Optimising the ramp design in an open pit mine

I. Kaabachi, F. Meunier, and N. Morales

Universidad de Chile, Chile

In open pit mining, ore and waste are hauled outside of the excavation area by trucks that move along a system of ramps. The design of these ramps must be carefully addressed because they have to comply with geomechanical constraints to ensure the stability of the pit walls, provide access to the different levels and phases, but also maximise the value of the extracted ore and reduce the extraction of waste.

Because an optimisation approach able to deal with the design of the full ramp system seems to be out of reach at the moment, this paper addresses the problem of designing one ramp providing access to a single pushback, which has been determined beforehand. The optimisation objective is therefore to find a ramp that gives access to one given potential volume, while minimising its impact on the economic value of the volume.

Two versions of the problem are formalised. One version allows the extraction of ore and waste only if this extraction is necessary for the ramp system. This version somehow minimises the intrinsic value of the ramp system. In the second version, it is possible to extract more than is needed by the ramp system. For instance, this takes into account the possibility of attaining ore on the wall opposite to the ramp as soon as the ramp reaches its level.

In the setting mentioned above, the contribution of this work is three-fold. First, the computational complexity of the two different versions of the problem is presented and discussed; in particular, they are shown to be NP-hard. Second, compact integer programs modelling these two versions are proposed. Third, an efficient GRASP algorithm able to cope with the large instances of the problem is proposed. To assess the proposed approaches, the models and algorithms are applied to real instances with an off-the-shelf solver, illustrating and discussing their potential and relevance for the two versions of the problem. They are also compared with other approaches from the literature in terms of value and resulting designs.

INTRODUCTION

Strategic mine planning is a key step in the conception of an operation that largely determines the economic value of a mining project. One of the key aspects of strategic planning is the determination of the mine design.

In the case of open-pit mining, mine design starts with reference pushbacks that are obtained by the application of optimisation models, and algorithms that look to maximise economic value. Because these methods rely on the block model for their computation, the reference pushbacks that they produce as output are discrete, i.e., they are sets of blocks. Moreover, in general, the only design parameter that these models and algorithms consider is the global slope angle. In particular, they do not take into account the space required for equipment to operate or move between levels and sectors of the mine, ramp parameters (like width or decline), inter-ramp angles, bench width, etc.

APCOM 2021
Online via Zoom, 30 August – 1 September 2021
The Southern African Institute of Mining and Metallurgy

In fact, despite its significant relevance for the value of a project, the process of designing an open-pit mine from the reference pushbacks is mostly manual; there are neither algorithmic or mathematical models to solve this problem, nor standardised methodology based on numerical approaches to address it. On the contrary, the design of an open-pit mine is mostly an iterative, trial and error, process where the planner utilises specialised computer-aided design software to generate a feasible design while trying to follow the reference pushbacks as closely as possible. Consequently, the overall procedure can take a long time and its results depend substantially on the expertise of the user.

The need to count on optimisation methods to support the design of mines has been acknowledged by several authors, who have addressed the issue in different ways.

The question of generating discrete pushbacks with additional geometrical properties to account for space for the operation of equipment has been addressed by (Bai *et al*. 2018), who propose a set of "nice geometrical properties" (continuity, smoothness) and geometric operators that act on the reference pushback to look for these properties. (Tabesh, Mieth, and Askari-Nasab, 2014) developed a mathematical model to generate pushbacks. They combined the model with a heuristic approach, and post-processing techniques to obtain practicable forms for the pits. Finally, (Loor and Morales 2020) proposed a genetic algorithm approach to generate nested pits with operational bottom space, and used it to generate real designs, which compare favourably to manual ones in terms of net present value.

In underground mining, (Brazil, Thomas, and Weng, 2005) and (Brazil and Thomas, 2007) address the problem by assimilating the access ramp to a curve consisting of segments and circumference sections, and optimise its length, subject to a fixed curvature and gradient, given starting and end points, and zones that the ramp cannot go through. Based on these developments, (Yardimci and Karpuz, 2017) added minimum curvature constraints and applied a genetic algorithm to minimise the overall underground ramp's cost.

In open-pit mining, the ramp optimisation has not been studied as much as in underground mining. (Yarmuch *et al*. 2020) address the problem of ramp design inside and outside the pit, focusing on the length of the ramp, which is associated with transportation costs. For this, outside the pit, they model the problem as a shortest-path problem, while for the ramp inside the pit they propose a mathematical program and develop a heuristic to solve it. They apply their approach in a mine operation and show that their solutions compare favourably with regards to a manual design.

The work that is closer to the one proposed here is by (Morales, Nancel-Penard, and Parra 2017). Taking a reference discrete pushback as input that does not have enough space for a ramp, their approach produces via integer linear programming (ILP) a new discrete pushback near the reference one, but this time with space for the ramp. Moreover, their approach aims at maximising the economic value, which is more general than minimising transportation costs. This model was then used in a paper by (Nancel-Penard *et al*. 2019), where it was applied in a case study to show that the output can be used as a guide to generate designs which compare well against manual ones.

In this paper, the same problem is addressed, i.e., finding a discrete pushback with space for the design of ramps, such that the resulting pushback is close to a reference one and it contains a maximum value. However, the focus here is on different aspects. First, two different versions of the problem are presented (each with a distinct mathematical formulation). After a proof of its NP-hardness (which means that most likely there is no polynomial algorithm to find an optimal solution), a greedy randomised adaptive search procedure (GRASP) is proposed to find solutions to the problem. Next, the algorithm is applied in five instances, based on the Marvin block model; the solutions found using this approach generate pits which are very close in value to the original one.

STATEMENT OF THE MINING PROBLEM

In this paper, it is assumed that the reference pushback has already been determined. The question under study is the trajectory of the ramp through the set of blocks of the open mine that do not belong to the pushback. The goal is to determine a ramp of maximal value, that gives access to every level of the pushback, and that satisfies geomechanical constraints, e.g., that the slope of the ramp is not too high or that the ramp is large enough. The value of the ramp is the sum of the values of the blocks extracted for its construction. This value is the main criterion: it does not include the value of the reference pushback determined beforehand (and whose value is much bigger by several orders of magnitude).

The main decision of the mathematical problem is to determine, for each level, one block through which the ramp goes. These blocks are called interpolation blocks. The selection of the interpolation blocks must consider the design parameters of the pit. In the modelling, the overall slope angle, the ramp's width and decline, and the inter-ramp angle are considered. (All these values may change between zones of the orebody.) These parameters are an input to the problem; however, they are not considered explicitly in the formulation, but rather defined by certain relations between the blocks, which are now briefly described. (See Section 3 for detailed definitions.)

- The interpolation blocks define a path that goes from the surface, down to the bottom of the pit. This is used to model the decline of the ramp by requiring that two consecutive interpolation blocks must be at a sufficient distance, so that the segment between the interpolation blocks has a decline within a given range.
- The overall wall slope angle is used to determine a set of blocks that need to be extracted to ensure the wall stability. This is modelled using precedences between blocks, as in the case of other standard problems (for example, the ultimate pit), i.e., for each extracted block there is a set of predecessors that need to be extracted.
- Two consecutive interpolation blocks define a set of blocks that must be extracted when they are selected to create the ramp; these are the blocks between them. Moreover, the set of these blocks has a certain width that represents the ramp width. In this sense, it is assumed that the interpolation blocks always correspond to the part of the ramp that is next to the pit wall (i.e., that belong to the outer part of the ramp). This is modelled also using precedences, but this time between pairs of interpolation blocks and blocks. That is, for any pair (b, b') of consecutive interpolation blocks, there is a set of predecessors. Furthermore, because the ramp may be far from the inner part of the reference pit, the predecessors of a pair (b, b') may consider extra blocks to make sure that the ramp does not depart from the pit wall.
- Finally, if a pair (b, b') of interpolation blocks is chosen, the blocks located just below them cannot be extracted. This limits the places where the ramp can go in lower levels, thus ensuring that the ramp is not floating.

Some of the concepts described above are depicted in Figure 1, where a section view of a block model is presented. The blocks are represented by squares, and the set V that contains the boundary of the pit to be calculated is delineated. An interpolation block is painted in black. The sets of blocks to be extracted due to this are coloured in light blue. This set is composed of blocks located above the black block, which must be extracted due to slope precedences, and blocks located at the same level, which have to be extracted to make space for the ramp and the pit itself. Finally, some blocks are coloured in light green to indicate that they cannot be removed, because they became the floor of the ramp.

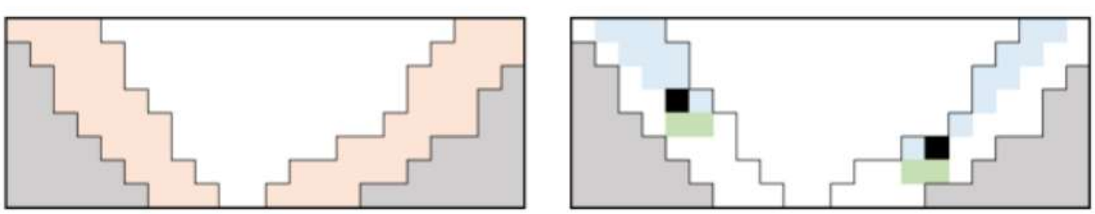

Figure 1. Side view of a block model (frame). On the left, in white, blocks not considered in the optimisation problem, the boundary (orange) and blocks that cannot be extracted in grey. On the right, two interpolation blocks (black), blocks that must be removed (light blue) and blocks that must stay (light green).

MATHEMATICAL FORMALISATION

As indicated before, two versions of the problem are considered. In both versions there are two directed graphs (also called digraphs). A first digraph, denoted by $D_1 = (V, P)$, is used to represent the precedence constraints: $(b, b') \in P$ means that to extract block b it is necessary to first extract block b'. A second digraph, denoted by $D_2 = (V', R)$, with $V' = V \cup \{s, t\}$, is used to represent the selection of interpolation blocks, i.e., $(b, b') \in R$ means that it is possible to construct a ramp from b to b' while keeping its decline within an acceptable range.

Given an arc $a = (b, b') \in R$, two sets are considered: $L_a \subseteq V$ is the set of blocks that need to be extracted if the segment between b and b' is part of the ramp, and $S_a \subseteq V$ is the set of blocks that must remain unextracted if the segment between b and b' is part of the ramp. A valid $s - t$ path is an $s - t$ path p in D_2 such that there exists a subset $X(p) \subseteq V$ satisfying:

- $S_a \cap X(p) = \emptyset$ for all $a \in arcs(p)$.
- $L_a \subseteq X(p)$ for all $a \in arcs(p)$.
- For all $b \in X(p)$ and $(b, b') \in R$, the inclusion $b' \in X(p)$ holds.

The set $X(p)$ corresponds to the set of blocks that must be extracted for building the ramp encoded by the path p.

Each block $b \in V$ has a value, denoted by r_b, which can be positive or negative. The value of a valid $s - t$ path p is equal to $\sum_{b \in X(p)} r_b$. The goal is to determine an $s - t$ valid path of maximum value.

In the first version of the problem, the set $X(p)$ is moreover constrained to be the minimal one for inclusion, i.e., the smallest pit defined by the path p is sought (it is easy to check that such a minimal set exists).

In the second version, each block has a *level*. The set $X(p)$ is allowed to contain every block situated at a level reached by the ramp, but when a block is extracted, there must be a horizontal access to that block from the ramp.

COMPUTATIONAL COMPLEXITY

The main message of this section is that, with the current modelling, the problem is "NP-hard," namely that there is no hope of finding an algorithm to optimally solve the problem, and always running with short computing times. This result is expressed as a formal mathematical statement, because it helps understand exactly under which conditions the claimed limitation holds, and what could be done to get rid of them. For instance, the question whether it remains NP-hard for a modelling, taking the geometric aspect of the problem more closely into account remains open.

Moreover, a complete proof of this result is provided. This is done for the sake of completeness; the proof has absolutely no practical interest and cannot lead to any improvement of the methods used to solve the problem.

Proposition 1. *The problem (in its two versions) is NP-hard, even if every set L_a has a block not shared with any other $L_{a'}$ $(a \neq a')$.*

Proof. The proof consists in reducing Set Cover to it. Given a ground set U, a collection F of subsets of U, and an integer k, the problem Set Cover consists of deciding whether there exists $F' \subseteq F$ with $|F'| = k$ and $\cup F' = U$.

Define then D_1 and D_2 as follows. Let V be $\{1,2, \dots, k+1\} \cup \{v_E : E \in F\} \cup U$. Set $P = \{(v_E, u) : E \in F, u \in E\}$.

For each $b \in \{1,2,\dots,k\}$, put in R as many parallel arcs from b to $b+1$ that there are subsets in F, and each subset in F is assigned to a distinct arc between b and $b+1$. Note that each subset is eventually assigned to k arcs: one from 1 to 2, one from 2 to 3, and so on. For each $a \in R$ with b and $b+1$ as endpoints, set $S_a = \{b, b+1\}$ and $L_a = \{v_E\}$, with E being the subset in F assigned to a. Finally, set $r_b = 1$ if $b \in U$ and $r_b = 0$ otherwise.

There is no difference between the two versions of the problem for this specific instance. It is then straightforward to check that there exists $F' \subseteq F$ with $|F'| = k$ and $\cup F' = U$ if, and only if there is a valid path with a value at least U.

MATHEMATICAL PROGRAMS

A mathematical formulation for each version of the problem is now given. It takes the form of an ILP, with a slight difference between the two versions. The way to solve it is also briefly discussed. For the first version of the problem, it is an exact modelling: solutions of the problem, and of the integer linear program coincide. For the second version, it is a heuristic modelling: the solutions of the integer linear programs form a subset of the solutions of the problem.

Integer linear program: common part.
Define two sets of variables: binary variables x_b, defined for each$b \in V$, and binary variables y_a, defined for each $a \in R$. The definitions are the following:

(1) $$x_b = 1 \text{ if block } b \text{ is extracted} \wedge 0 \text{ otherwise.}$$

(2) $$y_{a=(b,b')} = 1 \text{ if a ramp is constructed from } b \text{ to } b' \wedge 0 \text{ if not.}$$

The objective function consists of maximising the total value of the extracted blocks

(3) $$W = \sum_{b \in B} r_b x_b$$

The constraints are

(4) $$y_a + x_b \leq 1 \qquad \forall a \in R, b \in S_a$$

(5) $$y_a \leq x_b \qquad \forall a \in R, b \in L_a$$

(6) $$x_b \leq x_{b'} \qquad \forall (b, b') \in P$$

(7) $$\sum_{b':(b',b) \in R} y_{(b',b)} = \sum_{b':(b,b') \in R} y_{(b,b')} \qquad \forall b \in V$$

(8) $$\sum_{b:(s,b) \in R} y_{(s,b)} = 1$$

Constraint (4) prohibits extraction of blocks that must stay because of the ramp. Constraint (5) imposes extraction of blocks required by the ramp. Constraint (6) forces the extraction of blocks due to precedence. Constraint (7) is a way to make the $y_{(b,b')}$ encode a path and constraint (8) makes the path start at s. (Actually, these constraints impose just that the $y_{(b,b')}$ encode an $s - t$ path together with circuits, but the fact that each arc $a \in R$ corresponds to a level change means that there will eventually be no circuit.) Note that a constraint similar to constraint (8) for t is not necessary since the other constraints allow neither s, nor any b to be the endpoint of the path.

These variables, constraints, and objective function are common to the two versions of the problem. Each version has a specific constraint, which is now introduced.

First version
In the first version of the problem, where the set $X(p)$ has to be minimal, the following constraint is also needed.

(9) $$x_b = \sum_{a:b\in L_a} y_a \forall b \in V$$

This constraint means that if a block is extracted, this is due to the ramp going 'below' it.

Second version
In the second version, where the set $X(p)$ does not have to be minimal, the following constraint is added.

(9) $$x_b \leq \sum_{a:aendsatlevelofb} y_a \forall b \in V$$

This constraint means that a block can be extracted as soon as the ramp reaches its level. In order to ensure the existence of a horizontal access to every extracted block from the ramp, the set P also contains arcs oriented to the pit (determined by breadth-first-search, which ensures that the extracted blocks are on the shortest paths to the pit). The resolution of the ILP will then create for each extracted block a 'path' of extracted blocks to the pit.

This is the heuristic part: the second version only requires the existence of an access from any extracted block to the pit, and there might be many ways to satisfy this requirement. Yet, the trick of adding arcs oriented to the pit in P makes the extraction of a block by the ILP lead to the 'automatic' extraction of a predetermined sequence of blocks to the pit, without any freedom on the way to select these blocks.

SOLUTION APPROACH

A natural way to solve the ILP is with the help of an off-the-shelf solver. This is the first ILP-based approach.

Unfortunately, the problem quickly becomes very large, and the resolution is time-consuming for any solver. A way to shorten the computation time consists of the following heuristics run beforehand:

- Divide the circumference of the reduced pit into m connected parts of about the same length (where m is a parameter of the method).
- Select in each part the most profitable block.
- Solve the ILP with a set R in which the arcs leaving s have been reduced to those of the form (s, b), with b selected at the previous step. (The vertex s has thus an outdegree equal to m.)

This second ILP-based approach is called the restricted one.

Another approach proposed in this paper is a simple and natural greedy procedure. It progressively builds the ramp by adding the arcs in a greedy manner: it starts with the most valuable arc in R that leaves s; an iteration extends the current partial ramp by selecting the most valuable arc in R from the current endpoint; ties are broken arbitrarily. The value of an arc is the value of the extra blocks that have to be extracted because of its addition. For both versions, the models of Section 5 are used to determine which blocks to extract in function of the chosen arcs. In particular, an ILP solver is used at each iteration to determine which arc to choose.

The greedy approach is improved by randomising some steps. More precisely, the standard greedy randomised adaptive search procedure (GRASP) is followed. Feo and Resende introduced and developed this metaheuristic in the late 80s and developed it in the 90s (Feo and Rosende 1989; Feo and Rosende 1995). It has been used in many combinatorial problems. The method is an extension of the greedy algorithm in which a solution to the problem is constructed iteratively, by adding 'elements' or 'pieces' to the current candidate, until a solution is formed. To select the next element to be added, a value or ranking function is utilised. The difference between the greedy algorithm and GRASP is that in the former approach, the element with the highest rank is added in each iteration, while, in the latter, this element is chosen randomly from a list of the elements with highest rank.

Using GRASP in mine planning optimisation is not new. For example, (Riff, Otto, and Bonnaire 2009) applied GRASP to solve a scheduling problem in underground mining extracted by a caving method.

EXPERIMENTAL RESULTS

Instances and compared methods

The data sets utilised for numerical experiments are based in Marvin, which is a commonly known block model from the literature. From this block model, five different instances are considered. They are described in Table 1. Three instances require the ramp to have a width of two blocks (their name is in the form of M30-XX). Two instances require the ramp to have a width of one block (their name is in the form of M15-XX). The '-XX' indicates how many levels are considered.

The precedence constraints for the first version of the problem are only the vertical ones. For the second version, 'horizontal' constraints are added to these precedence constraints as explained in Section 5: for each block b, a shortest horizontal path to the pit is computed; all blocks on this shortest path have to be extracted when b is extracted. It models the fact that, if a block that is not near the ramp has to be extracted, then it has to be reached from the pit. For each version of the problem, the four methods described in Section 6 are experimented with (two based on ILP and two greedy algorithms). For the restricted ILP-approach, $m = 3$ is set (i.e., the circumference of the pit is subdivided into three connected parts of almost equal length), but for the instances Marvin-30, only two parts contain valid starting blocks. For GRASP, in a procedure generating a valid $s - t$ path, each iteration consists of selecting uniformly, at random, among the three best arcs; the procedure has been repeated 50 times; only the best path (in terms of value) is kept.

Gurobi v9.0.0 has been used for the resolution of the ILPs (Gurobi 2018). All experiments have been performed on a computer with 32 GiB of RAM and 4 cores at 1.80 GHz and Linux Ubuntu 18.04.3 LTS, 64-bit operating system. There is a time limit of 7,200 s for the experiments.

Table 1. Summary of numerical instances

ID of Instance	# Benches	# Blocks in V	# Blocks in reduced pit
M30-05	5	1,314	3,032
M30-10	10	2,392	3,594
M30-15	15	3,063	3,596
M15-10	10	5,083	33,392
M15-20	20	9,364	53,168

Numerical Results
The computational results are gathered in Tables 2 to 4.

Tables 2 and 3 summarise the results related to the value of the ramps and the computation time required to find the respective solutions, for the first and second versions of the model. The first column contains the name of the various instances on which the methods have been experimented. These instances are described in Section 7.1. Then, there are four groups of two columns: one group for each method 'Greedy', 'GRASP', 'ILP', and 'Restricted-ILP'. For each group, the left-hand column shows the value obtained with the method and the right-hand column shows the computational time required by the method. Regarding 'Greedy', almost all computation time is spent on the reading of the instances.

The true optimal value could in theory only be proved with the method ILP. A star (*) has been added when the optimal value was obtained. Otherwise, the optimality gap is provided.

For 'Restricted ILP', all models have been solved to optimality (but it is only a lower bound on the true optimal value, probably close to it).

Table 4 reports the results in terms of the value of the pit. That is, it compares the value of the original pit with the value that takes into account the ramp design. This is obtained by adding the value of the ramp (i.e., of the blocks that need to be extracted for making space for the ramp) to the value of the reduced pit, which is used as the reference volume.

Discussion
Tables 2 and 3 make clear that the first version of the problem is very hard to solve. Indeed, within the computation limit of six hours, only the smallest instance could be solved up to optimality (M30-05), and other instances have gaps around 100% or more. Moreover, when the blocks are profitable, computing times are much smaller. The reason why ILP is so bad when blocks have negative value remains unexplained. The restricted ILP approach is a good option when near optimal solutions are sought.

As expected, the second version provides in general better solutions than the first version; indeed, any solution of the first version is a solution of the second version, but this latter allows more possibilities of removing valuable blocks. The instance M30-05 is the only exception, and this is due to the heuristic way the existence of a 'path' of extracted blocks from any extracted block to the pit (see Section 5.3) is taken into account. This shows that finding a more clever and more accurate way to model this constraint could be interesting, and improve the quality of the solutions.

Overall, the heuristics perform very well, obtaining solutions that are near the optimal ones in a fraction of the time. More importantly, Table 4 shows that the heuristics generate volumes with space for the ramps, but with a value that is nearly the same as that when the reduced pit is used as a reference. Indeed, except in one case, all values are within 0.013% of the value of the reduced pit. This is a very good result when compared with the literature, which reports gaps up to 1% for Marvin and up to 10% in other block models (Morales, Nancel-Penard, and Parra 2017).

Table 2. Computational results for ramp design in open pit mine (first version)

Instance	Greedy		GRASP		ILP		Restricted ILP	
	W (USD)	Time (s)	*W* (USD)	Time (s)	*W* (USD)	Time (s)	*W* (USD)	Time (s)
M30-05	-2,352	42	-2,235	63	-2,206*	1,871	-2,322*	6
M30-10	-12,187,166	64	-10,089,185	98	-9,521,431 gap=158.24%	3,342	-9,135,577*	4,402
M30-15	25,769	68	27,477	112	18,300 gap=323.69%	4,340	31,440*	6,824
M15-10	-3,944	87	-2,313	129	-2,058 gap=98.94%	7,189	-1,965*	5,122
M15-20	-9,438	95	-8,161	148	-14,738 gap=145.14%	6,928	-7,748*	7,186

Table 3. Computational results for ramp design in open pit mine (second version)

Instance	Greedy		GRASP		ILP		Restricted ILP	
	W (USD)	Time (s)	*W* (USD)	Time (s)	*W* (USD)	Time (s)	*W* (USD)	Time (s)
M30-05	-4,955	20	-2,572	46	-2,304*	479	-2,430	3
M30-10	-523,497	41	-413,924	85	-373,603 gap=2,667%	7,196	-364,118	688
M30-15	40,428	47	42,923	97	44,924*	611	42,820	46
M15-10	-3,434	71	-2,276	118	-1,945 gap=67.29%	7,198	-1,612*	165
M15-20	-8.044	86	-7,083	125	-6,782 gap=85.29%	7,152	-5,607*	454

Table 4. Impact of ramp in value of the pits

Instance	Reduced Pit Value	Gain/Loss in Value First Version (%)				Gain/Loss in Value Second Version (%)			
		Greedy	GRASP	ILP	Rest. ILP	Greedy	GRASP	ILP	Rest. ILP
M30-05	123,058,769	-0.002%	-0.002%	-0.002%	-0.002%	-0.004%	-0.002%	-0.002%	-0.002%
M30-10	341,084,968	-3.573%	-2.958%	-2.792%	-2.678%	-0.153%	-0.121%	-0.110%	-0.107%
M30-15	341,838,517	0.008%	0.008%	0.005%	0.009%	0.012%	0.013%	0.013%	0.013%

M15-10	189,357,415	-0.002%	-0.001%	-0.001%	-0.001%	-0.002%	-0.001%	-0.001%	-0.001%
M15-20	1,019,009,031	-0.001%	-0.001%	-0.001%	-0.001%	0.000%	-0.001%	-0.001%	-0.001%

CONCLUSIONS

This paper addresses the strategic problem of open-pit ramp design, for which previous works have suggested utilising an optimisation model that uses an initial pit as a reference, and finds another that is close in volume, while containing enough space for the ramp design. Following this line of work, mathematical formulations modelling two versions of the problem are proposed, and shown to be NP-hard. Therefore, four different solutions approaches are proposed: an exact algorithm (with a computation limit), a heuristic based on mathematical programming, a greedy heuristic, and a GRASP algorithm.

In terms of computational results, the quality of the solutions obtained using an exact algorithm emphasise the difficulty of the problem, as the feasible solutions found within a two-hour time limit present optimality gaps of 100% or more in many cases. However, the proposed heuristics can find solutions which are competitive in value but can be found in a fraction of the time (less than two minutes).

In terms of the application, the results are very promising, because in almost all instances, the resulting pit has a value with a loss in value that is smaller than 0.1% of the original pit. (In the only instance where this is not true, the loss is about 3%, which is consistent with other results in the literature for similar cases). These results encourage the application of the algorithm in other block models, and to look for extensions incorporating multiple pits.

ACKNOWLEDGEMENTS

This work was partially supported by Université Mohammed VI Polytechnique, Morocco and ANID Basal Funding Grant AFB180004. Last but not least, the authors warmly thank Pierre Nancel-Penard for his help in the preparation of the instances.

REFERENCES

BAI, X., Marcotte, D., Gamache M., Gregory, D., and 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.

Brazil, M. and Thomas, D. A. (2017). Network optimization for the design of underground mines. Networks, 49:40-50.

Brazil, M., Thomas, D. A., Weng, Jia F., Rubinstein, J. Hyam, and H. Lee D. (2005). Cost optimisation for underground mining networks. Optimization and Engineering, 6(2):241-256.

Feo, Thomas A. and Resende, Mauricio G. C. (1989). A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters, 8(2):67-71.

Feo, Thomas A. and Resende, Mauricio G. C. (1995). Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization, 6(2):109-133.

Gurobi (2018). Gurobi™ Gurobi Optimization. http://www.gurobi.com, 2018. Accessed 18 January 2021.

Loor, V. and Morales, N. (2020). Applying artificial intelligence for optimal production scheduling and phase design in open pit mining. In Proceedings of the 9th international conference on Mass Mining.

Morales, N., Nancel-Penard, P., and Parra, A. (2017). An integer linear programming model for optimising open pit ramp design. In Proceedings of the 38th International Symposium APCOM proceedings, pages 9-16.

Nancel-Penard, P., Parra A., Morales, N. Díaz, C., and Widzyk-Capehart, E. (2019). Value-optimal design of ramps in open pit mining. Archives of Mining Sciences, vol. 64(No 2):399-413.

Riff, M. C., and Otto, E., and Bonnaire X. (2009). A new strategy based on GRASP to solve a macro mine planning. In Jan Rauch, Zbigniew W. Ra´s, Petr Berka, and Tapio Elomaa, editors, Foundations of Intelligent Systems, pages 483-492, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tabesh M., Mieth C., and 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.

Yardimci, A. G., Karpuz, C. Optimized path planning in underground mine ramp design using genetic algorithm. In Proceedings of the Symposium of Mine Planning and Equipment Selection, pages 247-252. Division of Operation and Maintenance Engineering, Lulea University of Technology.

Yarmuch, J. L., Brazil, M., Rybunstein H., and Thomas D. A. (2020). Optimum ramp design in open pit mines. Computers & Operations Research, 115:104739.

Islem Kaabachi

PhD in Computer Science
CERMICS Laboratory

I am a post doctoral researcher in the field of Operations research & Optimization hosted by Prof. Frédéric Meunier. Currently, I am part of the optimization team in the CERMICS laboratory at École des Ponts et Chaussées ParisTech, France. Before then, I completed my PhD in Computer Science supervised by Prof. Saoussen Krichen in the LARODEC laboratory at Institut Supérieur de Gestion de Tunis, Tunisie . My thesis focused on the transportation logistics management using decision support systems.