convenience, we also define $H'_b \subset H_b$, which is formed by the blocks that represent the ramp, i.e., the first blocks from b such that they form the best approximation to the width of the ramp. Notice that H_b depends only on block b .
3. $X_b \subset B$ of blocks that cannot be removed if b is an interpolation block of the ramp. To compute X_b , we consider the set of blocks located immediately below H'_b .

Figure 5(a) presents an example of the construction of H_b , where a block $b \in B_k$ is displayed as well as a block $b' \in \mathcal{B}_k - B_k$. b' is the closest block to b (in Euclidean distance), therefore the set H_b is the set of blocks that intersect the segment that goes from the center of b to the center of b' .

We are also given, as input, the directed graph $\mathcal{D} = (B, \mathcal{A})$. The set \mathcal{A} is constructed of pairs of blocks (b, b') such that if $b \in B_k$ (for some k) then $b' \in B_{k+1}$ and $h/d(b, b') \in [\tan(\theta) - \beta, \tan(\theta) + \beta]$, where $d(b, b')$ is the Euclidean distance between the centers of the blocks, h is the height of a bench, θ is the slope of the ramp and β is a tolerance. This definition of \mathcal{A} allows the ramp having *switchbacks*, i.e., for example, to change direction from turning clockwise to anti-clockwise, but potentially it may be also useful to limit \mathcal{A} to arcs to be in one direction only.

We denote $\mathcal{A}_b = \{b' \in B : (b, b') \in \mathcal{A}\}$ as the set of blocks from which it is possible to reach b and $E_{b,b''} \subset B$ the elementary path as the blocks connecting b and b'' , that is $E_{b,b''}$ is the set of blocks such that they intersect the segment between b and the block immediately on top of b'' . Notice that if $b \in B_k$ then $\mathcal{A}_b \subset B_{k+1}$ and $E_{b,b''} \subset B_k$.

An example to understand the construction of \mathcal{A} is presented in Figure 5(b), where a block $b \in B_k$ is displayed in red and the contour of the pit boundary is presented by the thick lines. The set $\mathcal{A}_b \subset B_{k+1}$ corresponds to the blocks painted in yellow, i.e., those such that if a ramp from b to them has a feasible slope. Finally, a block $b'' \in \mathcal{A}_b$ is highlighted and the segment that joins b and b'' is drawn. The elementary path $E_{b,b''}$ is the set of blocks painted in orange, i.e., blocks in B_k that intersect the segment.

The variables in the model are the following:

$$y_b = \begin{cases} 1 & \text{if block } b \text{ is extracted,} \\ 0 & \text{otherwise.} \end{cases}$$

Variable y_b is defined for all $b \in \bar{B}$, i.e. it includes the aggregations of the reduced pit. Then

$$x_{b,b'} = \begin{cases} 1 & \text{if blocks } b \text{ and } b' \text{ are consecutive interpolation blocks of the selected ramp} \\ 0 & \text{otherwise.} \end{cases}$$

Variable $x_{b,b'}$ is defined for all $(b, b') \in \mathcal{A}$ and therefore it is not defined for r_k (aggregations in the reduced pit).

The objective function to be maximized is the total profit of the excavated blocks

$$V = \sum_{b \in B} p_b y_b + \sum_{k=1}^K \sum_{(b,b') \in \mathcal{A}: b' \in B_k} \hat{p}_k x_{b,b'} \quad (1)$$

The constraints of the model are the following

$$y_b \leq y_{b'} \quad \forall b \in B, \forall b' \in P_b \quad (2)$$

$$x_{b,b'} \leq \sum_{b'' \in \mathcal{A}_b} x_{b''} \quad \forall (b, b') \in \mathcal{A} \quad (3)$$

$$x_{b,b'} \leq 1 \quad \forall k = 1, \dots, K, \forall (b, b') \in \mathcal{A}, b' \in B_k \quad (4)$$

$$x_{b,b'} \leq y_{b''} \quad \forall (b, b') \in \mathcal{A}, b'' \in E_{b,b'} \quad (5)$$

$$y_b \leq y_{b'} \quad \forall b \in B, \forall b' \in H_b \quad (6)$$

$$x_b + y_{b'} \leq 1 \quad \forall b \in B, \forall b' \in X_b \quad (7)$$

$$x_{b,b'} \in \{0, 1\} \quad \forall (b, b') \in \mathcal{A} \quad (8)$$

$$y_b \in \{0, 1\} \quad \forall b \in \bar{B} \quad (9)$$

Constraints (2) impose the slope precedences for the shape of the pushback walls. Constraints (3) and (4) force that there is a path and that it is unique. Constraints (5) impose the extraction of blocks in the elementary path between two blocks in the ramp. Constraints (6) and (7) prevent riling and force that the ramp is not in the air.

Some applications may require that the starting point at the surface is fixed in advance. This can be easily achieved by setting one additional constraint that $\sum_{(b,b') \in \mathcal{A}: b_0 \in E_{b,b'}} x_{b,b'} \geq 1$, for the given point block b_0 that is the required starting point.

4.2 Pushback design algorithm

The pushback design algorithm takes the output of the mathematical model and generates an operational design that complies with overall slope angle, inter-ramp angle, ramp width, and slope, as well as bench width and height. Notice that, as such, the algorithm does not generate the system of ramps required for a real mine with multiple pushbacks; however, it can be used iteratively, together with the mathematical program, to address the more general problem.

The input of the algorithm is given by the block model and the interpolation blocks, which are encoded in the variables $x_{b,b'}$, i.e., it does not use the optimized pushback directly.

Some definitions that are relevant for the description of the algorithm are the following: (i) In some parts, the algorithm takes a discretized area (a set of contiguous blocks in 2-D) and generates a smooth area. This is accomplished by selecting the blocks' coordinates in the boundary of the discretized area and then utilizing a moving average of the (x, y) coordinates. We call this process *smoothing*. (ii) For a curve \mathcal{C} in 3-D, and points $p, p' \in \mathcal{C}$, the distance $d_{\mathcal{C}}(p, p')$ between these points, measured on the curve \mathcal{C} .

The algorithm was implemented in Python using the Shapely library (Gillies et al., 2007–) to work with curves and projections, and the MineLink library (Delphos, 2018) to work with blocks. Minelink has been developed by Delphos mine planning laboratory at the University of Chile and can be licensed freely for academic purposes.

1. **2-D reference curve.** In this stage, the coordinates of the interpolation blocks are used to construct a reference curve, which is projected afterwards in 3-D to represent the ramp's center. The curve is 2-D (so we can assume all z coordinates to be equal) and is constructed as follows:

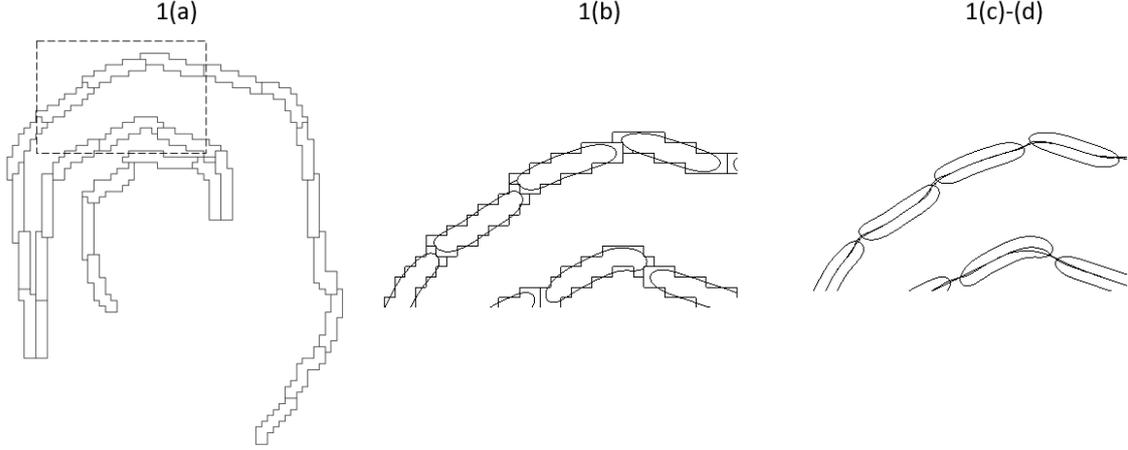


Fig. 6 Step 1 of the pushback design algorithm. First, each level is discretized into sets of blocks. Then these sets are smoothed and connected to generate a reference curve in 2-D.

- (a) For each level k , let H_k be the set of blocks required to be extracted to make room for the ramp, i.e., if arc $(b_1, b_2) \in \mathcal{A}$ is chosen as part of the ramp, then $H_k = H'_{b_1} \cup \bigcup_{b \in E_{b_1, b_2}} H'_b$.
- (b) The contour of H_k is smoothed.
- (c) Inside of the smoothed contour, the reference curve is computed as the median point between the borders, i.e., the curve splits each smoothed contour by two halves, following its longest dimension.
- (d) Outside the smoothed contour, the reference curve is obtained simply by connecting segments corresponding to two consecutive smoothed contours.

Figure 6 illustrates an example of this step, where the sets H_k are displayed for a mine. The figure also zooms in the delineated area to show the smoothing of the sets and the resulting reference curve.

2. **Correction of the 2-D reference curve.** This stage also works in 2-D and corrects the (x, y) coordinates such that the ramp does not intersect with itself. The process works as follows:
 - (a) Using the reference curve and the ramp width, a left curve \mathcal{L} and a right curve \mathcal{R} are constructed, parallel to the reference curve. To fix ideas, assume that the curves start at the bottom of the pit and end at the surface.
 - (b) Beginning with their starting points, points located farther up in the ramp are moved to ensure that they are at a distance at least equal to the ramp's width. The points are moved to coordinates that are closest when moving in the excavation's outer direction; that is, the ramp expands outward, in the opposite direction of the reduced pit.
3. **Projection of the ramp into 3-D.** In this stage, the reference curve's z coordinates are re-assigned: the points are moved to their actual positions in 3-D.
 - (a) First, given a parameter $s > 0$ (a small value, used to discretize the ramp), two sets of points $\{\ell_j\}_j \subset \mathcal{L}$ and $\{\rho_j\}_j \subset \mathcal{R}$ are computed: (i) ℓ_0 (resp. ρ_0) is the first point of \mathcal{L} (resp. \mathcal{R}). (ii) To compute ℓ_j , points $\ell' \in \mathcal{L}$ such that $d_{\mathcal{L}}(\ell_{j-1}, \ell') = s$ and $\rho' \in \mathcal{R}$ such that $d_{\mathcal{R}}(\rho_{j-1}, \rho') = s$ are computed. Then each point is projected in the opposite curve to produce a new point, i.e., ℓ' is projected on \mathcal{R} to generate a point $\rho'' \in \mathcal{R}$ and ρ' is projected on \mathcal{L} to produce a point ℓ'' . Then ℓ_j is the closest point to ℓ_{j-1} between ℓ' and ℓ'' and ρ_j is the point, among ρ' and ρ'' that is closest to ρ_{j-1} .
We observe that, as a result of the previous process, both curves \mathcal{L} and \mathcal{R} are partitioned into segments of a length at most s , however depending on the curvature, segment j would be of different length for each curve.
 - (b) Points γ_j are constructed on the reference curve by intersecting the curve with the segment joining ℓ_j and ρ_j .

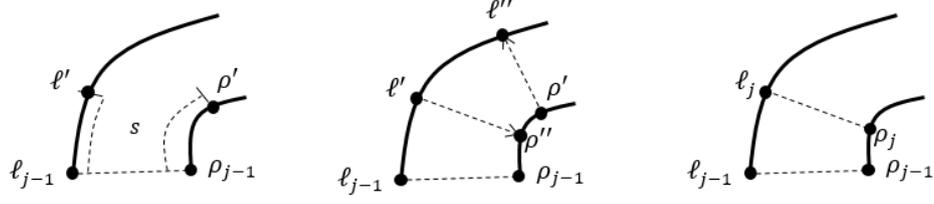


Fig. 7 Step 3(a) of the pushback design algorithm: Calculation of points ℓ_j, ρ_j from previously computed points ℓ_{j-1}, ρ_{j-1}

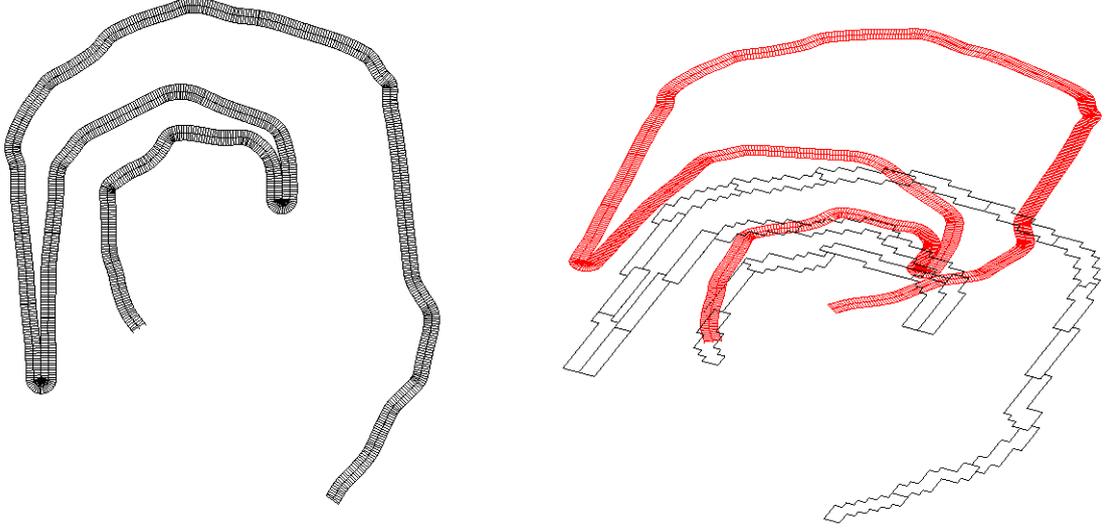


Fig. 8 Step 3(c) of the pushback design algorithm: Segmentation of the ramp (black) and projection on 3-D (red).

- (c) The z coordinates are assigned starting at points corresponding at the bottom and then moving up, as follows (i) The points corresponding to the deepest level are assigned the coordinate of the floor of the deepest level of the ramp. (ii) Assuming that γ_{j-1} as a z coordinate z_{j-1} , then z_j (the coordinate z of γ_j) is calculated as follows:

- If $|d_{\mathcal{L}}(\ell_{j-1}, \ell_j)| < \varepsilon$ or $|d_{\mathcal{L}}(\rho_{j-1}, \rho_j)| < \varepsilon$ (for some tolerance ε), the ramp is turning in a switchback, and therefore it gains no height, thus $z_j = z_{j-1}$.
- Otherwise, $z_j = z_{j-1} + d_{\mathcal{L}}(\gamma_{j-1}, \gamma_j) \cos(\theta)$, where θ is the slope of the ramp.

- (d) The points ℓ_j and ρ_j are assigned a z coordinate equal to z_j .

Figure 7 shows an example of step 3(a) of the algorithm, with two points ℓ_{j-1}, ρ_{j-1} already computed. These points are then used to construct points ℓ', ℓ'', ρ' and ρ'' and finally the next points ℓ_j, ρ_j are selected as the closest in each curve. Figure 8 presents the sets of points $\{\ell_j\}_j, \{\rho_j\}_j$ before (in black) and after (in red) the z coordinates are assigned.

4. **Computation of the designed pushback.** In this stage, the algorithm computes the contour of the designed pushback by utilizing the 3-D ramp of the previous steps. The designed pushback is computed from three volumes $V^{\text{walls}}, V^{\text{sup}}$ and V^{inf} , such the contour of the designed pushback is the contour of $V^{\text{walls}} \cup V^{\text{sup}} \setminus V^{\text{inf}}$. The volumes are computed as follows:

- (a) V^{walls} is the volume corresponding to the pushback walls. To compute this volume, for each level k , the set of blocks extracted at that level is smoothed, producing a curve C_k associated with the floor of that level. Then, V^{walls} is the minimum volume such that it respects the slope angles and contains these curves.

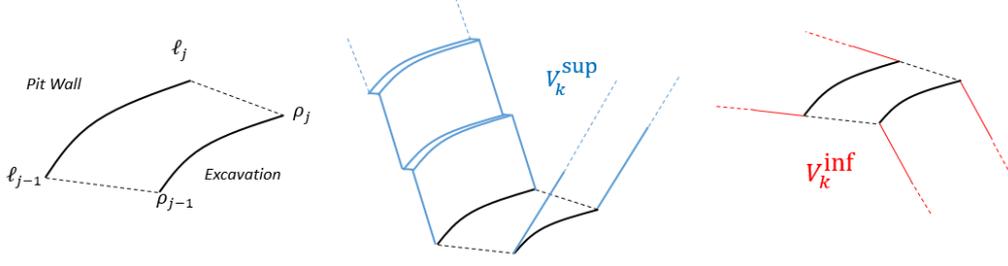


Fig. 9 Volumes to be extracted given a section of the ramp. V_j^{sup} is the projected volume for extraction, and V_j^{inf} is the projected volume to be left unextracted.

- (b) V^{sup} is the minimum volume required to be extracted to expose the ramp. For each j the polygon \mathcal{Q}_j with vertices $\rho_{j-1}, \ell_{j-1}, \ell_j$ and ρ_j , the volume V_j^{sup} required to extract \mathcal{Q} is computed. Then $V^{\text{sup}} = \cup_j V_j^{\text{sup}}$
- (c) V^{sup} is the minimum volume to support the ramp. For this, for each \mathcal{Q}_j and, similarly V_j^{inf} is the minimum volume supporting \mathcal{Q}_j (i.e. that cannot be removed) and $V^{\text{inf}} = \cup_j V_j^{\text{inf}}$. The sets V^{sup} and V_j^{inf} are depicted in Figure 9, which shows the vertices of a polygon \mathcal{Q}_j . The volume to be extracted because of this is V_j^{sup} , with part of its contour in blue, and the volume that cannot be extracted is V_j^{inf} , with its contour delineated in red.

Finally, once the designed pushback is generated, it is exported as a 3-D CAD file that can be loaded in standard design software.

5 Numerical experiments

We used three block models for the experiments: “Marvin30”, “Marvin15” and “Mine3”. The first two correspond to the same deposit but differ in the blocks’ sizes; thus, the total number of blocks and how accurately they can approximate a continuous volume. Using these two blocks models is interesting to understand how the solution behaves in terms of quality and performance when the blocks’ dimensions change.

For each block model, we utilize the methodology to generate several alternative designs (a total of 15 instances overall) to analyze the methodology’s performance. For this, we compare the economic value and tonnage of the original discretized pushback with the economic value and tonnage of the two outputs of the methodology: the optimized pushback and the operational pushback. However, for Mine3, where the methodology has the worst performance, we also compare it with manual designs.

Table 1 summarized the input data for the three models. The table columns are the name of the instance, the total number of blocks, the size of the physical blocks, and the following attributes of the original pushback: the economic value, its total tonnage, and the number of blocks. We observe that the original pushbacks’ economic values correspond to the ultimate pit and are exact, i.e., optimal. However, even though Marvin15 and Marvin30 correspond to the same deposit, there is a small increase in the pit value due to the block size reduction, which allows the ultimate pit to avoid some unprofitable parts.

For all the instances, the original pushback corresponds to a final pit computed with a overall slope angle of 47° , the maximum ramp slope is of 10%, and the ramp has to be 30 meters wide as a minimum, which means that Marvin30, Marvin15, and Mine3 required 1, 2 and 3 blocks to represent the width of the ramp, respectively.

The first useful application of the model comes from the fact that it can quickly determine the number of turns that the ramp performs on a given wall, which is essential because the final overall slope angle depends on that angle. In Marvin30 (and Marvin15), there were two ramp

Table 1 Block models input for numerical experiments and values of the original pushbacks.

Instance name	Block Model		Original Pushback		
	# Blocks	Block Size	Value MUSD	Tonnage MT	Blocks #
Marvin30	53,271	$30 \times 30 \times 30$	1,405	543.5	8,840
Marvin15	426,168	$15 \times 15 \times 15$	1,442	516.4	66,433
Mine3	607,200	$10 \times 10 \times 10$	246	60.8	23,371

turns in all walls (there is only one overall slope angle for the whole mine, which corresponded to the overall slope angle used to compute the ultimate pit, i.e., 47°). For some Mine3 test cases, two or three ramps appeared depending; thus, two different overall slope angles were used to keep the inter-ramp angle of 53° . For the cases with two ramps turns, the overall slope angle was 45° . For the cases with three ramp turns, this corresponds to an overall slope angle of 42° .

It is worth noting that having a constant overall angle is not entirely realistic. Indeed, different geomechanical conditions may translate into different slope angles for different zones of the mine. Considering this is not a constraint for the mathematical program; however, our current implementation of the pushback design algorithm only supports a constant angle.

The MineLink library was used to develop a C++ code that generates the input data sets and implement the mathematical model given by (1)-(9). The library was utilized to manage the data, compute precedences and calculate the original pushbacks.

For all ramp design computations, the mathematical model was implemented and solved using Gurobi 6.5 Gurobi (2018). The code was implemented in C++ and used the Gurobi API. All experiments were run on a PC with 10 CPU Xeon E5 2660 v3 (2.60 GHz) with 120 GB of RAM under a Microsoft Windows 7 environment. Except for the relative optimality gap, that we set to 1.5% in all instances, only default Gurobi parameters were used. The ramp pushback design algorithm was implemented using the Python language and the Shapely library.

5.1 Numerical results

In this section, we report the results obtained when running the optimization model on the 15 instances utilized in this study. We report the following information (Tables 2, 3 and 4): The instance number (“Instance #”), the inner and outer boundary sizes (“ δ^- ” and “ δ^+ ”), the number of variables (“# Vars.”) and the constraints (“# Constr.”, in millions) in the optimization model, the optimality gap (“Gap %”) reported by the solver, the execution time (Time), the difference (%) in terms of economic value (“ Δ Opt. P. Values”) and tonnage (“ Δ Opt. P. Tons”) between the optimized and original pushbacks, and similarly the difference (%) in economic value and tonnage between the operational pushback and the original pushback (“ Δ Oper. P. Value” and “ Δ Oper. P. Tons”, respectively).

Notice that the time reported in the tables corresponds to the execution of the solver for the mathematical program and does not consider the time required by the pushback design algorithm. The reason for this is that the execution of the algorithm takes fewer than 15 seconds, therefore, it is not relevant for the analysis and negligible for the application.

For all instances, we asked a group of users, with different degrees of experience, to elaborate a design of the pit manually. To keep the results as objective as possible, these planners did not have access to the output of the methodology. Instead, they used the block model, the original discretized pushback, and a mine design software to produce their designs.

Marvin30 dataset

Table 2 presents the results for five design instances applied to Marvin30, for different sizes of the inner and outer parts of the boundary.

Table 2 Results of the methodology applied on different instances based on Marvin30.

Instance #	δ^-	δ^+	# Vars.	# Constr. ($\times 10^6$)	Gap [%]	Time [s]	Δ Opt. P. Values [%]	Δ Opt. P. Tons [%]	Δ Oper. P. Value [%]	Δ Oper. P. Tons [%]
1	2	0	73,891	2.6	0.28	185	1.6%	5.6%	0.3%	5.7%
2	2	1	92,802	3.7	0.14	304	3.2%	-3.8%	0.6%	-5.5%
3	1	1	69,898	1.7	0.21	47	3.1%	-4.5%	-0.6%	-6.4%
4	1	2	90,650	3.5	0.28	182	3.2%	-7.8%	0.0%	-9.8%
5	0	2	68,761	1.7	0.00	36	3.2%	-10.4%	-0.6%	-13.2%

We observe from the results that the methodology always generated operational designs with an economic value very close to that of the original pushback ($< 1\%$ difference). That is, a mining engineer could choose any of the solutions proposed and be sure to retrieve most of the economic value of the original pit.

A second interesting result is that the optimized pushbacks' economic values are higher than those of the original ones. This is significant because practitioners regard the ultimate pit's value as an ideal, unattainable, maximum value. One possible reason for this is that the inter-ramp angle is greater than the overall one; therefore, the predecessors of blocks are different from those of the final pit computation. Consequently, the solution considering the ramp had the flexibility to avoid extraction of some low-grade areas and recover a slightly higher amount of mineral that led to a better economic value (up to $+ 3.2\%$). However, given the large size of the blocks and approximations done by the model, the operational design cannot retrieve this consistent increment in value and instead produces an operational pushback with a value that is almost the same as the ultimate pit. These results are most likely due to the large size of the blocks.

Surprisingly, when comparing the weight (tonnage) of the optimized pushback with the original one, the tonnage decreases as the pit boundary is moved outside the original pushback (i.e., larger values of δ^+ and smaller values of δ^-), which in turn has a significant impact on the economic value. However, while the tonnage of the operational pushback follows a similar trend, its economic values remain close to the ultimate pit: the pushback design algorithm could keep the economic value despite being forced to extract blocks outside the ultimate pit.

It is also worth noting that the computational times were less than 5 minutes; however, these instances have a fixed starting point. Indeed, when the starting point is free, only the case $\delta^- = 0, \delta^+ = 2$ could be solved within a 1-day execution time. In this instance, the execution time was 933 seconds, the gap was 0.49%, and the increment in economic value and tonnage were 1.3% and 4.4%, respectively. However, we do not regard the free starting point's complexity as a big issue, as several potential candidates could be evaluated quickly enough.

For the instances reported in Table 2, the solution of the root node objective value provides a very good bound, as it is very close to the integer objective value obtained by the solver (from 0.003% to $+1.0\%$). This result could be useful, for example, when looking for strategies for improving the execution times, for example, using an aggressive cut-off method. See, for example, Klotz and Newman (2013).

When compared to the manual designs, the differences in value and tonnage were negligible, i.e., the algorithm performed equivalently to the best manual design generated by the users.

Marvin15 dataset

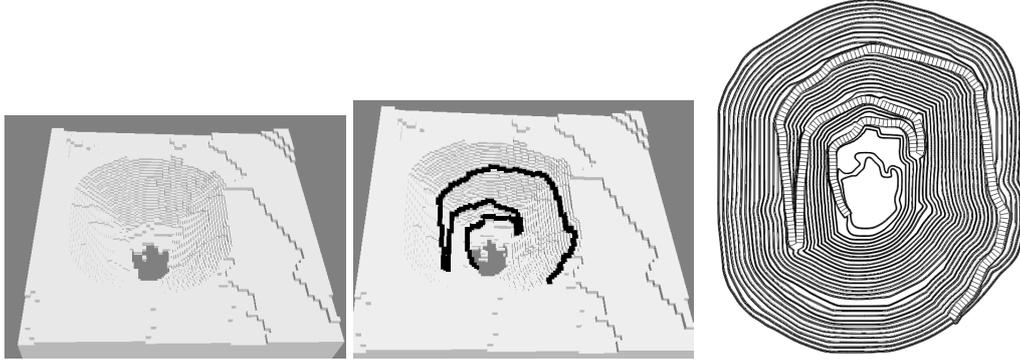
Table 3 presents the results in the four design instances based on the block model Marvin15, which were obtained for different values of δ^- and δ^+ .

As expected, because the size of Marvin15 instances is larger than the Marvin30 ones, the computing time required to find the solutions increased significantly, even if the starting point is fixed in advance, up to about 5 hours in the worst case, which is about the limit for practical applications.

When compared to the case of Marvin30, we observe that for Marvin15, the differences between economic values and tonnages of the optimized and original pushbacks are smaller, most likely

Table 3 Results of the methodology applied on instances based on Marvin15.

Instance #	δ^-	δ^+	# Vars.	# Constr. ($\times 10^6$)	Gap [%]	Time [s]	Δ Opt. P. Values [%]	Δ Opt. P. Tons [%]	Δ Oper. P. Value [%]	Δ Oper. P. Tons [%]
6	2	0	513,136	24.9	0.49	11,541	-0.3%	5.1%	-1.6%	3.8%
7	1	1	496,158	21.1	0.49	15,116	-0.1%	-0.4%	-1.9%	-2.0%
8	1	2	538,179	24.6	0.42	18,499	-0.5%	-2.1%	-1.9%	-2.7%
9	0	2	490,560	19.4	0.21	2,941	-1.0%	-6.4%	-3.4%	-8.6%

**Fig. 10** Example of the original pushback, designed pushback and designed phase.

because when using smaller block sizes, the approximation of ramp shape and the slopes becomes more accurate. On the other hand, the optimized pushbacks no longer reach values higher than the ultimate pit. In any case, the difference in value is at most within 3.4%, which is very reasonable for a design computed by an algorithm. Further on, this happens for a value $\delta^- = 0$, which implies that if the solution reaches some bench, it has to extract all blocks in the ultimate pit corresponding to that bench. Finally, as in Marvin30, the tonnages of the outputs (optimized and operational pushbacks) decrease as δ^- is smaller, i.e., when the pit boundary is outside the ultimate pit. However, when $\delta^- > 0$, all values are at most 2% far from the original pushback, hence provide acceptable results from where an engineer may choose. This latter observation suggests using blocks as small as possible because they approximate volumes more accurately.

An example of the original pushback, the optimized pushback and the operational pushback obtained using the pushback design algorithm can be found in Figure 10.

As in the case of Marvin15, the difference in tonnage and value between the designs produced by the methodology and those generated manually was marginal.

Mine3 dataset

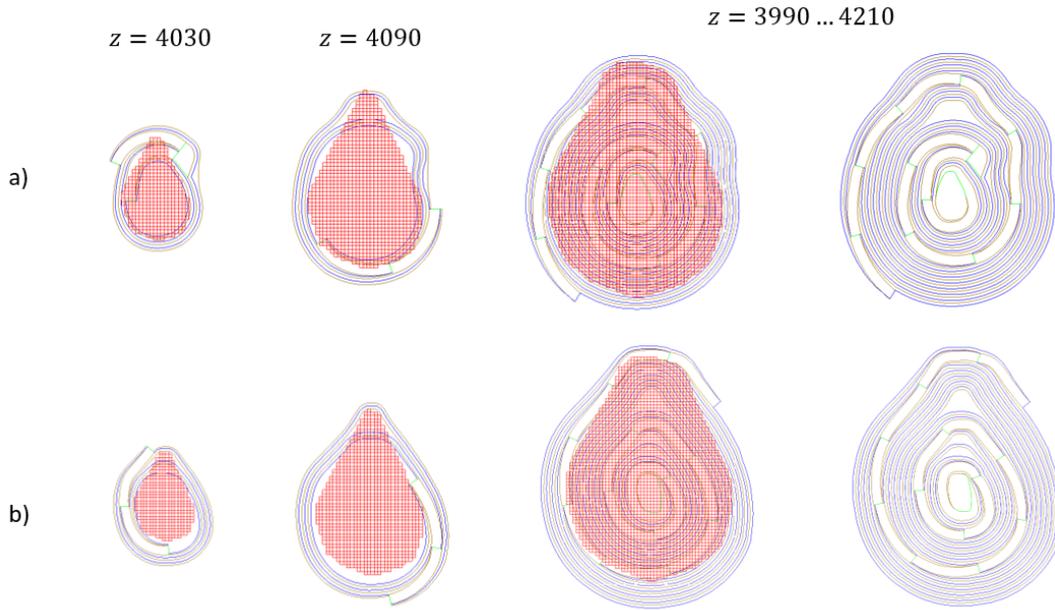
Table 4 presents the results for six instances based on the block model Mine3. In this case, all instances have values $\delta^- = 2, \delta^+ = 1$ and therefore these parameters are not reported in the table. On the other hand, the geometry of this case was more complex, because the number of turns of the ramps was different depending on the wall and the case. Because of this, different global slope angles had to be utilized at different sectors of the mine. The cases had either one or two global slope angles, which are in column “Overall angle”.

Economic values of ramp design for the Mine3 instances were from - 9.3% to - 7.3% comparatively to the one of the original pushback. This relatively large difference was that the original pushback did not have enough space for ramps to reach its bottom, and therefore the tonnage was similar, but there was a loss in value.

In this case we also compared the results obtained by the methodology to manual designs; however, for Mine3 the analysis is more interesting because of the significant reduction in the

Table 4 Results of the methodology applied on different instances based on Mine3.

Inst. #	Overall Angle [°]	# Vars.	# Constr. ($\times 10^6$)	Gap [%]	Time [s]	Δ Opt. P. Values [%]	Δ Opt. P. Tons [%]	Δ Oper. P. Value [%]	Δ Oper. P. Tons [%]
10	45	80,332	5.5	0.44	28,888	-8.5%	-2.3%	-13.8%	-4.6%
11	45 & 42	80,201	5.4	1.32	5,408	-7.7%	-3.1%	-11.4%	-5.1%
12	45	80,332	5.5	1.35	5,241	-9.3%	-4.9%	-13.4%	-7.2%
13	45 & 42	80,201	5.4	1.32	8,007	-7.3%	-2.6%	-11.0%	-4.1%
14	45 & 42	80,332	5.5	1.32	7,047	-7.7%	-3.9%	-11.8%	-6.4%
15	45 & 42	80,201	5.4	1.34	3,294	-8.9%	-1.8%	-12.6%	-3.9%

**Fig. 11** Plan view of two manual designs and the original pushback for the Mine3 case. a) Design with a switchback. b) Design without switchback.

pushbacks' economic values proposed by the methodology; hence, we created four different manual operational designs for a comparison.

Figure 11 presents the two best manual designs for Mine3. In this case, the planners' strategy was to increase the pit's size to access the deepest levels, but as it turns out, that required extracting additional waste. Indeed, the values for both designs are the following. The first design (top of the figure) corresponds to a ramp that contains switchback: it had an increase of 11.0% in tonnage and a loss of 17.7% in value. The second design (bottom of the figure) had no switchback, but an increase of 11.6% in tonnage with a 12.6% loss in economic value. As it turns out, both cases performed similarly to the methodology.

It follows that all the studied instances, the methodology proposed in this work performs very similar to a human operator: If elaborating an operational design of the mine is possible for a human planner, the methodology is also capable of finding one, with similar values. Contrarily, if the initial pushback is hard to be approximated, both human and computerized methods produce designs that deviate from the original tonnages and economic values; however, they are similar to each other. We consider this a very significant result, which points in the direction of looking for a more robust methodology to generate and select the initial discretized pushback. At the minimum, the automated methodology would suggest the planner to change the selected pushback for design.

At the best, an integrated methodology for pushback selection and ramp design could be developed to eliminate this issue.

6 Conclusions

In this paper, we have presented a methodology capable of automatically generating an operational design, including ramps, for an open-pit mine. The methodology consists of two stages. First, an optimization model takes a discretized pushback as a reference and produces another discretized pushback, but optimized for value and having space to accommodate a ramp. Second, a pushback design algorithm that fits an operational pushback design close to the optimized one. The methodology was then implemented and applied to three block models and evaluated over fifteen experiments.

The numerical results show that the methodology generates high-value designs close in tonnage and shape to the original pushback. Indeed, for two of the three cases, the methodology proposed designs with an economic value reduction within 2% or less and at most 6% variation in tonnage. In the third case, the loss of economic value was significant (more than 10%) but very similar to the one obtained by a manual design.

The results are promising, yet they leave room for improvement because they suggest that smaller blocks produce better results, but this happens at the expense of a sharp increase in the optimization model's computational time. These two observations motivate the research of a more efficient algorithmic approach to solve the optimization problem. We observe that the root node continuous objective value seems to be near the optimum integer objective value, and this information may be used to accelerate the computation, such as doing an aggressive cutoff. Other approaches to handle this problem can be to use meta-heuristics as in Mousavi et al. (2016), local search or hybrid techniques (Behnamian, 2017).

References

- Altiti A, Alrawashdeh R, Alnawafleh H (2021) Open Pit Mining. DOI 10.5772/intechopen.92208
- Asad M, Topal E (2011) Net present value maximization model for optimum cut-off grade policy of open pit mining operations. *Journal of the Southern African Institute of Mining and Metallurgy* 111:741 – 750
- Bai X, Marcotte D, Gamache M, Gregory D, Lapworth A (2018) Automatic generation of feasible mining pushbacks for open pit strategic planning. *The Journal of The Southern African Institute of Mining and Metallurgy* 118(5):515–530
- Behnamian J (2017) Matheuristic for decentralized factories scheduling problem. *Applied Mathematical Modelling*
- Bienstock D, Zuckerberg M (2009) A new LP algorithm for precedence constrained production scheduling. *Optimization Online*
- Bienstock D, Zuckerberg M (2010) Solving LP relaxations of large-scale precedence constrained problems. In: *IPCO*, pp 1–14
- Brazil M, Thomas DA (2007) Network optimization for the design of underground mines. *Networks* 49:40–50
- Brazil M, Thomas DA, Weng JF, Rubinstein JH, Lee DH (2005) Cost optimisation for underground mining networks. *Optimization and Engineering* 6(2):241–256
- Brazil M, Grossman PA, Lee DH, Rubinstein JH, Thomas DA, Wormald NC (2007) Constrained path optimisation for underground mine layout. In: *World Congress on Engineering*, pp 856–861
- Cochilco (2017) Mining in chile: Future and challenges. <http://www.cochilco.cl>, accessed 30 May 2017
- Couzens T (1979) Aspects of Production Planning: Operating Layout and Phase Plans, AGARD Report, AIME, pp 217–231

- Cullenbine C, Wood R, Newman A (2011) A sliding time window heuristic for open pit mine block sequencing. *Optimization Letters* 5(3):365–377
- Dagdelen K (2001) Open pit optimization-strategies for improving economics of mining projects through mine planning. In: 17th International Mining Congress and Exhibition of Turkey, pp 117–121
- Delphos (2018) Minelink M.P.L. <http://www.delphoslab.cl>, accessed 18 July 2018
- Dowd P, Onur A (1992) Optimising open pit design and sequencing. In: Proc. 23rd International Symposium of Application of Computers and Operations Research, pp 411–422
- Dubins LE (1957) On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics* 79:497–516
- Espejo N, Morales N, Nancel-Penard P (2019) A methodology for automatic ramp design in open pit mines. *Journal of Mining Engineering and Research* 1(2):87–93
- Espinoza D, Goycoolea M, Moreno E, Newman A (2013) Minelib: a library of open pit mining problems. *Annals of Operations Research* 206(1):93–114
- Gemcom (2018) Gemcom Whittle™ Strategic Mine Planning software. <http://www.gemcomsoftware.com/products/whittle>, accessed 18 July 2018
- Gilani SO, Sattarvand J (2015) A new heuristic non-linear approach for modeling the variable slope angles in open pit mine planning algorithms. *Acta Montanistica Slovaca* 20(2):251–259
- Gill T (1999) Express road routing: The application of an optimal haul road generator to real world data. *Optimizing with Whittle Whittle Programming Pty Ltd, Perth, Western Australia* pp 71–80
- Gillies S, et al. (2007–) Shapely: manipulation and analysis of geometric objects. URL <https://github.com/Toblerity/Shapely>
- Gurobi (2018) Gurobi™ Gurobi Optimization. <http://www.gurobi.com>, accessed 18 July 2018
- Hartman HL, Mutmanský JM (2002) *Introductory Mining Engineering*, 1st edn. John Wiley and Sons
- Hochbaum DS (2008) The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research* 56:992–1009
- Hochbaum DS, Chen A (2000) Performance analysis and best implementations of old and new algorithms for the open-pit mining problem. *Operations Research* 48:894–914
- Hochbaum DS, Orlin JB (2012) Simplifications and speedups of the pseudoflow algorithm. *Networks* 61:40–57
- Hustrulid W, Kuchta M, Martin R (2013) *Open Pit Mine Planning and Design, Two Volume Set & CD-ROM Pack*. CRC Press, URL <https://books.google.cl/books?id=3XTOBQAAQBAJ>
- Johnson T (1968) Optimum open-pit mine production scheduling. PhD thesis, Operations Research Department, University of California, Berkeley
- Johnson T (1969) *A Decade Of Digital Computing In The Mineral Industry*, chap Optimum open-pit mine production scheduling, pp 539–562
- Jélvez E, Morales N, Nancel-Penard P, Peypouquet J, Reyes P (2016) Aggregation heuristic for the open-pit block scheduling problem. *European Journal Of Operational Research* 249(3):1169–1177
- Khalokakaie R, Dowd PA, Fowell RJ (2000) Lerchs–grossmann algorithm with variable slope angles. *Mining Technology* 109(2):77–85, <https://doi.org/10.1179/mnt.2000.109.2.77>
- Klotz E, Newman AM (2013) Practical guidelines for solving difficult mixed integer linear programs. *Surveys in Operations Research and Management Science* 18(1–2):18 – 32
- Lambert W, Brickey A, Newman A, Eureka K (2014) Open-pit block-sequencing formulations: A tutorial. *Interfaces* 44(2):127–142
- Lamghari A, Dimitrakopoulos R, Ferland JA (2014) A hybrid method based on linear programming and variable neighborhood descent for scheduling production in open-pit mines. *Journal of Global Optimization* 63(3):555–582
- Lerchs H, Grossman H (1965) Optimal design of open-pit mines. *Transactions CIM* 58:47–54
- Letelier OR, Espinoza D, Goycoolea M, Moreno E, noz GM (2020) Production scheduling for strategic open pit mine planning: A mixed-integer programming approach. *Operations Research*

- Li F, Klette R (2007) Rubberband algorithms for solving various 2d or 3d shortest path problems. In: ICCTA, pp 9–19
- Liu S, Kozan E (2016) New graph-based algorithm to efficiently solve large scale open pit mining optimization problems. *Expert Systems With Applications* 43:59–65
- Montané S, Nancel-Penard P, Morales N (2019) Optimization and sequencing a semiautomated ramp design in underground mining: a case study. In: *Proceedings of the 28th Symposium on Mineplanning and Equipment Selection*, pp 139–145
- Morales N, Nancel-Penard P, Parra A (2017) An integer linear programming model for optimising open pit ramp design. In: *Proceedings of the 38th International Symposium APCOM proceedings*, pp 9–16
- Moreno R, Reyes-Jara M, Nancel-Penard P (2020) A comparison between open-pit ramp design obtained by varying design characteristics and through linear optimization. In: *Proceeding of 8th Mass Mining Conference*, pp 1442–1450
- Mousavi A, Kozan E, Liu SQ (2016) Comparative analysis of three metaheuristics for short-term open pit block sequencing. *Journal of Heuristics* 22(3):301–329
- Nancel-Penard P, Morales N (2021) Optimizing pushback design considering minimum mining width for open pit strategic planning. *Engineering and Optimization* To appear.
- Nancel-Penard P, Parra A, Morales N, Díaz C, Widzyk-Capehart E (2019) Value-optimal design of ramps in open-pit mining. *Archives of Mining Sciences* 64(2):399–413
- Picard J (1976) Maximal closure of a graph and applications to combinatorial problems. *Management Science* 22:1268–1272
- Read J, Stacey P (2009) Guidelines for Open Pit Slope Design. DOI 10.1071/9780643101104
- Samavati M, Essam D, Nehring M, Sarker R (2017) A local branching heuristic for the open pit mine production scheduling problem. *European Journal of Operational Research* 257(1):261–271
- Sattarvand J, Shisvan M (2012) Modelling of accurate variable slope angles in open-pit mine design using spline interpolation. *Archives of Mining Sciences* 57(4):921–932
- Sussmann HJ (1995) Shortest 3-dimensional paths with a prescribed curvature bound. In: *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol 4, pp 3306–3312
- Tabesh M, Mieth C, Askari Nasab H (2014) A multi-step approach to long-term open-pit production planning. *International Journal of Mining and Mineral Engineering* 5:273–298
- Thompson R, Visser A (1999) Management of unpaved road networks on opencast mines. *Transportation Research Record Journal of the Transportation Research Board* 1652:217–224
- Yardimci AG, Karpuz C (2017) Optimized path planning in underground mine ramp design using genetic algorithm. In: *MPES, Division of Operation and Maintenance Engineering*, Luleå University of Technology, pp 247–252
- Yarmuch JL, Brazil M, Rubinstein H, Thomas DA (2020) Optimum ramp design in open pit mines. *Computers & Operations Research* 115:104739
- Yarmuch JL, Brazil M, Rubinstein H, Thomas DA (2021) A mathematical model for mineable pushback designs. *International Journal of Mining, Reclamation and Environment*