



A heuristic sublevel stope optimizer with multiple raises

by X. Bai*, D. Marcotte*, and R. Simon*

Synopsis

A new heuristic sublevel mining stope optimizer is presented. The optimizer seeks the best locations and lengths of a series of vertical raises that, together with the blocks linked to each raise, define a mining stope. Five design constraints – the footwall angle, the hangingwall angle, the number of raises, the maximum distance of a block from a raise, and the minimum width required to move the farthest block towards the raise – allow the shape of the sub-stopes associated with each raise to be controlled. The optimization is done on the locations and lengths of raises using a genetic algorithm to efficiently sample the parameters' space. For each raise, a local network is defined in cylindrical coordinates around the raise such as to impose the design constraints. A maxflow algorithm on the local network is used to determine the optimal sub-stope for each raise. All sub-stopes are combined to define the global stope for the entire deposit. The best global stope is obtained using a genetic algorithm to find the raise parameters providing the best profit over the entire deposit. Two synthetic cases and one real deposit are used to evaluate the new algorithm and compare the results with the single-raise optimizer. The multiple raises approach leads to significantly improved economics compared with the single-raise stope optimizer, and the dilution is also substantially reduced compared to the single-raise case.

Keywords

underground mining, stope optimization, mining constraints, maximum flow algorithm, cylindrical coordinate transformation, multiple raises.

Introduction

In underground mining, stope design affects the profit and safety of the operation. Stope design requires: (1) a prior ore reserve model as input data, usually obtained by estimation or simulation using geostatistical tools (David, 1988; Journel and Huijbregts, 1978), and (2) the geotechnical constraints, including the hangingwall and footwall angles, the stope dimensions, the *in situ* stress tensor, the rock strength, and the local geological structures. The general procedure of stope optimization is to decide which volumes are included in the stope and which are not so that, under the geotechnical constraints, the resulting stope produces the greatest profit possible.

In the last few decades, several approaches have been developed for stope optimization. These methods were reviewed by Ataee-Pour (2005) and Alford *et al.* (2007). The dynamic

programming method (Riddle, 1977) and branch and bound technique (Ovanic and Young, 1995) were used to optimize a stope in one or two dimensions. However, these methods fail to produce realistic stopes for complex three-dimensional (3D) deposits that cannot be simplified to two-dimensional (2D) mining problems. Some 3D techniques were also reported, including mathematical morphology tools (Serra, 1982; Deraisme *et al.*, 1984), floating stope technique (Alford, 1996), maximum value neighborhood method (Ataee-Pour, 2000), and octree division approach (Cheimanoff *et al.*, 1989). These heuristic methods cannot directly integrate the geotechnical constraints. Recently, Manchuk and Deutsch (2008) provided a simulated annealing-based algorithm, with some mining constraints incorporated. However, simulated annealing is very slow and the convergence to a global optimum is therefore not ensured in practice. Moreover, the restriction of perturbations to moves respecting all constraints on slopes in a three-dimensional setting can seriously hamper the capacity of the algorithm to find a good solution.

Bai *et al.* (2013) developed a stope optimizer based on graph theory. The basis of the approach is the vertical raise initiating the opening necessary for blasting, which plays a similar role as the surface in open-pit mining. A cylindrical coordinate system is defined around the raise. Then a network is built where cylindrical blocks are linked towards the raise such as to impose *de facto* the geotechnical constraints. The optimal block selection is obtained by applying efficient maximum flow methods over the network. Two important design parameters were defined: the maximum distance a block can be

* Department of Civil, Geological and Mining Engineering, École Polytechnique de Montréal, Canada.

© The Southern African Institute of Mining and Metallurgy, 2014. ISSN 2225-6253. Paper received Jun. 2013; revised paper received Sep. 2013.

A heuristic sublevel stope optimizer with multiple raises

to reach the raise (R) (or distance of influence of the raise), and the minimum width required to move this farthest block to the raise (y_R) (see Figures 1 and 2). The stope obtained is optimal for the raise location and extent chosen, and for the R , y_R , and footwall and hangingwall angles imposed. The global stope optimization then simplifies to finding the best raise location and extent within the orebody. The optimizer was shown to provide good results on a number of simple deposits, both synthetic and real.

Although quite appealing, the single-raise optimizer has a few drawbacks. Firstly, it cannot provide a satisfying solution for large deposits or lenses where more than one raise is needed. In that case, repetitive application of the single-raise optimization needs to take into account the interactions between the raises. Also, for small deposits or lenses with curved shapes (e.g. following folds), the single-raise optimal solution could provide more dilution and less profit than a manual solution obtained with more raises, each raise having a smaller distance of influence R .

In this paper, the authors aim at solving these drawbacks. The algorithm of the single-raise optimizer is extended to multiple-raise situations, keeping the core component of generating a sub-stope for each raise. In the multiple-raise framework, each sub-stope is a feasible geometry with controllable maximum dimensions. Optimization is done on the set of raise parameters using a genetic algorithm to efficiently sample the parameter space. For a given set of raise parameters, each raise is optimized with the single-raise optimizer, thus defining as many sub-stopes as there are raises. Each sub-stope meets the design parameters and is optimal for that particular raise location and length and set of design constraints. The union of all points within one or more of the sub-stopes defines heuristically the global stope. Three deposits, two synthetic and one real, are considered where the results obtained with the single raise and multiple raises are compared and discussed. For the sake of simplicity, the true grade values of the deposit are assumed known everywhere or obtained using a conditionally unbiased estimator (David *et al.*, 1984), so the effect of the uncertainty on grades with regard to the stope design is not considered in this study.

Methods

Stope optimization with a single raise

The stope optimizer using a single raise proposed by Bai *et al.* (2013) is reviewed. After a brief summary of the required graph theory for mining optimization, the network construction used to represent mining constraints is described and the stope optimizer workflow is presented.

Graph theory in mining optimization

The ore block model and the mining constraints are represented as a weighted directed graph (or network) $G = (V, A)$, where the vertices V denote the ore blocks and the oriented arcs A define the precedence relations between blocks so as to incorporate the mining constraints. The profit from mining block i is p_i . It is computed from the block i ore grade and tonnage, the recovery factor, the mining and processing costs, and the mineral price (Lane, 1988).

The stope optimization amounts to finding the closed set of nodes $V' \subseteq V$ such that $\sum_{i \in V'} p_i$ is a maximum. Let Γ_i be the subset of immediate successor nodes to node i , representing the set of blocks that need to be mined prior to block i . The maximum closure problem is formulated as:

$$\text{Maximize } \sum_{i=1}^N p_i x_i \quad [1]$$

$$\text{Subject to } x_i - x_j \leq 0, \forall i \in V, j \in \Gamma_i \quad [2]$$

$$x_i = 0 \text{ or } 1, \forall i \in V \quad [3]$$

where x_i equals 1 when the block is selected, 0 if not, and index j refers to a successor of block i .

For a typical deposit, the integer program involves a lengthy computational time due to the large number of ore blocks N . The Lerchs-Grossman algorithm (LGA) (Lerchs and Grossman, 1965) presented an effective tool to solve the open-pit mining problem implemented in some commercial software. Even more efficient methods appeared after the seminal paper of Picard (1976) proving that the maximum closure problem of the open-pit mine is equivalent to the minimum cut problem, hence allowing the application of maximum flow algorithms (e.g. Goldberg and Tarjan (1988); King *et al.* (1992)), which are substantially more efficient than the LGA (Hochbaum, 2001, 2002). The high efficiency enables the repetitive application of the algorithm while still keeping computing costs realistic.

Implementation of stope geometric constraints in network

To apply the network flow concept to stope optimization, the key is to find the free surface to start stoping, similar to the ground surface in an open-pit mine. Actually, in sublevel stoping, the raise, the vertical or sub-vertical tunnel, plays the role of the initial free surface. The introduction of cylindrical coordinates starting from the raise (Figure 1a) facilitates the control of geometric constraints as the stoping sequence can be expressed by linkages of the cylindrical blocks toward the raise. Bai *et al.* (2013) indicated how the common stope constraints are implemented with the different linkages in a graph. The hangingwall and footwall slope constraints define the precedence links in the vertical direction (Figure 1b). For a cylindrical system with blocks defined by Δr , $\Delta\theta$, Δz (see Figure 1b), and $\Delta z/\Delta r = 1$, one link upward and two links downward define a hangingwall slope of 45° and a footwall slope of 63.4° . Stope width is controlled by linkages in the horizontal plane, specifically one radial link and two side links for each block (see Figure 2a) and by two design parameters: the maximum extent of the stope from the raise R (or distance of influence of the raise) and the minimum width (y_R) needed to move, by gravity, the farthest block to the raise. For example, with $R = 30$ m, $\Delta\theta/\Delta r = 1$ degree per metre, three horizontal links towards the raise provide $y_R = 7.7$ m, (see Figure 2).

A third type of geotechnical constraint, the maximum stope height, is simply controlled by the length of the raise. The blocks above the top of the raise or under its bottom are not part of the network, hence are not contained in the stope.

Algorithm for a single raise

The optimization algorithm consists of two main parts. The first part, the stope optimizer, is the core of the approach. It

A heuristic sublevel stope optimizer with multiple raises

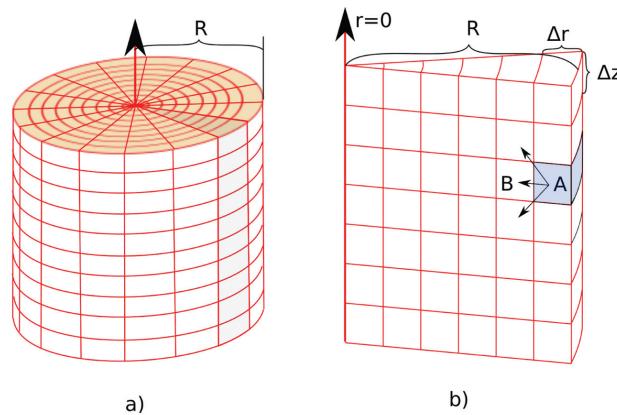


Figure 1—Block model under cylindrical coordinates (a), and typical arcs in vertical section in the proposed method (b)

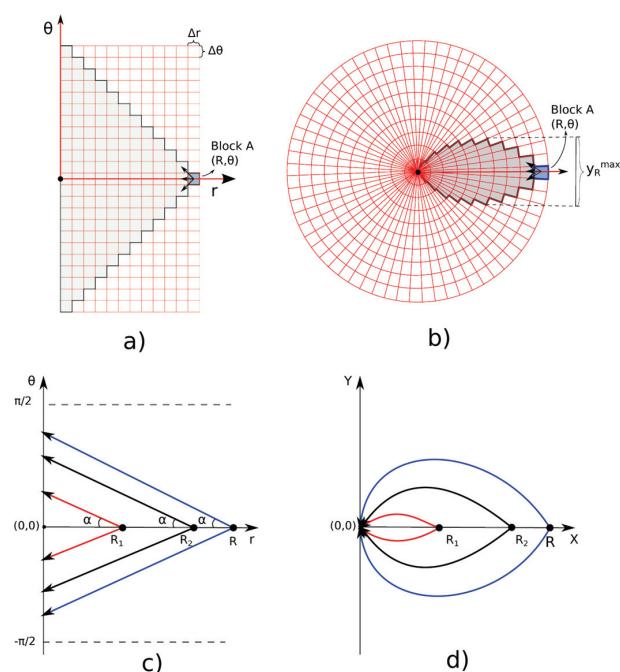


Figure 2—Horizontal plane showing (a) blocks and links defined in the cylindrical system and (b) corresponding blocks and links in the Cartesian system. Shaded blocks represent blocks to be removed to access block A. Trace of the envelopes defined by the lateral links in the cylindrical system (c) as they appear in the Cartesian system (d)

generates an optimal stope for a specified raise location and height, with chosen design parameters R and y_R . The stope optimizer includes the following steps:

1. Construct an economic block model in cylindrical coordinates with given raise location and height as the reference axis
2. Build the graph with vertical arcs to impose slope constraints, and horizontal arcs to impose width constraints
3. Construct the flow network by adding the source and sink nodes to the graph
4. Solve the maximum flow problem. The generated stope is conditionally optimal to the raise location and height.

The second part is to search the best raise location and height. This is done by global optimization on the raise location and height parameters, using as the objective function the stope value found with the stope optimizer.

The single-raise approach has some limitations. For example, when R is large, a relatively wide stope is produced as many blocks have to be mined before the farthest blocks are accessed. When the deposit is curved or inclined, this can lead to the mining of a substantial amount of waste as shown in Figure 3. In other scenarios, isolated clusters of ore could be left in the ground because the ore clusters do not pay for the additional waste included. In these cases, a better approach would be to use more than one raise so as to define smaller sub-stopes, hence diminishing the effects due to curvature or inclination of the orebody, and simultaneously allowing more flexibility to reach isolated clusters of ore.

Stope optimization with multiple raises

Similar to the single-raise algorithm, the multiple-raise algorithm is comprised of two main parts: (1) the stope generator with multiple raises based on a series of separate network flow problems, one for each raise; and (2) the optimization of the best parameters for the raises' locations, extents, and zones of influence.

Stope generator with multiple raises

Each raise is first treated separately with the single-raise optimizer described previously. For each raise, a cylindrical coordinate system is defined, and the stope constraints are implemented through the precedence relations in the associated network. The maximum extent that a raise can access, or distance of influence R_i , is defined to control the maximum size of the sub-stope for raise i . As a result, an optimal sub-stope is generated for each raise (see Figure 4). The sub-stopes in cylindrical coordinates are converted to a common regular grid in Cartesian coordinates. For the conversion, the status of a grid point in Cartesian coordinates (in or out of the stope) is identified by the status of the nearest cylindrical block centroid in each sub-stope. It suffices that a grid point belongs to any of the sub-stopes to be identified as being in the global stope. Hence, the global stope is the union of all the sub-stopes. The profit from the global stope is calculated on the Cartesian grid, so as to avoid counting any part of the stope twice (or more).

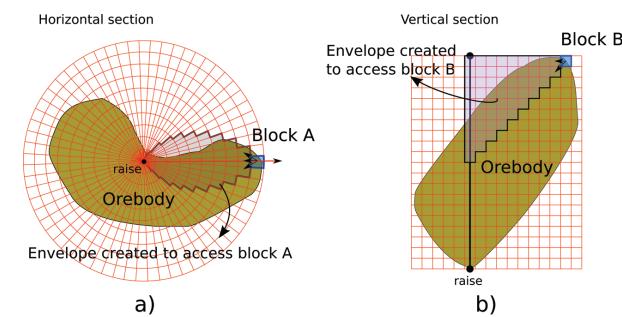


Figure 3—Illustration of possible problems with one raise: (a) in a horizontal section, the envelope from A to the raise includes a large quantity of waste; (b) in a vertical section, waste has to be mined in the upper part due to the network associated with the single raise

A heuristic sublevel stope optimizer with multiple raises

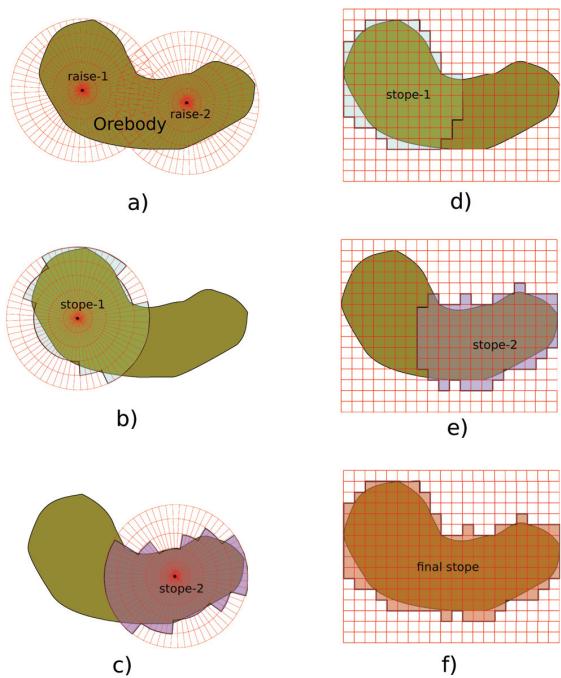


Figure 4 – Conceptual model of the stope generator with multiple raises in a horizontal section: (a) two ore models in cylindrical coordinates, one for each raise, are established; (b) and (c) first and second sub-stopes in a cylindrical coordinate obtained by the maxflow method on the two separate networks; (d) and (e) the sub-stopes (b) and (c) converted on the Cartesian grid; (f) the final stope in the Cartesian grid from (d) and (e)

Optimization of multiple raises parameters with a genetic algorithm

In the model, each raise is parameterized as, $(x_i, y_i, z_i^b, z_i^t, R_i)$, $i = 1, \dots, n$, where x_i and y_i denote the coordinates of raise i in horizontal section, z_i^b and z_i^t represent its bottom and top elevation, R_i is the maximum distance a block can be from the raise i , and n is the number of raises. The global stope is obtained as the union of sub-stopes that are each optimal in their local cylindrical system.

A good set of parameters for the raises is found using a genetic algorithm (Holland, 1975) to allow an efficient sampling of the parameters' space. The genetic algorithm tries to mimic the natural evolution of a population. An instructive example of the application of genetic algorithms to mining optimization is presented in Armstrong *et al.* (2012). Here, we define an individual as a single set of multiple raise parameters. Starting from a population of individuals, one creates new individuals in the population by crossover and mutations. In our algorithm, a vector of multiple raise parameters represents the individual chromosome with $5n$ genes, since there are 5 parameters to optimize for each raise. The profit from the global stope associated with the raise parameters measures the fitness of the individual to its environment. The algorithm is initiated by generating an initial population. The fitness of each individual in the population is evaluated. A certain proportion among more fit individuals is randomly selected as parents. Each set of parents mates and creates a child whose genes are inherited from them (crossover). Moreover, a certain proportion of

mutations are generated by introducing, in the child, chromosome - genes that do not come from the parents. The mutations enable new areas of the parameters' space to be explored. The less fit individuals in the population are eliminated so as to keep the size of the population constant. With the iterations, the average fitness of the population increases, until the convergence or another stopping criterion is reached (see Figure 5).

The genetic algorithm proposed follows the following steps:

- **Initial population**—The initial population comprises two parts: more fit individuals and random individuals. The more fit individuals ensure the inclusion of good genes. They can be obtained in two ways: (a) intuitive good raise parameters selected by the user; or (b) fast optimization with an initial low-resolution ore model. Random individuals covering the whole range of raise parameters are added to the initial population to allow sufficient genetic diversity so as to better explore the parameters' space and identify interesting area (Haupt and Haupt, 2004).
- **Parent selection**—The basic principle of parent selection is to give more fit individuals a higher priority to be parents. Less fit individuals have less probability to be selected for mating. Although the chance is small, it is important to allow the less fit individuals to pass down their genes for the purpose of diversifying the population. The fitness proportionate selection (or roulette wheel selection) method (Back, 1996) is used for parents selection. The individual i with fitness value (profit) f_i has the probability of being selected $prob_i = f_i / \sum_{j=1}^M f_j$, where M represents the number of individuals in the population. To implement this, all the fitness values are normalized to be located in $[0, 1]$. The normalized values are sorted in ascending order and then transformed to a cumulative normalized fitness value (CNFV). A random number Rnd is drawn

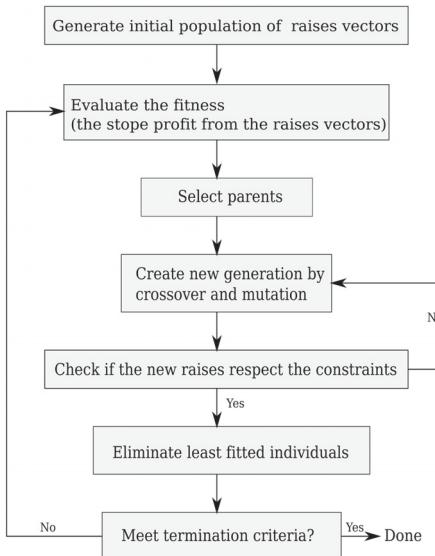


Figure 5—Genetic algorithm diagram to search for the best raises' parameters

A heuristic sublevel stope optimizer with multiple raises

from [0,1], and the first one with CNFV greater than Rnd is selected for the mating pool. The individuals in the mating pool are randomly paired.

- Genetic operator. Two mating methods are employed: crossover and mutation. The crossover is done by picking genes randomly from parents and combining them to define a new individual, following the formula:

$$\mathbf{X}_{\text{new}} = \sum_{i=1}^m \beta_i \mathbf{X}_i \text{ with } \sum_{i=1}^m \beta_i = 1, \beta_i = 0 \text{ or } 1 \quad [4]$$

where \mathbf{X}_{new} denotes the new individual, \mathbf{X}_i represents the parent i , and β_i is a 0-1 variable indicating whether the gene is inherited or not. In this way, a child is basically the recombination of genes from its parents. Mutation can be applied to a child to allow the child genes to depart substantially from those of its parents. This is done using (Haupt and Haupt, 2004):

$$\mathbf{X}'_{\text{new}} = \mathbf{X}_{\text{new}} + \sigma (\mathbf{Z} \circ \mathbf{I}) \quad [5]$$

where \circ denotes the Hadamard (or element-wise) product between the vectors; \mathbf{Z} is a column vector of random numbers drawn from the standard normal distribution; \mathbf{I} is a 0-1 column vector indicating the genes to be mutated, and the σ scalar controls the extent of the mutation. A posterior check is applied to the mutated genes so as to ensure they remain in feasible ranges for the raise parameters.

- *Termination of iterations*—The loop stops when a series of successive iterations does not improve the best individual fitness or the average fitness of the population, or when a maximum number of iterations has been reached.

Results

Parameters in the algorithm

To test and evaluate the proposed methods, three orebody models are used: two synthetic deposits and one real deposit (ore block model estimated by kriging). The two synthetic models illustrate typical scenarios where the single-raise algorithm partly fails and where the multiple-raise algorithm is expected to perform better. The initial synthetic block model is expressed on a Cartesian grid of spacing 1 m \times 1 m \times 1 m. For the larger real deposit model, the Cartesian grid is defined at every 2 m so as to save some computing time for the interpolation (it was checked that the results obtained are robust to this choice). The networks have one link vertically upward and two links downward, and three links horizontally. Therefore, for the three cases, the hangingwall angle is 45°, and the footwall angle is 63°. The $\Delta\theta$ is computed so as to ensure approximately the desired y_R (see Table I).

Table I
Geometric and design parameters, discretization, and optimized raise parameters

Parameters Raise type	Case 1		Case 2		Case 3	
	M ¹	S ²	M	S	M	S
Economic parameters						
Mining and processing costs (\$/t ore)			←	50	→	
Metal price (\$/kg)			←	10	→	
Recovery rate			←	0.9	→	
Rock density			←	3	→	
Mean ore grade (%)	2	2	0.25	0.25	0.54	0.54
Geometric parameters for stope						
Minimum footwall angle (deg)			←	45	→	
Minimum footwall angle (deg)			←	63	→	
Maximum height (m)	50	50	40	40	150	150
Minimum height (m)	10	10	15	15	50	50
Discretization						
dz (m)	0.5	0.5	0.5	0.5	1	1
dr (m)	0.5	0.5	0.5	0.5	1	1
Optimized raises parameters						
<i>Raise 1</i>						
Location X (m)	34.4	18.9	30.9	33.1	3164.7	3097.5
Location Y (m)	-20.9	20.9	21.4	18.7	-76.6	-72.5
Bottom level (m)	-127.3	-132.5	-124.5	-124.2	-493.8	-497.3
Top level (m)	112.3	-107.0	-105.0	-106.1	-397.8	-352.8
Maximum radius R (m)	16.2	27.3	12.7	33.7	18.9	78.7
<i>Raise 2</i>						
Location X (m)	10.2		48.0		3121.7	
Location Y (m)	20.2		18.9		-73.7	
Bottom level (m)	-132.9		-125.2		-498.0	
Top level (m)	-107.1		-105.2		-352.4	
Maximum radius R (m)	16.1		14.8		46.4	
<i>Raise 3</i>						
Location X (m)			12.0		3121.6	
Location Y (m)			17.5		-74.5	
Bottom level (m)			-125.2		-438.5	
Top level (m)			-105.7		-352.9	
Maximum radius R (m)			14.7		45.9	

¹M: Optimization with multiple raises

²S: Optimization with single raise

A heuristic sublevel stope optimizer with multiple raises

For the genetic algorithm, the initial population size is $40 \times n$, where n is the number of raises. Two more-fit individuals are included in the initial population. The first one is the solution obtained by optimization at a lower resolution. The second is obtained by spreading the raises uniformly within the deposit. Three parents are used to create a new individual. The mutation rate is selected to be 0.1 for each gene, so that many offspring will include mutations of their parent genes. In each iteration, $20 \times n$ new individuals are created and the same number of least fit individuals are eliminated to keep the size of the population stable. The optimization stops when the number of iterations reaches 100, or when the most fit individual among the population does not improve in 10 successive iterations. In the examples tested, these choices constitute a good tradeoff to ensure simultaneously a good final solution and to keep the computation time tractable. Increasing significantly the number of offspring or the population size, or making the termination rules more stringent, would possibly provide a slightly better final solution but at the cost of additional computation time. Admittedly, these values might have to be adapted to the particular deposit being studied. Bai (2013) verified with two simple cases that the preceding choices for the genetic algorithm were sufficient to ensure retrieving the known optimal value with high probability.

The design parameter y_R controls the minimum width the sub-stope must have for a block located at distance R from the raise. This value is likely to vary according to the rock mechanics condition of the deposit in the area where the stope is created. A smaller y_R allows dilution to be reduced and profit increased. However, it should not be too small, otherwise there is the risk of ore jamming within the stope. Rock mechanical conditions and experience with mining in a particular geological environment should guide the choice of this parameter. With the examples tested, a value around $y_R = R/3$ seems to provide visually sensible shapes. Here, to diminish the number of factors to study, we choose $y_{Ri} = R_i/3$ for all cases. The effect of y_R is further discussed later.

The first synthetic case represents two distinct mineralized lenses (Figure 6). The optimal single-raise solution (c and d) locates the raise in the waste approximately at mid-distance of the lens centroids. In contrast, the multiple-raises solution locates, as expected, the two raises close to the centroid of each lens (Figures 6e and f). Moreover, the radius of influence of each raise R_i is correctly identified as larger for the larger lens. This solution provides 13.5% more profit than the single-raise solution and the dilution of ore is reduced significantly from 21.6% to only 2.3% (Table II).

The second case is an ore vein with changing direction in horizontal section (Figures 7a and b). The vein is approximately 60 m long by 10 m wide by 20 m high. The stope is designed with three raises (see Figure 7). The value of the stope with multiple raises is 10.7% higher than with the single raise (\$976 000 vs. \$882 000). It includes less waste (cost \$4 400 vs. \$57 700) and misses less ore (\$4 100 vs. \$45 000). The dilution of the multiple raises solution is one-third the dilution of the single raise (3.1% vs. 10.4%) (Table II).

The kriging block model of a metal deposit in Canada is used as the third case study (the name and location of deposit are undisclosed for confidentiality reasons). A portion

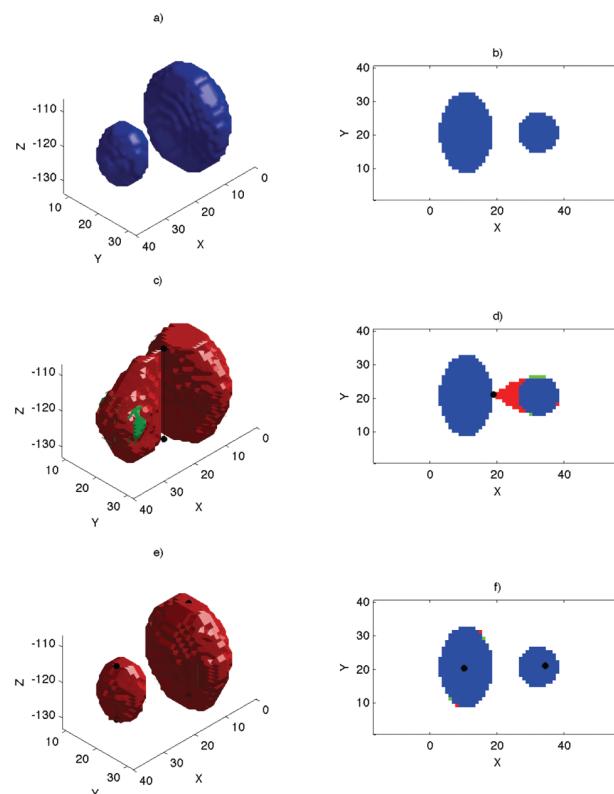


Figure 6—Case 1, simulated ore model and created stopes: (a) 3D-view of the orebody; (b) x-y horizontal section of the orebody at $z=-120$ m; (c) 3D view of the optimized stope with a single raise; (d) x-y horizontal section of the single raise-stope at $z=-120$ m, showing ore in stope (blue), waste in stope (red), and ore out of stope (green); (e) 3D view of the optimized stope by multiple raises; (f) x-y horizontal section of the multiple raises' stope at $z=-120$ m. Raises in black. Design parameters as in Table I

Table II
Economic evaluation of the case studies

Cases	Raise type	Stope profit (000\$)	Missed ore value (000\$K)	Waste value in stope (000\$)	Dilution volume rate 1	Profit improved
Case 1	Multiple	2622	24.2	-24.0	2.3 %	13.41 %
	Single	2312	84.6	-273.6	21.6 %	
Case 2	Multiple	976	4.1	-4.4	3.1 %	10.68 %
	Single	882	45.0	-57.7	10.4 %	
Case 3	Multiple	8933	90.4	-44.2	3.4 %	7.28 %
	Single	8327	543.8	-197.0	8.1 %	

¹Dilution volume rate = Volume of waste in stope / Volume of stope

A heuristic sublevel stope optimizer with multiple raises

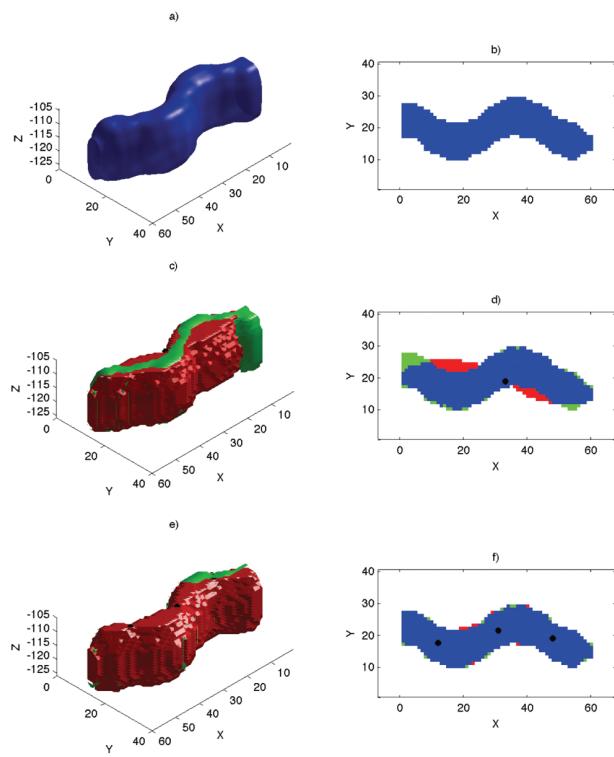


Figure 7—Case 2, simulated ore model and created stopes: (a) 3D-view of the orebody; (b) x-y horizontal section of the orebody at $z=-120$ m; (c) 3D view of the optimized stope with a single raise; (d) x-y horizontal section of the single raise stope at $z=-120$ m, showing ore in stope (blue), waste in stope (red), and ore out of stope (green); (e) 3D view of the optimized stope by multiple raises; (f) x-y horizontal section of the multiple raises' stope at $z=-120$ m. Raises in black. Design parameters as in Table I

of the deposit of size $108 \text{ m} \times 68 \text{ m} \times 148 \text{ m}$ (Figures 8a, b, and c), is selected for stope design for the sublevel stoping method. Three raises are used to optimize the stope as shown in Figures 8g, h, and i. The optimized raise parameters are given in Table I. This time, the profit of the multiple-raise stope is 7.3% higher than the profit of the single-raise stope. The dilution for the multiple raises is 3.4% of the stope volume compared to 8.1% for the single-raise solution.

Discussion

We have developed an improved stope optimizer for the sublevel stoping method. The new multiple-raise stope optimizer is an extension of the single-raise optimizer presented in Bai *et al.* (2013). In all test cases, the multiple-raise algorithm provides stopes with higher profit and less dilution compared to the single-raise optimizer.

In the three test cases presented, the best global stope from multiple raises generated between 7.3% to 13.4% more profit than the best stope obtained with a single raise. The improvement in the multiple raise heuristic solution compared to the single-raise approach is due to its increased flexibility.

The distance of influence of a raise R_i is the parameter controlling the size of a sub-stope. It is ensured that the geotechnical constraints are respected within each sub-stope,

therefore are respected in the global stope. The effect of the y_{Ri} parameter is case-specific. For a single raise and a given R , diminishing y_R necessarily increases the profit. However, y_R should not be taken too small otherwise there is a risk of ore jamming within the stope. Rock mechanical conditions and experience with the mining in that particular geological environment should guide the choice of this parameter. A value around $y_R = R/3$ seems to provide visually sensible shapes in the tests conducted. The sensitivity of the best solution to this parameter is not expected to be high. As an example, in case 3 the profits obtained with $y_R = R/2$, $R/3$, and $R/4$ are respectively \$8 909 000, \$8 933 000, and \$8 948 000, showing differences of only 0.4% between $R/2$ and $R/4$ and 0.17% between $R/3$ and $R/4$. Similar small differences were observed for the ore and the waste in the stope.

The computation time of the algorithm depends of two main factors: the time required for one iteration of the multiple-raise stope generator and the time required to explore the raise parameters' space by GA (or eventually an alternate search method). To give a rough idea, case 2 has $40 \times 60 \times 31$ blocks in a Cartesian grid, and the stope generator with three raises takes between two and ten seconds (on a laptop) to produce a stope. The exact time depends on the raise extent and on R_i which, for a given discretization, control the number of blocks. To find the best raise locations, the GA evolves during 57 generations, which takes around three hours of computation, the stope generator being called a total of $57 \times 20 \times 3 + 40 \times 3 = 3540$ times.

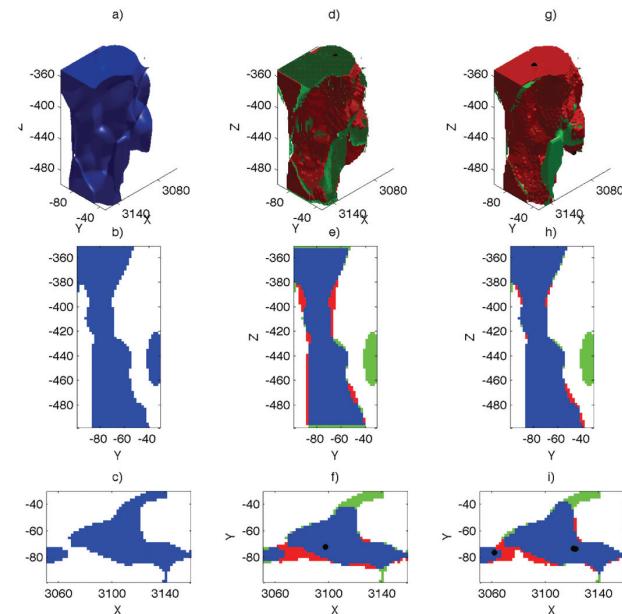


Figure 8—Case 3, test with a real ore deposit: (a) 3D-view of the orebody; (b) y-z vertical section of the orebody at $x=3130$ m; (c) x-y horizontal section at $z=-424$ m; (d) 3D view of the optimized stope with a single raise; (e) y-z vertical section at $x=3130$ m; (f) x-y horizontal section at $z=-424$ m; (g) optimized stope with multiple raises, (h) y-z vertical section at $x=3130$ m; (i) x-y horizontal section at $z=-424$ m. In (d) and (g), stopes are in red, ore out of stope is in green. (e), (f), (h), and (i), ore in blue, waste in red, and ore out of stope in green, raises in black. Design parameters as in Table I

A heuristic sublevel stope optimizer with multiple raises

One limitation of the proposed approach is the restriction to vertical raises. It would cause a higher dilution rate for scenarios of inclined deposits, which usually adopt inclined raises in reality. The generalization of the method to an inclined raise is far from evident due to the loss of symmetry with respect to the gravity force vector. The solution to the problem needs further investigation.

Also, in the proposed approach, the cost of development of access to the top and bottom levels of the raises was neglected. When the multiple raises are located at different levels, the relative additional costs would reduce the benefit of this approach compared to the single raise. Moreover, it was supposed that the raises could be located rather freely within the deposit without imposing constraints on the elevations of the beginning and the end of the various raises. An alternative strategy, closer to the practice for larger deposits, would be to optimize the common height between levels (within specified bounds) and impose each raise to span the entire height. The optimization would then simplify to find the best elevation for the first level and find the best number and locations of raises within each level. This modification is currently being investigated.

At first glance, the adoption of a cylindrical system of coordinates might appear as an unnecessary complication. In fact, it seems difficult (if even possible) to define the network directly in the Cartesian system of coordinates to ensure simultaneously all slope constraints, the raise width, and the distance of influence of a raise. Referring to Figure 2d, it is obvious that the links to impose vary according to the distance of influence considered. For each block, it would be necessary to compute the corresponding envelope of preceding blocks. Moreover, for a given distance of influence, any given internal block would be covered by many such envelopes coming from outer blocks. The network associated with the different envelopes for a given block might easily differ from one envelope to the other, rendering the network definition impossible. This complexity vanishes with the cylindrical system.

Conclusions

The proposed method was shown to provide good heuristic stope solutions for typical geometries of curved or inclined deposits. The solutions respect the geotechnical constraints of the sublevel stoping method, including footwall and hangingwall slopes, and stope minimum and maximum heights. The best stopes obtained with multiple raises for the three cases considered exhibit significantly larger profit (+7.3% to +13.4% increase) and less dilution (58% to 89% reduction) compared to the best solution by the single-raise method. The gain of the multiple-raises approach is due to the increased flexibility compared to the single-raise case.

Acknowledgement

This research was made possible by research grants provided by the Chinese Scholarship Council and the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors are grateful to two anonymous reviewers for their constructive comments and editing suggestions that helped improve the manuscript.

References

ALFORD, C. 1996. Optimisation in underground mine design. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, vol. 33, no. 5. pp. 220A–220A.

ALFORD, C., BRAZIL, M., and LEE, D.H. 2007. Optimisation in underground mining. *Handbook of Operations Research in Natural Resources*. Weintraub, A., Romero, C., Bjørndal, T., and Epstein, R. (eds.). Springer, New York. pp. 561–577.

ARMSTRONG, M., VINCENT, A., GALLI, A., and MHEUT, C. 2012. Genetic algorithms and scenario reduction. *Ninth International Geostatistics Congress*, Oslo, Norway.

ATAEE-POUR, M. 2000. A heuristic algorithm to optimise stope boundaries. Ph.D. thesis, University of Wollongong, Australia.

ATAEE-POUR, M. 2005. A critical survey of the existing stope layout optimization techniques. *Journal of Mining Science*, vol. 41. pp. 447–466.

BACK, T. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. 1st edn. Oxford University Press, USA.

BAI, X. 2013. Optimization of underground stope with network flow method. PhD thesis, École Polytechnique de Montréal, Canada.

BAI, X., MARCOTTE, D., and SIMON, R. 2013. Underground stope optimization with network flow method. *Computers and Geosciences*, vol. 52. pp. 361–371.

CHEIMANOFF, N.M., DELIAC, E.P., and MALLET, J.L. 1989. GEOCAD: an alternative CAD and artificial intelligence tool that helps moving from geological resources to mineable reserves. *21st International Symposium on the Application of Computers and Operations Research in the Mineral Industry*. Society for Mining, Metallurgy and Exploration Inc., Colorado. pp. 471–478.

DAVID, M. 1988. *Handbook of Applied Advanced Geostatistical Ore Reserve Estimation*. Elsevier.

DAVID, M., MARCOTTE, D., and SOULIE, M. 1984. Conditional bias in kriging and a suggested correction. *Geostatistics for Natural Resource Characterization*. Part 1. Verly, G., David, M., Journel, A.G., and Marechal, A. (eds.). Reidel, Dordrecht, Netherlands. Volume 122 of NATOASI C. pp. 217–230.

DERAISME, J., DE FOUCET, C., and FRAISSE, H. 1984. Geostatistical orebody model for computer optimization of profits from different underground mining methods. *Proceedings of the 18th International Conference on the Application of Computers and Operations Research in the Mining Industry (APCOM)*, London, England. pp. 583–590.

GOLDBERG, A. and TARJAN, R.E. 1988. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, vol. 35. pp. 921–940.

HAUPT, R.L. and HAUPT, S.E. 2004. The continuous genetic algorithm. *Practical Genetic Algorithms*. John Wiley & Sons. pp. 51–66.

HOCHEBAUM, D.S. 2001. A new-old algorithm for minimum-cut and maximum-flow in closure graphs. *Networks*, vol. 37. pp. 171–193.

HOCHEBAUM, D.S. 2002. Solving integer programs over monotone inequalities in three variables: a framework for half integrality and good approximations. *European Journal of Operational Research*, vol. 140. pp. 291–321.

HOLLAND, J.H. 1975. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, Ann Arbor.

JOURNEL, A.G. and HUIJBREGTS, C.J. 1978. *Mining Geostatistics*. Academic Press, London.

KING, V., RAO, S., and TARJAN, R. 1992. A faster deterministic maximum flow algorithm. *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*. Academic Press, Orlando, FL. pp. 157–164.

LANE, K.F. 1988. *The Economic Definition of Ore: Cut-off Grades in Theory and Practice*. Mining Journal Books, London.

LERCHS, H. and GROSSMAN, I.F. 1965. Optimum design of open-pit mines. - *CIM Bulletin*, vol. 58. pp. 47–54.

MANCHUK, J. and DEUTSCH, C. 2008. Optimizing stope designs and sequences in underground mines. *SME Transactions*, vol. 324. pp. 67–75.

OVANIC, J. and YOUNG, D.S. 1995. Economic optimisation of stope geometry using separable programming with special branch and bound techniques. *Third Canadian Conference on Computer Applications in the Mineral Industry*. Balkema, Rotterdam. pp. 129–135.

PICARD, J.C. 1976. Maximal closure of a graph and applications to combinatorial problems. *Management Science*, vol. 22. pp. 1268–1272.

RIDDLE, J.M. 1977. A dynamic programming solution of a block-caving mine layout. *Proceedings of the Fourteenth International Symposium on the Application of Computers and Operations Research in the Mineral Industry*, October 4–8, 1976. Society for Mining, Metallurgy and Exploration Inc., Colorado. pp. 767–780.

SERRA, J.P. 1982. *Image Analysis and Mathematical Morphology*. Academic Press, New York. ◆