Original Paper

A New Semi-greedy Approach to Enhance Drillhole Planning

Raphaël Dutaut[1,2] and Denis Marcotte[1]

In advanced exploration projects or operating mines, the process of allocating capital for infill drilling programs is a significant and recurrent challenge. Within a large company, the different mine sites and projects compete for the available funds for drilling. To maximize a project's value to its company, a drillhole location optimizer can be used as an objective tool to compare drilling campaigns. The fast semi-greedy optimizer presented here can allow for the obtention of close to optimal solutions to the coverage problem with up to three orders of magnitude less computing time needed than with integer programming. The heuristic approach is flexible as it allows dynamic updating of block values once new drillholes are selected in the solution, as opposed to existing methods based on static block values. The block values used for optimization incorporate kriging estimate and variance, estimate of indicator at cutoff grade and distances to existing or newly selected drillholes. The heuristic approach tends to locate new drillholes within the maximum risk areas, i.e., within less informed zones predicted as being ore zones. Applied to different deposits, it enables, after suitable normalization, comparison of different drilling campaigns and allocation of budgets accordingly.

KEY WORDS: Geostatistics, Infill drilling, Risk control, Semi-greedy optimization, Heuristic.

INTRODUCTION

During exploration and mining phases, the costs of infill drilling represent a significant portion of total project costs. Deposit definition is usually the result of a staged process with confidence in mineral resource estimates being gradually upgraded from inferred to indicated and measured categories (Canadian National Instrument 43-101). Uncertainty on a deposit can represent a high risk for the company in terms of investment (e.g., construction of a new mine or extensions) or actual production. Par-

ker (2012) indicates that changes in the net present value (NPV) can be six times the variations on grade, so that a 10% error in grade estimation can result in a 60% variation in NPV. In order to decrease the risks associated with resource estimation, obtaining more information through further drilling is the standard practice. Face with limited budget, optimization of drillhole localization becomes of strategic importance for mining companies.

Infill drillhole planning is generally performed manually by an experienced geologist. The drilling plan is then submitted for approval. To reduce subjectivity inherent to the exercise, drillhole location optimization has been studied since the late 1970s (Kim et al. 1977; Scheck and Chou 1983). These attempts focused mainly on selecting drillholes so as to minimize kriging variance (McBratney

[1]Civil, Geological and Mining Department, Polytechnique Montréal, C.P. 6079, Succ. Centre-Ville, Montreal, QC H3C 3A7, Canada.
[2]To whom correspondence should be addressed; e-mail: raphael.dutaut@polymtl.ca

et al. 1981; Gershon et al. 1988; Mohammadi et al. 2012; Jafrasteh and Fathianpour 2017). The main limitation of kriging variance is that it depends only on data location and the variogram model, but not on actual grade estimates. Mohammadi et al. (2012) proposed to use the product of kriging variance and kriged grade estimate to favor localization of new drillhole in high-grade areas with large uncertainty.

Conditional simulations have also been considered for drillhole location optimization. They allow generation of equiprobable realizations of the random function (Chilès and Delfiner 2012). Both grade and domains (or lithologies) can be simulated with different algorithms such as sequential simulation (Journal 1974), turning bands (Matheron 1973), moving average (Chilès and Delfiner 2012) or simulated annealing (Kirkpatrick et al. 1983) among many other methods. For each node or block, multiple realizations can be generated, providing a measure of local uncertainty. Pilger et al. (2001), Pinheiro et al. (2017) or Zagré et al. (2018) used various forms of conditional simulations to define the uncertainty parameter to be minimized.

The drillhole optimization problem was modeled by Bilal et al. (2013) and Zagré et al. (2018) as a set covering problem. In this approach, if a block's distance to the closest drillhole is less than a selected distance threshold, it is considered fully covered; otherwise, it is considered fully uncovered. It is then possible to formulate the best coverage for a set of blocks by a set of drillholes as an integer program as mentioned in Bilal et al. (2014). The coverage approach has two main drawbacks. First, it is very slow for large problems. Second, it uses a crude definition of coverage. More realistically, a block should be considered as only partially covered when some drillholes are found close to the block.

We propose two main improvements to the previous approaches. First, we define an objective function focusing on the riskier parts of the deposits and allowing easy updating of block values (BVs) without having to redo the kriging every time a drillhole is selected. Second, we present a new heuristic approach using a semi-greedy method that executes quickly and allows, for the first time, a partial coverage of blocks. The proposed approach is applied to a real deposit where it is also compared to integer programming solution for execution time and optimality gap in the special case of a 0–1 weighting function.

METHODOLOGY

Objective Function

The proposed block value (BV) function is:

$$BV(x|h) = Z_k^*(x) \cdot \sigma_k^{2*}(x) \cdot I_a^*(x) \cdot \prod_{i=1}^{n} \omega_1(x|h_i) \quad (1)$$

where h is the set of existing drillholes and already selected drillholes, n is the cardinal of the set h, $BV(x|h)$ indicates that the BV depends of the set h, $Z_k^*(x)$ is the kriged grade estimated at block x, $\sigma_k^{2*}(x)$ is the block kriging variance and $I_a^*(x)$ is the indicator value (Journal 1982) estimated at block center using cutoff grade a. Only existing drillholes (not the selected ones) are used to compute $Z_k^*(x)$, $\sigma_k^{2*}(x)$ and $I_a^*(x)$. Hence, these quantities have to be computed only once. Only the weighting function $\omega_1(x|h_i)$ varies, between 0 and 1, with the selected drillhole h_i. It is 1 for blocks outside the h_i, and it is 0 for a block x that is crossed at its center by drillhole h_i. Equation (1) gives high values to blocks with high-grade estimate $Z_k^*(x)$, high uncertainty $\sigma_k^{2*}(x)$ and high probability $I_a^*(x)$ to be above the cutoff grade. For a given grade estimate and uncertainty, it seems logical to prioritize covering of blocks that can be exploited at a lower cutoff grade (hence with higher $I_a^*(x)$). Once a given drillhole h_i is selected, the likelihood of selecting another drillhole in the same area diminishes drastically because of the function $\omega_1(h_i)$. This avoids clustering of drillholes in a high-grade area.

Similarly, we define the coverage of a candidate drillhole h_i as:

$$K(h_i) = \sum_{j=1}^{nb} BV(x_j|h) \cdot \omega_2(x|h_i) \quad (2)$$

where nb is the number of blocks and $\omega_2(x|h_i)$ is a decreasing function, which is equal to 1 when x is crossed at its center by h_i and 0 for blocks outside the distance of influence of drillhole h_i. Note that ω_1 diminishes the value of a block that is already informed by at least one nearby drillhole and ω_2 decreases the coverage of a block that is located farther from the candidate drillhole. The classical approach used in coverage optimization (e.g., Zagré et al. 2018) implicitly uses ω_1 as a 0–1 step function at the drillhole distance of influence and ω_2 as a 1–0 step function at this same distance. Our formulation is, therefore,

more general as it allows continuous variation of the BV and drillhole influence.

The objective function to minimize is simply the sum of BVs in Eq. 1 given the existing and added drillholes, thus:

$$OF = \sum_{i=1}^{nb} BV(x|h \cup h_{\text{add}}) \qquad (3)$$

In the semi-greedy approach, we do not seek to minimize exactly Eq. 3 but rather to find good solutions by sequentially choosing drillholes with high $K(h_i)$ values in Eq. 2 as described in the next section below.

The quality of a solution is best expressed by its coverage value:

$$C_{\text{Sol}} = \frac{OF_0 - OF_{\text{Sol}}}{OF_0} \qquad (4)$$

where OF_0 is the value of the objective function using only the existing drillholes and OF_{Sol} is the value of the objective function obtained with the solution proposed. Figure 1 presents a theoretical 2D example of the dynamic updating of weighting schemes. The example area has an extent of 100 by 100 units, and two existing virtual data (red squares) were positioned at [0, 50] and [100, 50]. The two upper images present ω_1 (left) and ω_2 (right) for a first tested point positioned at [50, 25]. The red circle lines are drawn at 25 units from the existing and tested positions (distance of influence). The two bottom images show the updated weights at a second point (at [50, 75]) once the first point is accepted.

Semi-greedy Optimization

The drillhole positioning optimization problem belongs to the family of the "set covering problem" (Bilal 2014), one of the 21 NP-complete problems defined by Karp (1972). Some heuristic algorithm used for drillhole positioning optimization includes simulating annealing (Pinheiro et al. 2017), genetic algorithm (Soltani et al. 2011), bees colonies (Jafrasteh and Fathianpour 2017), particle swarm (Fatehi et al. 2017) and tabu search (Zagré et al. 2018). These algorithms are CPU time intensive for large 3D models. To circumvent this limitation, a fast and flexible semi-greedy heuristic algorithm (Hart and Shogan 1987) is proposed.

The algorithm was developed in MATLAB software and works for either 2D datasets (best suited for a single lens or a tabular deposit) or 3D datasets. In 3D, a series of possible collar positions, orientations, dips and lengths of drillholes are evaluated to provide close to an optimal solution.

Semi-greedy Algorithm

The semi-greedy algorithm works sequentially, one drillhole at a time. A new drillhole is randomly selected among a list of *nlist* best drillhole candidates. A new selected drillhole is added to the set of already selected drillholes. The blocks' values are updated, and the coverage function of the remaining candidate drillholes is recomputed and sorted to account for the newly selected drillhole. The sequential selection continues until the drilling budget is exhausted. Because there is a random component involved when selecting one drillhole from the sorted candidate list, the process can be repeated *ntrial* times, each time providing a different subset of selected drillholes. In the end, the best solution among the *ntrial* is kept. In the semi-greedy algorithm, there are three parameters to specify: the size *nlist* of the list of best drillhole candidates, the number *ntrial* of trials to run and the distance of influence of a drillhole (*dmin*).

- Initialization. The input consists of:
- blocks to consider with their block-values BV ($x|h$) (Eq. 1)
- existing drillholes
- candidate drillholes
- length of best candidate list (*nlist*)
- number of trials (*ntrial*)
- distance of influence of a drillhole (*dmin*)
- available drilling budget

- For each trial, until drilling budget is exhausted
- compute BV ($x|h$) per block x considering the existing and already selected drillholes h (Eq. 1)
- compute coverage value $K(h_i)$ (Eq. 2) for the remaining candidate drillholes and sort it in decreasing order of coverage
- randomly select one among the *nlist* best drillhole candidates
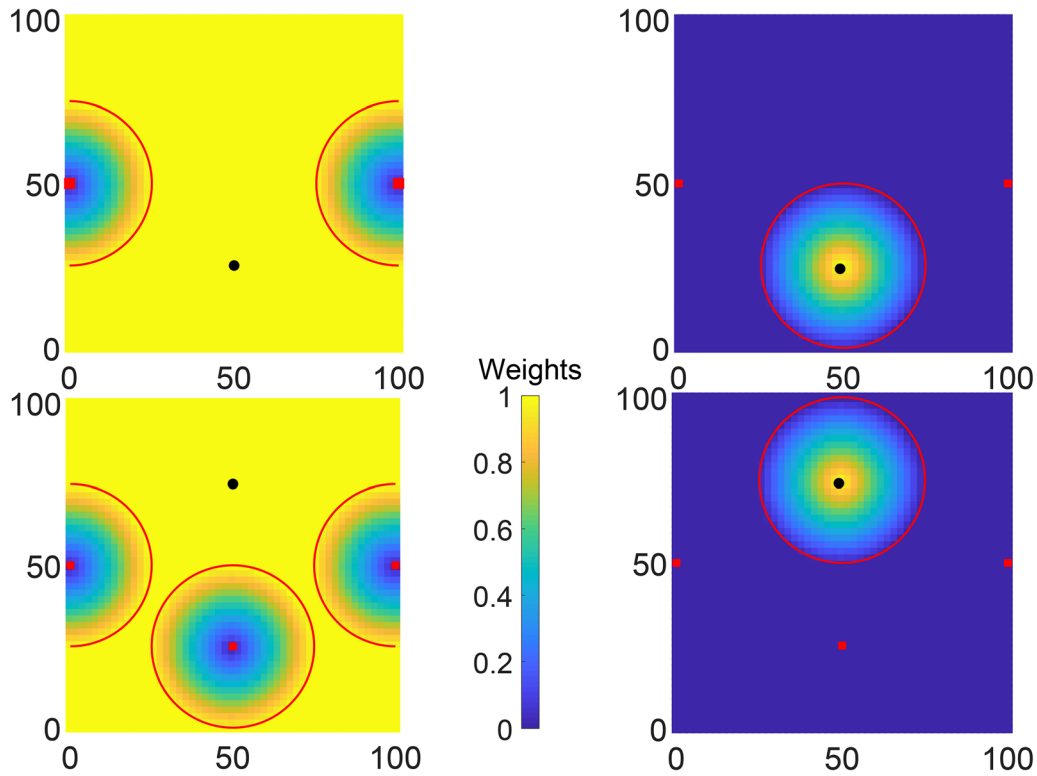
Figure 1. Weighting scheme theoretical example. Upper row presents ω_1 (left) and ω_2 (right) for the point (50, 25) (black dot) accounting for existing or already selected drillholes (two red dots). Bottom row shows the updated weights for point (50, 75) once point (50, 25) has been selected.

- Compute *OF* value (Eq. 3) for current solution. The solution with the best coverage among the *ntrial* solutions (Eq. 4) is kept.

When updating the blocks' values, only function $\omega_1(x|h_i)$ modifies the *BV*s. Kriging estimates (grade and indicator) are left unchanged as the selected drillhole is not yet materialized. Although kriging variance can be recomputed to account for the reduction in uncertainty brought by the newly selected drillhole, this was not implemented in our approach. The weight function $\omega_1(x|h_i)$ essentially fulfills this purpose by reducing the value of a block already informed by a nearby drillhole (real or selected). This avoids repetition of kriging for each potential candidate drillhole, which drastically reduces the CPU time required.

The Role of nlist Parameter

The parameter *nlist* represents the length of the list of the current best candidates. When *nlist* = 1,

we have the greedy solution, i.e., at each step we take the best drillhole available. When *nlist* = number of candidate drillholes, the selection is purely random.

A simple example illustrates that the semi-greedy approach provides better solutions than a greedy approach. Consider a series of potential drillholes with unit separation along a line and values: 1.1–2.1–3.1–4–3–2–1. To simplify, we use the step weighting function ω_2 (with *dmin* = 1.5) and we suppose two drillholes are needed and can only be located at the above points. The greedy approach first selects point 4 (best) and then point 2.1 (as points 3.1 and 3 are already fully covered by point 4, their value becomes 0). The total coverage value is 1.1 + 2.1 + 3.1 + 4 + 3 = 13.3. If we use a semi-greedy approach say with *nlist* = 2, points 3.1–4 are candidates for the first draw. Assume 3.1 is selected, the next draw will be between 3 and 2 (as 2.1 and 4 are already fully covered, their value is updated to 0). Suppose point 2 is selected. The total coverage is then 2.1 + 3.1 + 4 + 3 + 2 + 1 = 15.2, which is a significant improvement over the greedy choice.

Note that, in this simple example, the optimal choice is to select points 2.1 and 3, which give a total coverage of 1.1 + 2.1 + 3.1 + 4 + 3 + 2=15.3. This optimal choice would be available in the semi-greedy approach with *nlist* ≥ 3 and will eventually be selected provided we draw often enough (i.e., large *ntrial*).

Choice of Distance of Influence dmin

The choice of the distance of influence can be based on the geologist's knowledge and experience with the consideration of the variogram effective correlation range, the geometric features of the deposit, similar outcrops or of already exploited areas. A noticeable advantage of our approach is that it enables to have a decreasing weight function from 1 for distance zero to 0 at the distance of influence, instead of only a step function as in Zagré et al. (2018).

For linearly varying weight functions $\omega_1(x)$ and $\omega_2(x)$, increasing *dmin* diminishes $BV(x|h)$ for each block but increases the weights given to each block when computing drillhole coverage with Eq. 2. The solutions obtained are relatively robust to the choice of dmin for linearly varying weight functions (see next section).

CASE STUDY

Datasets Presentations

The proposed methodology and algorithm were applied to two different datasets from the Rosebel Gold Mine (RGM) in Suriname (South America). The RGM is situated within the greenstone belt of the Guianese shield, and it is composed of multiple orogenic gold deposits (Daoust et al. 2011). RGM produced approximately 5 Moz of gold since the beginning of its commercial production in 2004.

To assess both 2D and 3D capabilities of the software, two partial datasets were extracted from two different deposits from the RGM complex. Multiplicative factors were applied to the gold grade values to protect confidentiality without altering any of the results or conclusions in the present research.

2D Application

The 2D dataset corresponds to a subzone of the RGM's deposit. The ore zone selected corresponds to a vertical shear zone extending approximately over 400 m along strike and 500 m vertically. In total, 96 drillholes crossing the ore zone were selected to provide initial information.

The gold grades of the intersecting portions of the drillhole, within the ore zone, were calculated as a single composite, and the horizontal width of the ore zone was saved. A 10 × 10 m regular grid was created to define blocks within the deposit. The same grid was used as possible definition drillhole locations. Variogram models were used to interpolate the variables grade, thickness and grade indicator above a cutoff grade of 1 g/t on the grid. The *BV* function Eq. 1 was then calculated for each block including ore thickness. The basic statistics of the initial *BV*s are presented in Table 1. Locations farther than 100 m from existing drillholes were considered too remote to be considered for definition drillholes. A crisp 1–0 coverage distance was applied to each drillhole to allow direct comparison with integer programming results. A *dmin* of 26 m, specific to the Rosebel case, was selected as a compromise: It is small enough to be compatible with full coverage but large enough to allow some overlapping between drillhole locations. The *dmin* to be used is expected to vary depending on geological continuity, drill spacing and smu size specific to the case under study. The impact of the *dmin* distance on results is assessed in the next section below.

Figure 2 presents a 2D longitudinal view of the existing drillholes (black points), and the *BV*s obtained considering the existing drillholes. The selected drillholes by semi-greedy algorithm (red points) for sets of 1, 5, 10 and 20 drillholes are indicated. The selected drillholes effectively cover the high values areas. It can be noted that the exact position varies with the number of drillholes recommended. Hence, for example, the best location when $N = 1$ is not necessarily retained for solutions corresponding to a larger N, which matches the expected behavior (Fig. 2).

Table 1. Block values basic statistics for the 2D dataset

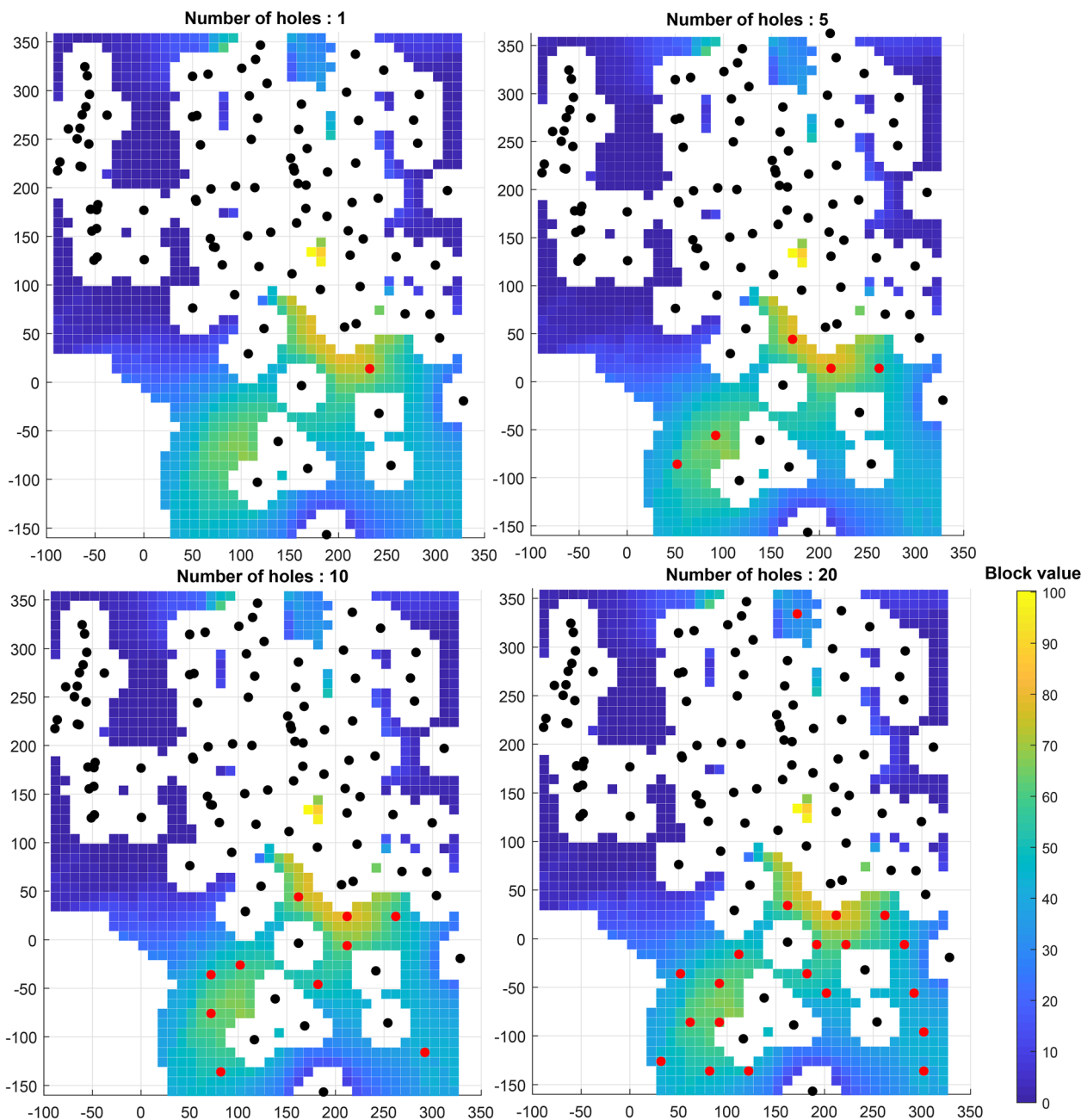| Number of blocks	Minimum	Maximum	Mean	SD
873	0	100.2	27.0	22.9

Figure 2. Longitudinal 2D results view. Black dots correspond to existing drillholes with ore intersects. Red dots are the resulting drillholes from the optimization. *BV* considering only the existing drillholes is illustrated.

With a 0–1 weight function, we can compare the best values of the semi-greedy algorithm to the exact optimal solution obtained by integer programming. For this, we use the linear program described in Zagré et al. (2018). We stress that the integer programming approach is limited to the step function weighting scheme and does not allow for partial coverage of blocks, contrarily to our semi-greedy approach.

Figure 3 presents the coverage expressed in percentage of the total *BV*s, as a function of the number of drillholes optimized (a proxy for budget)
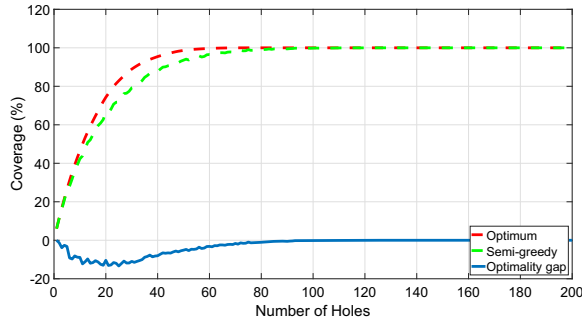
Figure 3. Coverage as a function of number of drillholes. Semi-greedy with *ntrial* = 100 and *nlist* = 50.

Figure 4. (**a**) Coverage for 20 drillholes as a function of *nlist*, using *ntrial* = 300. (**b**) Optimization coverage results for 1–200 drillholes for different *nlist* with *ntrial* = 300.

Figure 5. Coverage results for varying *ntrial* and number of drillholes, with *nlist* = 50.

by both integer programming and the proposed semi-greedy algorithm. With *dmin* = 26 m and a maximum distance of 100 m from existing drillholes, approximately 80 drillholes suffice to cover entirely the area. Both curves initially show a fast increase in coverage and then a slower rate of improvement, due to then filling the areas with lower *BV*s. The blue line represents the optimality gap for each solution expressed as the ratio $(C_{sg} - C_{opt})/C_{opt}$ (C_{sg} is coverage with semi-greedy; C_{opt} is coverage with integer programming). The maximal optimality gap reaches around 13% for N between 15 and 25. Afterwards, the optimality gap decreases progressively.

Two parameters mainly influence the quality of the results and the calculation time of the semi-greedy algorithm: *nlist* and *ntrial*. Figure 4a shows the semi-greedy coverage results for 20 drillholes as a function of *nlist* (using *ntrial* = 300). Increasing *nlist* rapidly improves the quality of the solution at the beginning, but when *nlist* becomes too large, the total coverage decreases. Indeed, setting *nlist* equal to the total number of drillholes available would be equivalent to randomly selecting N drillholes and keeping the best draw among *ntrial* trials, a poor method of optimization. Conversely, setting *nlist* to 1 corresponds to a purely greedy solution, again, not the best optimization approach. An easy method to select *nlist* is to simply run the algorithm with a series of increasing values for *nlist* until the maximum has been identified. Because the semi-greedy algorithm is fast, this approach is tractable.

Figure 4b presents the results of applying the semi-greedy algorithm with different *nlist* for 1–200 drillholes (with *ntrial* = 300). The best results were obtained for *nlist* between 10 and 100 for all number of drillholes to be selected. This suggests that a good
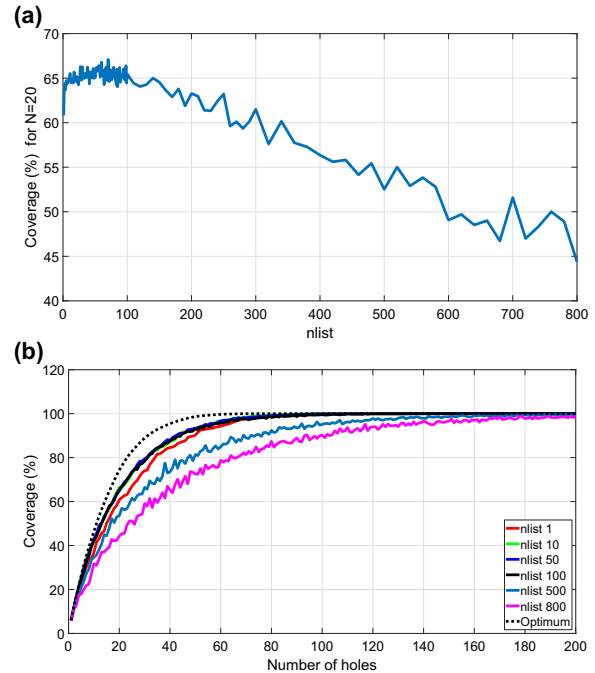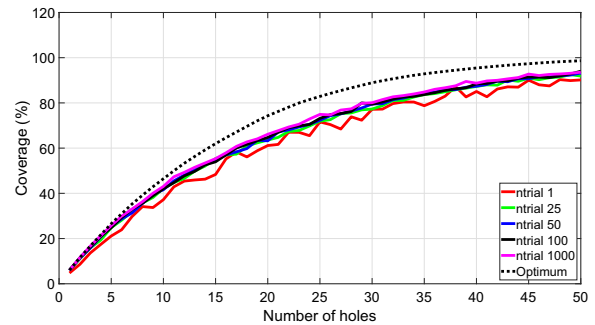
choice for *nlist* is relatively independent of the drilling budget represented here by the number of drillholes.

The other parameter that can be adjusted is the number of trials *ntrial*. Using a fixed *nlist* = 50, Figure 5 demonstrates convergence to a decent solution after a relatively low number of trials (i.e., 25–50). Increasing *ntrial* to above 100 improves the results only marginally but increases computation time linearly.
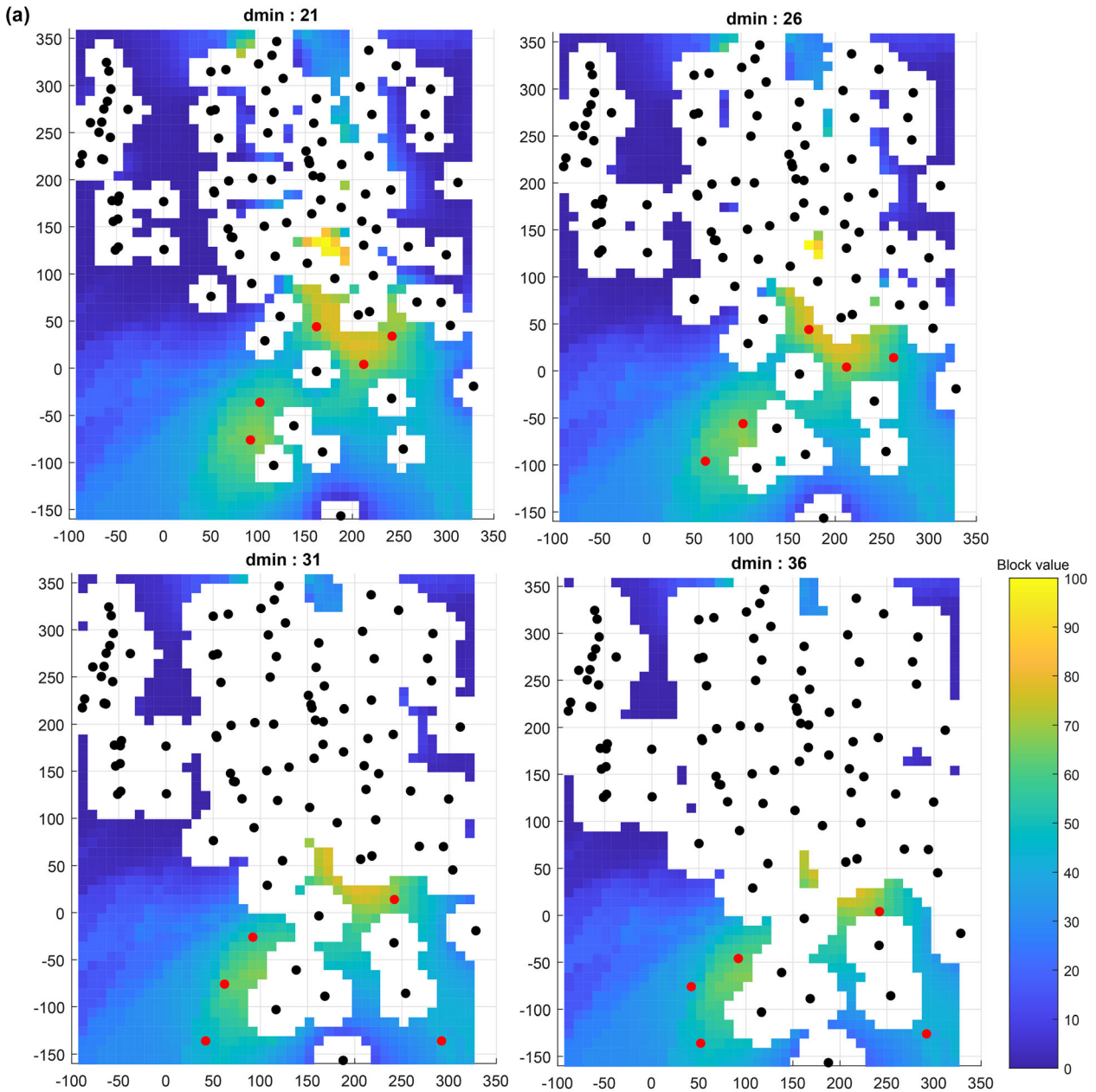
(a)

Figure 6. Longitudinal 2D results for five drillholes and increasing *dmin* with step (**a**) and linear (**b**) weight functions. Existing drillholes (black dots) and drillholes selected (red dots) by semi-greedy algorithm with *nlist* = 50 and *ntrial* = 1000. Map of *BV*s considering only the existing drillholes is illustrated.

Influence of dmin for Step or Linear Weight Functions

The use of step functions in the previous sections was done to allow direct comparison of the semi-greedy solutions to the optimal solutions obtained by integer programming. It is obviously more realistic to assume that the coverage of a block due to the presence of a drillhole decreases regularly as a function of distance. The influence of choice of *dmin* for both types of weight functions is examined.

Results obtained for five new drillholes with *dmin* values of 21, 26, 31 and 36 used with step and linear functions are shown in Figure 6. The corre-
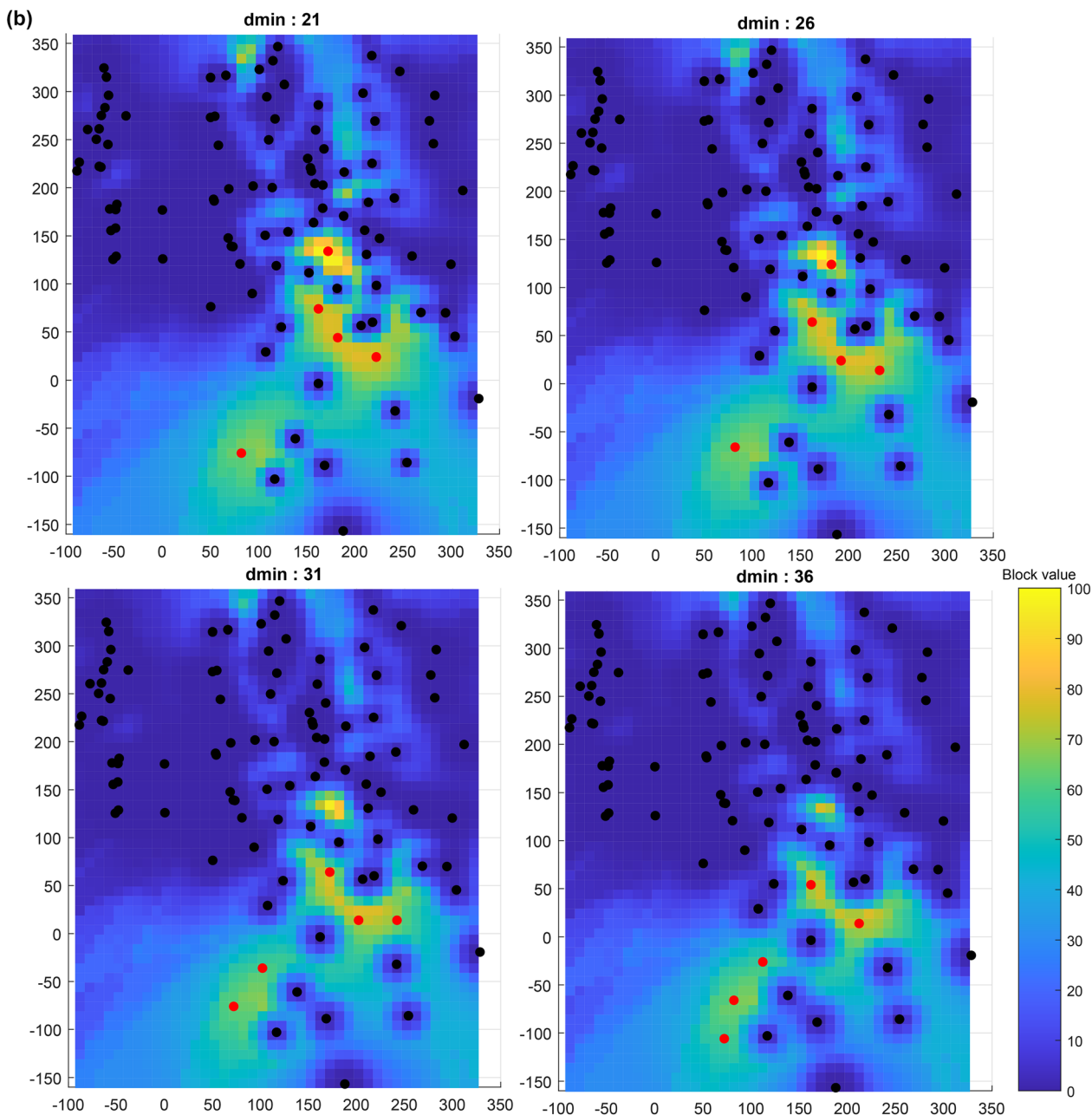
Figure 6. continued.

sponding coverage value obtained (computed with Eq. 4) is given in Table 2. As expected, the choice of *dmin* is more influential for the step functions than for the linearly varying functions. For the step functions, increasing *dmin* to 26 m removes the drillhole located in the central part of the deposit as this area becomes covered by the existing drillholes. Increasing further *dmin* increases separation be-

tween drillholes and locate them more on the periphery. The coverage C_{sol} increases with *dmin* as total *BV* to be covered diminishes, and the area covered by each drillhole increases. We retained a value of *dmin* = 26 m because it is close to typical separation for definition drilling.

In the linear case, a drillhole remains selected in the central area with *dmin* = 26 m (contrary to the

Table 2. Coverages for five drillholes as a function of *dmin* for the step and linear weight functions

dmin (m)	21	26	31	36
Step functions (%)	12	21	29	37
Linear functions (%)	4	6	8	11

Table 3. Block values basic statistics for the 3D dataset

Number of blocks	Minimum	Maximum	Mean	SD
19,447	0	1.67	0.031	0.078

step function). Further increasing *dmin* also tends to increase drillhole separation and place them more on the periphery but less so than as the step function. These results indicate lesser dependency of C_{sol} and drillhole location with the linear functions than with the step functions.

3D Application

The 3D dataset corresponds to one of the RGM active pits. The mineralization is composed of a succession of sheared zone with quartz veins of different orientations. The extension of the mineralization is about 1 km in easting and 200 m on northing with vertical extension of about 300 m. The mineralization has been partly mined out, but some pushbacks are planned in the coming years. The actual topography was used to generate possible drillpad locations on a regularly spaced 25-m pattern.

A restricted area of 400 m of extension (Easting) was selected within the entire 3D area. In total, 178 diamond drillholes totaling 23,635 m and 3855 6-m composites were available in this zone. The deposit was discretized in blocks of size 10 × 10 m horizontally × 8 m vertically. *BV*s were calculated using Eq. 1 for all blocks with a cutoff of 0.3 g/t to define the ore indicator variable. The *BV*s statistics of the 3D case are presented in Table 3. They differ significantly from the statistics presented in the 2D case, because of the varying grade, spacing and economical cutoff (indicator). Moreover, thickness does not intervene in the 3D case.

The zone of influence of each drillhole was also set at 26 m for ω_1 and ω_2 functions. These were defined as step functions to allow comparison with

integer programming results. Figure 7a shows the existing drillholes composites (in black) with *BV*s > 0.1 as blue squares and the possible collar positions extracted from topography in green. Figure 7b presents the evaluated drillpads as well as all the generated composites from drillholes. In order to account for mineralization orientation as well as existing drillholes, drillhole azimuth orientations of 165, 180 and 195 were forced, associated with plunges of 60, 75 and 90, respectively. A fixed drillhole length of 102 m was used. With these parameters, a total of 3672 candidate drillholes were defined.

Optimization result for $N = 10$ drillholes (totaling 1020 m) is presented in Figure 8, using *nlist* = 20, *ntrial* = 30. The proposed drillholes (in red) are localized within area of high *BV*s and spaced by at least 26 m from each other and from existing drillholes.

Figure 9 shows the coverage as a function of the number of drillholes for optimal integer programming solution compared to the semi-greedy solution for the 3D case dataset. The gap of optimality is generally less than 10%. The proposed algorithm ran all the cases shown in the figure in about 1240 s, compared to the 455,403 s (about 127 h) it took for the integer programming.

Simultaneous Optimization

An interesting application of the proposed method is to optimize two or more deposits simultaneously and use the results to support budget allocation. This requires a full or partial normalization of the components defining Eq. 1 for the *BV*s: grades, variograms sills and/or indicator cutoffs. It is then possible to optimize the coverage of the riskiest areas for both deposits simultaneously. The choice of the normalization depends on the main purpose of the definition campaign. As an example, Figure 10 presents the results of a simultaneous optimization for the previous 2D and 3D examples. Variograms sills were normalized to 1, and cutoffs were set to 1 and 0.3 g/t for the 2D and 3D datasets, respectively. In this example, the composite grades were normalized by dividing them by the mean grade of all composites in the respective datasets. A flat topography was added to the 2D dataset so that it could be treated as a 3D case. Both datasets were then combined, and the semi-greedy algorithm was applied to the combined dataset. In this application,
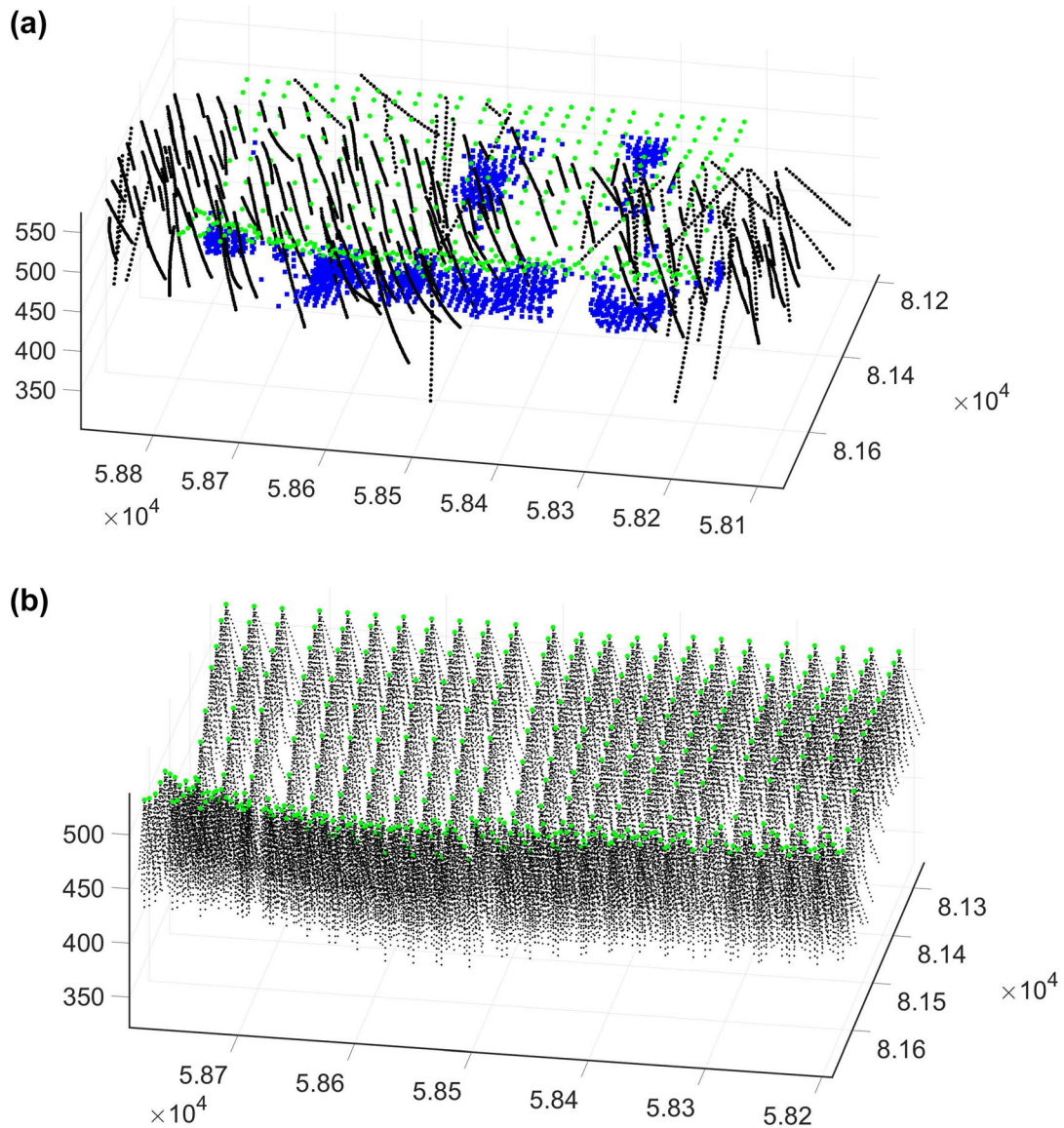
(a)

(b)

Figure 7. (**a**) A total of 178 existing drillholes composites (black dots) and blocks with $BV > 0.1$ (blue squares), within the selected area. (**b**) Trace of the 3672 drillholes that fulfill constraints on drillpad locations (green dots), direction and dips.

ω_1 and ω_2 were set as a linear function [from 0 to 1 using $dmin$ = 26 m].

Figure 10 shows the total coverage obtained for both projects combined (red line) as a function of the allocated budget (in m) from 500 to 10,000 m. For a relatively small budget of up to 3000 m, most of the added drillholes were positioned within the 3D zone. When the drilling budget exceeded 3000 m, then most of the additional drillholes were located in the 2D project as most blocks with high

OF values in the 3D case were already partially covered by the drillholes added in the initial steps.

DISCUSSION AND CONCLUSIONS

The most common practice for drillhole optimization consists of minimizing the kriging variance or statistics extracted from conditional simulations results. Often, professionals simply implement systematic and regular spacing regardless of any geo-
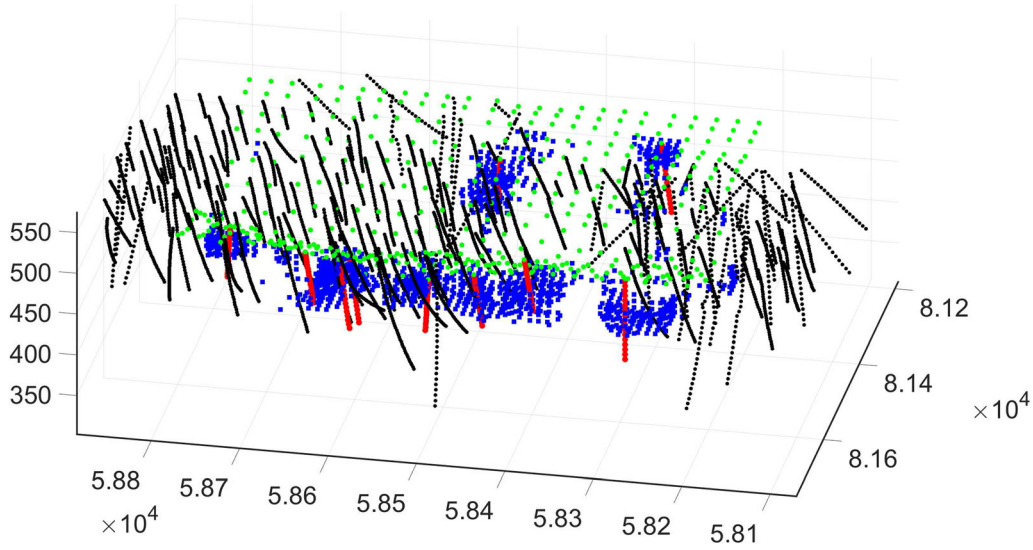
Figure 8. Topography (green), existing drillholes (black), 10 selected new drillholes (red) and blocks with $OF > 0.1$ (blue).
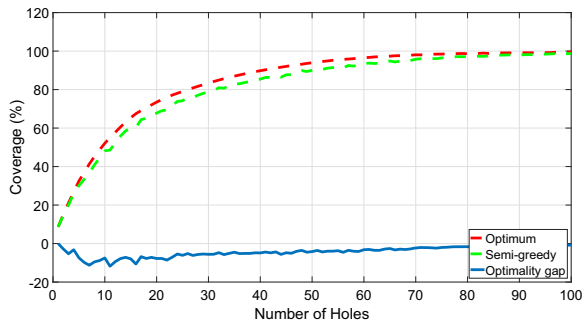

Figure 9. 3D coverage as a function of number of drillholes.
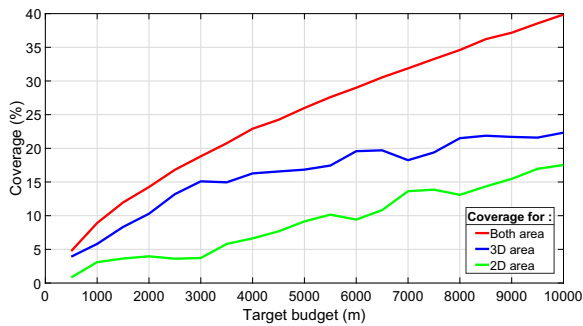

Figure 10. Coverage as a function of drilling budget (in meters) for the two datasets optimized simultaneously.

statistical parameters. These approaches do not capture entirely the economic risks of a project or operation, which generally lie within the high-grade ore zones. An excess or lack of drillholes within the areas of risk is possible. This would correspond to a loss for the company (either as a risk or opportunity). The proposed OF function incorporates economical risk by maximizing the coverage of additional definition drillholes. Other OF could have been used with our heuristic method. As an example, conditional variances obtained by simulation could replace kriging variance. Here, we limited our study to a reasonable OF reflecting the main goals to achieve for the Rosebel case study: avoiding drill-hole clustering, targeting primarily the areas with expected high grades, and reflecting the economic factors by the probability to be ore material. More research is needed to compare the utility of different OF variants for decision purposes in a variety of geological and economical contexts.

In our test, the semi-greedy algorithm was close to three orders of magnitude faster than integer programming. This computational advantage is expected to increase with larger instances. However, the main advantage relative to integer programming is to allow dynamic updating of blocks values as new drillholes are selected. We stress that both grade kriging and indicator kriging remain unchanged as updating is taken into account entirely by the weight functions ω_1 and ω_2. The use of ω_2 function can be

seen as a fast alternative to updating of kriging variances each time a new drillhole is tentatively selected. Updating of kriging variance would mean performing $(nfor - 1) \times ntrial$ local kriging on the blocks found in the neighborhood of each added drillhole. This would increase by many orders of magnitude the time required to run the algorithm. The semi-greedy approach allows partial covering of a block by a drillhole, something unavailable with integer programming. Partial covering can also be implemented in other heuristic approaches, but then computing time is expected to become an issue due to the iterative nature rather than sequential (as with the semi-greedy approach) of these algorithms.

Our results indicate that relatively small values of *nlist* and *ntrial* are required to get close to the best results reachable by using the semi-greedy algorithm. This finding is interesting as computing time increases with both *nlist* and *ntrial*.

For the particular case of a 0–1 coverage function, the optimality gap of the semi-greedy algorithm varies in the 0–13% range in our examples. This gap is deemed acceptable considering the lack of realism of the 0–1 coverage function, the long computing time of the integer programming approach, and the relative imprecision of the different components (kriging estimate, kriging variance and kriged indicator) appearing in the definition of the *BV* function.

Sensitivity to the choice of *dmin* was shown to be higher for the step functions than for the more realistic linear functions. This finding supports the use of partial covering introduced in this paper. For both step and linear weight functions, increasing *dmin* tends to spread the drillholes more on the periphery of the field. However, this effect is much stronger with the step functions than with the linear functions. The value of *dmin* was selected based on typical separation for definition drilling. It can be adjusted to account for the kind of deposit studied, the perceived geological continuity and the correlation range.

Our tests clearly indicate that our algorithm does not cluster drillholes in high-grade area despite the presence of grade estimate in the *BV* definition (Eq. 1). The weight function ω_1 diminishes strongly the value of blocks surrounding a selected drillhole and thus prevents any risk of clustering.

A simple test showed that it is possible to consider two deposits at once to assist decision making for drilling budget allocation. This requires suitable normalization of the components defining the

*BV*s. More research is needed to determine whether this can be extended to more than two deposits.

The proposed approach can be customized to include additional knowledge about the deposit. For example, if a certain lithology is known to control mineralization, the *BV*s can be calculated only inside the 3D solid of the mineralization, the rest of the *BV*s being set to 0. Metallurgical values or geotechnical parameters can also be incorporated in the *BV*s' definition (Eq. 1).

The new simple semi-greedy algorithm proposed is proven to be an efficient and flexible tool for selection, under budget constraints, of drillholes providing good coverage. Being sequential rather than iterative, the algorithm is fast. It has allowed up to three orders of magnitude CPU time reduction compared to integer programming. It is also flexible as it allows partial coverage of blocks, which is more realistic than coverage functions used in previous studies. The proposed method provides a useful tool to support managers' decision about drillhole allocations in competing projects.

ACKNOWLEDGMENTS

REFERENCES

Bilal, N. (2014). *Métaheuristiques hybrides pour les problèmes de recouvrement et recouvrementpartiel d'ensembles appliqués au problème de positionnement des trous de forage dans les mines.*Ph.D. thesis Polytechnique Montréal.

Bilal, N., Galinier, P., & Guibault, F. (2013). A new formulation of the set covering problem formetaheuristic approaches. *ISRN Operations Research*, 2013.

Bilal, N., Galinier, P., & Guibault, F. (2014). An iterated-tabu-search heuristic for a variant of the partial set covering problem. *Journal of Heuristics, 20*, 143–164.

Chilès, J.-P., & Delfiner, P. (2012). *Geostatistics*. Berlin: Wiley. https://doi.org/10.1002/9781118136188.

Daoust, C., Voicu, G., Brisson, H., & Gauthier, M. (2011). Geological setting of the Paleoproterozoic Rosebel gold district, Guiana shield, Suriname. *Journal of South American Earth Sciences, 32*, 222–245.

Fatehi, M., Haroni, H. A., & Morshedy, A. H. (2017). Designing infill directional drilling in mineral exploration by using

particle swarm optimization algorithm. *Arabian Journal of Geosciences*. https://doi.org/10.1007/s12517-017-3209-4.

Gershon, M., Allen, L. E., & Manley, G. (1988). Application of a new approach for drillholes location optimization. *International Journal of Surface Mining, Reclamation, and Environment, 2,* 27–31.

Hart, J. P., & Shogan, A. W. (1987). Semi-greedy heuristics: An empirical study. *Operations Research Letters, 6,* 107–114.

Jafrasteh, B., & Fathianpour, N. (2017). Optimal location of additional exploratory drillholes using a fuzzy-artificial bee colony algorithm. *Arabian Journal of Geosciences, 10,* 213. https://doi.org/10.1007/s12517-017-2948-6.

Journel, A. G. (1974). Geostatistics for conditional simulation of ore bodies. *Economic Geology, 69,* 673–687.

Journel, A. G. (1982). The indicator approach to estimation of spatial distributions. In *Proceedings of the 17th APCOM international symposium*. New York (pp. 793–806).

Karp, R. M. (1972). Reducibility among combinatorial problems. *In Complexity of computer computations* (pp. 85–103). Berlin: Springer. https://doi.org/10.1007/978-1-4684-2001-2_9.

Kim, Y. C., Myers, D. E., & Knudsen, H. (1977). *Advanced geostatistics in ore reserve estimation and mine planning* (*practitioner's guide*). Technical report Arizona Univ.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220,* 671–680.

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in Applied Probability, 5,* 439–468.

McBratney, A., Webster, R., & Burgess, T. (1981). The design of optimal sampling schemes for local estimation and mapping of regionalized variables—I: Theory and method. *Computers and Geosciences, 7,* 331–334.

Mohammadi, S. S., Hezarkhani, A., & Tercan, A. E. (2012). Optimally locating additional drillholes in three dimensions using grade and simulated annealing. *Journal of the Geological Society of India, 80,* 700–706.

Parker, H. M. (2012). Reconciliation principles for the mining industry. *Mining Technology, 121,* 160–176.

Pilger, G. G., Costa, J. F. C. L., & Koppe, J. C. (2001). Additional samples: Where they should be located. *Natural Resources Research, 10,* 197–207.

Pinheiro, M., Emery, X., Rocha, A. M. A., Miranda, T., & Lamas, L. (2017). Drillholes plans optimization methodology combining geostatistical simulation and simulated annealing. *Tunnelling and Underground Space Technology, 70,* 65–75.

Scheck, D. E., & Chou, D.-R. (1983). Optimum locations for exploratory drillholes. *International Journal of Mining Engineering, 1,* 343–355.

Soltani, S., Hezarkhani, A., Erhan Tercan, A., & Karimi, B. (2011). Use of genetic algorithm in optimally locating additional drillholes. *Journal of Mining Science, 47,* 62–72.

Zagré, G. E., Marcotte, D., Gamache, M., & Guibault, F. (2018). New tabu algorithm for positioning mining drillholes with blocks uncertainty. *Natural Resources Research, 28,* 609–629.