Optimizing a Mineral Value Chain with Market Uncertainty using Benders Decomposition

Jian Zhang[a,*], Barrie R. Nault[a], Roussos G. Dimitrakopoulos[b]

[a] *Haskayne School of Business, University of Calgary, 2500 University DR NW, Calgary, Alberta, Canada*

[b] *COSMO - Stochastic Mine Planning Laboratory, McGill University, 3450 University Street, Montreal, Quebec, Canada*

Abstract

A Benders decomposition-based method is developed to simultaneously optimize the upstream and the downstream of a mineral value chain. Mining blocks representing mineral deposits are dynamically aggregated based on the dual solution of the sub-problem to reduce the complexity in the upstream mine production scheduling. The production schedule obtained based on the aggregated scheduling units is then improved through a moving-window amelioration method. By observing the results of a series of numerical tests, we show that the proposed method effectively optimizes a mineral value chain by synchronizing the upstream mine production scheduling and the downstream material flow and process planning. The results of the numerical tests also show that ignoring the market uncertainty can result in the underestimation of profitability because of the underestimated value of low-grade materials. In order to adapt to the existence of market uncertainty, the stochastic optimizer suggests a higher processing capacity investment in the processing plant and a different long-term mine production schedule.

Keywords: OR in natural resources, Benders decomposition, mineral value chain, market uncertainty

1. Introduction

For the past decade, the inflation of mining costs and uncertainty in commodity market has made the natural resource industry increasingly vulnerable. In response, mining firms need to retrain their focus on long-term planning of the entire value chain including all value-added production operations from the extraction of raw minerals to the delivery of final products (or commodity). A typical mineral value chain (MVC) consists of one or more mines, and a material flow circuit that includes waste dumps, material stockpiles and a processing system transforming raw minerals to commodity.

Usually, optimising a MVC includes the optimization of the upstream mine production schedule (MPS) at

*Corresponding author

Email addresses: `jianzhang@ucalgary.ca` (Jian Zhang), `nault@ucalgary.ca` (Barrie R. Nault), `roussos.dimitrakopoulos@mcgill.ca` (Roussos G. Dimitrakopoulos)

the mines and the optimization of the downstream material flow and processing plan (MFPP) at the material flow circuit. The MPS optimization is to optimize when and where to mine over the planning horizon given a limited mining capacity. Because the valuable element(s) are usually distributed in different areas of a mine deposit and unevenly, a better MPS can increase the firm's profitability significantly by improving the mining sequence when the time value of the money is considered. MPS optimization is challenging for its large number of decision variables and constraints. A mine deposit is modeled by "blocks" and the number of blocks that are used to model a typical mine deposit can be tens of thousands or even millions. Additionally, a large number of slope constraints, or precedence constraints, have to be followed for the excavation of each block. The research on MPS optimization has focused on developing heuristics to find the near optimal solution, and a review of literature on MPS optimization can be found in Hochbaum & Chen (2000) and Newman et al. (2010). The MFPP optimization optimizes the volume of materials that are sent to different destinations after each processing stage. The complexity of a MFPP optimization depends on the structure of MVC and the factors considered. The structure of MVC is determined by the material flow circuit, and can be expanded when other factors, such as outsourcing, closed-loop processing, operational and market uncertainty, are considered.

Because of the complexity of MPS optimization and MFPP optimization described above, it is difficult to optimize MPS and MFPP in a unified model using general solvers for constrained programs. In the literature, two strategies are usually employed for MVC optimization. First, MPS optimization and MFPP optimization are conducted sequentially. That is, when optimizing MPS, the MFPP is ignored, and when optimizing MFPP, the MPS is treated as fixed. When MFPP is ignored, the MPS is optimized based on the value of each block, This is typically computed based on the simple formula, according to Hochbaum & Chen (2000), as:

$$b_i = extract(i) \cdot recovery \cdot price - ore(i) \cdot proc\_cost - weight(i) \cdot mine\_cost,$$

where $extract(i)$ is the weight (in tonnes) of extract contained in block i, $recovery$ is the recovery rate, $price$ is the commodity price (per tonne), $ore(i)$ is the amount of ore contained in block i, $proc\_cost$ is the cost of processing a ton of ore, $weight(i)$ is the weight of block i in tonnes, and $mine\_cost$ is the cost of mining a ton of rock. In this simple setting, the value of a block only depends on the grade and the weight of a block, as well as the recovery rate. The commodity price and the processing cost are assumed not to change with downstream MFPP. The work on MPS optimization with the 'static' block values similar to the definition above can be found from Osanloo et al. (2008), and the recent work on MPS optimization includes Caccetta & Hill (2003), Chicoisne et al. (2012) and Lamghari & Dimitrakopoulos (2012). After the MPS is decided, the material output of each period is fixed, which forms the basis for MFPP optimization. The work on MFPP optimization with fixed production schedule at mines includes Hoerger et al. (1999) and Zhang & Dimitrakopoulos (2017b). The second stream of research on MVC optimization has focused on developing heuristics to find the near optimal solution of a unified MVC model. Epstein et al. (2012)

develop an algorithm that iteratively adds cuts to tighten the linear relaxation of a unified MVC optimization model so that the binary variables in the original model have binary solutions after solving the relaxed model. Montiel & Dimitrakopoulos (2015) use Tabu searching to solve a unified MVC model with multiple processing and transportation options. Goodfellow & Dimitrakopoulos (2016) use both particle swarm optimization and simulated annealing to optimize a general unified MVC model that accounts for geological uncertainty. However, because the solution space increases exponentially as the number of variables increases, it is difficult to guarantee the quality of the obtained solution when the structure of the MVC is complex.

We propose a modified Benders decomposition (BD) method to simultaneously optimize the upstream MPS and the downstream MFPP of a MVC in order to avoid the complexity incurred in the unified MVC model. The recent research on MVC optimization using decomposition-based method includes Blom et al. (2014) and Blom et al. (2016), in which the upstream MPS model and the downstream blending model are simultaneously optimized by iteratively changing the grade and quality targets assigned to the mines. However, the objective of their optimization model is to ensure that the processing rate at each facility is within preset bounds. In the present work, we focus on maximizing the economic value created by the entire MVC. Hence, the upstream MPS model and the downstream MFPP model are simultaneously optimized by dynamically changing the values of raw materials extracted at mines. Zhang & Dimitrakopoulos (2017a) developed a heuristic that dynamically changes the material value to coordinate the optimization of the upstream and the downstream optimizations, but the information obtained at each iteration is not preserved so that the convergence is slow. The BD iteration preserves the information obtained in each iteration by adding cuts generated in all iterations to the master problems.

BD has been widely used to optimize a multistage value chain in different areas and a detailed review of relevant papers can be found in Rahmaniani et al. (2017). Recent research on BD includes the optimizations of supply chain (Keyvanshokooh et al., 2016; Kergosien et al., 2017), smart grid (Soares et al., 2017), networks(de Sá et al., 2017; Mariel & Minner, 2017), and so on. According to the best of our knowledge, there is still a gap in the literature to apply BD in MVC optimization. In the BD proposed herein, the master problem, which is a MPS optimization problem, is reduced by aggregating blocks to larger scheduling units based on the economic value of each block. Because the block value is computed based on the dual solution of the sub-problem, i.e., the downstream MFPP problem, it actually reflects the margin created by the block to the entire value chain given the current MPS and MFPP. As BD proceeds, the value of each block changes dynamically to account for the updated MVC plan.

The paper is organized as follows. In Section 2, a typical MVC is modeled following the convention of the standard form for BD. In Section 3, the solution method based on BD is proposed. In Section 4, a series of numerical tests presented to test the efficiency of the proposed solution method and the importance of integrating of market uncertainty. The conclusions are presented in Section 5.

2. Mathematical formulation

A MVC planning problem is a two-stage profit maximization problem that can be modeled as standard form for BD as

$$\max_{\boldsymbol{x},\boldsymbol{y}} \boldsymbol{c}'\boldsymbol{y} + \boldsymbol{f}'\boldsymbol{x}, \tag{1a}$$

$$\text{s.t. } A\boldsymbol{x} + B\boldsymbol{y} \leq \boldsymbol{b}, \tag{1b}$$

$$\boldsymbol{y} \in \mathbb{Y}, \tag{1c}$$

$$\boldsymbol{x} \geq 0. \tag{1d}$$

In (1), $\boldsymbol{x}$ and $\boldsymbol{y}$ are the vectors of variables. $\boldsymbol{y}$ includes the binary variables that determines the upstream MPS. $\boldsymbol{x}$ includes the continuous variables that determine the downstream MFPP. The object function, (1a), maximizes the expected net present value (NPV) obtained during the planning horizon. $\boldsymbol{c}$ and $\boldsymbol{f}$ are the coefficient vectors relates to $\boldsymbol{y}$ and $\boldsymbol{x}$, respectively, where $\boldsymbol{c}'$ and $\boldsymbol{f}'$ represent the transposed vectors. The relationship between $\boldsymbol{y}$ and $\boldsymbol{x}$ is defined in (1b), where A and B are the left-hand-side (LHS) coefficient matrices for $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and $\boldsymbol{b}$ is the right-hand-side (RHS) constant vector. In (1c), $\mathbb{Y}$ defines the constraints on $\boldsymbol{y}$.

In order to present our method clearly, we use an example MVC to illustrate the general model proposed above. The example MVC contains a mine, a number of stockpiles, and a processing plant. However, the method proposed herein is general and applicable in a multi-mine MVC with a complex material flow circuit. Following the convention in the general form above, we define the symbols as in Table 1.

[Table 1 about here.]

The MILP that maximizes the MVC's expected NPV can be formulated as

$$\text{Maximize } \underbrace{-\sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{I}\frac{1}{[1+\gamma]^t}c_i^M q_{ij} y_{it}}_{(i)}$$
$$+\underbrace{\frac{1}{S}\sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{j=1}^{J}\frac{1}{[1+\gamma]^t}\left[\left[p_{ts}g_j - c^P\right]\left[x_{jts}^{MP} + x_{jts}^{HP}\right] - c^R x_{jts}^{HP} - c^H x_{jts}^{H}\right]}_{(ii)}, \tag{2a}$$

$$x_{jts}^{MP} + x_{jts}^{MH} - \sum_{i=1}^{I} q_{ij} y_{it} \leq 0, \qquad \forall j, \forall t, \forall s, \tag{2b}$$

$$\sum_{t=1}^{T} y_{it} \leq 1, \qquad \forall i, \tag{2c}$$

$$y_{it} - \sum_{\hat{t}=1}^{t} y_{\varepsilon\hat{t}} \leq 0, \qquad \forall i, \forall t, \forall \varepsilon \in \mathbb{P}_i, \tag{2d}$$

$$\sum_{I=1}^{I}\sum_{j=1}^{J} q_{ij} y_{it} \leq \overline{y}, \qquad \forall t, \tag{2e}$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J} q_{ij} y_{it} - \sum_{i=1}^{I}\sum_{j=1}^{J} q_{ij} y_{i,t-1} \leq \xi, \qquad \forall t \in \{2, \ldots, T\}, \forall s, \tag{2f}$$

$$y_{it} \in \{0, 1\}, \qquad \forall i, \forall t \tag{2g}$$

$$x_{j1s}^{H} - x_{j1s}^{MH} \leq 0, \qquad \forall j, \forall s, \tag{2h}$$

$$x_{jts}^{H} - x_{j,t-1,s}^{H} - x_{jts}^{MH} + x_{jts}^{HP} \leq 0, \qquad \forall j, \forall t \in \{2, \ldots, T\}, \forall s, \tag{2i}$$

$$\sum_{j=1}^{J} \left[x_{jts}^{MP} + x_{jts}^{HP}\right] - x^{P} \leq 0, \qquad \forall t, \forall s, \tag{2j}$$

$$\sum_{j=1}^{J} \left[g_j - \overline{g}^{P}\right] \left[x_{jts}^{MP} + x_{jts}^{HP}\right] \leq 0, \qquad \forall t, \forall s, \tag{2k}$$

$$\sum_{j=1}^{J} \left[\underline{g}^{P} - g_j\right] \left[x_{jts}^{MP} + x_{jts}^{HP}\right] \leq 0, \qquad \forall t, \forall s, \tag{2l}$$

$$x_{jts}^{MP}, x_{jts}^{MH}, x_{jts}^{HP}, x_{jts}^{H}, x^{P} \geq 0 \qquad \forall i, \forall t, \forall s. \tag{2m}$$

In the MVC optimization model (2), the objective function (2a) is to maximize the expected NPV generated by the MVC over the planning horizon. In the objective function, (i) corresponds to $\boldsymbol{c}'\boldsymbol{y}$ in (1a) and computes the total discounted mining cost. (ii) corresponds to $\boldsymbol{f}'\boldsymbol{x}$ in (1a) and computes the total expected discounted value generates by the downstream processing system. (2b) corresponds to (1b) that links $\boldsymbol{y}$ and $\boldsymbol{x}$, and it constrains that in each period, the total amount of materials sent to the downstream should not exceed the total amount of materials mined in the upstream. (2c) constrains that a block is only mined in a single period. (2d) constrains that any block is mined after its predecessors. (2e) constrains the total number of blocks mined in each period to be within the mining capacity. (2f) ensures the feasibility of the obtained schedule, where ξ is a preset smooting parameter. That is, the mining rate in each period should not excceed the mining rate in the previous period by too much. In the mining industry, dramatic fluctuation in production rate is not feasible because it incurs high operating cost by dismissing and recruiting workers, idling production equipment, and so on. Other indirect costs might also be caused due to environmental or political reasons, such as waste processing, employment rate, and so on. Because it is hard to quantify the cost and impact caused by the dramatic fluctuation in the production rate, the smoothing parameter is added to avoid costly and infeasible solutions. (2h) and (2i) constrain the levels of stockpiled materials at the end of each period. (2j) constrains the amount of materials processed at each period to be within the processing capacity invested at the beginning of the planning horizon. (2k) and (2l) constrain the grade of the blended materials processed during each period to be within a required range.

In the MVC optimization, because the raw materials mined upstream, no matter whether ore or waste, always have feasible destinations such as wasted piles, processing plants, stockpiles, and so on, we make the following assumption:

Assumption 1. *In the MVC planning problem, (1), there is always a feasible downstream MFPP, $\boldsymbol{x}$, given any feasible upstream MPS, $\boldsymbol{y} \in \mathbb{Y}$.*

Assumption 1 is true in MVC and other two-stage value chain planning problem if there is an uncapacitated channel, e.g., the waste pile herein, to dispose the unwanted yields from upstream. Note that the waste processing cost is ignored in the example MVC without loss of generality. In the case when the upstream problem has feasible solutions that make the downstream problem infeasible, we can 'soften' certain constraints of the downstream problem by adding penalties if those constraints are violated.

3. Solution method

Our solution method uses Benders decomposition (BD) to simultaneously optimize the decomposed MVC model. In each iteration, the master problem is reduced by aggregating the blocks dynamically based on the dual solution of the sub-problem. When the reduced master problem is solved, a moving window amelioration is performed to improve the obtained MPS.

3.1. Benders decomposition

In the original program (1), when the upstream schedule, $\boldsymbol{y}$, is fixed to $\hat{\boldsymbol{y}}$, we can obtain the primal sub problem as

$$\max_{\boldsymbol{x}} \boldsymbol{f}'\boldsymbol{x}, \tag{3a}$$

$$\text{s.t. } A\boldsymbol{x} \leq \boldsymbol{b} - B\hat{\boldsymbol{y}}, \tag{3b}$$

$$\boldsymbol{x} \geq 0. \tag{3c}$$

The dual of the sub problem can be formed as

$$\min_{\boldsymbol{u}} [\boldsymbol{b} - B\hat{\boldsymbol{y}}]'\boldsymbol{u}, \tag{4a}$$

$$A'\boldsymbol{u} \geq \boldsymbol{f}, \tag{4b}$$

$$\boldsymbol{u} \geq 0. \tag{4c}$$

According to Assumption 1, because there is always a feasible solution of $\boldsymbol{x}$ for the primal sub problem given any feasible MPS, $\boldsymbol{y}$, the dual of the sub problem is always bounded. Hence, the BD steps for MVC optimization can be modified as

Step 1. Initialize $\hat{\boldsymbol{y}}$ subject to (1c). Set the lower bound as $LB := -\infty$, the upper bound as $UB := +\infty$, and the gap limit ϵ to a positive value.

Step 2. WHILE $UB - LB > \epsilon$ DO

Solve the dual of the sub problem, (4), and get the extreme point, $\hat{\boldsymbol{u}}$.

Add cut

$$\pi \leq \boldsymbol{c}'\boldsymbol{y} + [\boldsymbol{b} - B\boldsymbol{y}]'\hat{\boldsymbol{u}}. \tag{5}$$

$LB := \max\{LB, \boldsymbol{c}'\hat{\boldsymbol{y}} + [\boldsymbol{b} - B\hat{\boldsymbol{y}}]'\hat{\boldsymbol{u}}\}$.

Solve the master problem as

$$\hat{\pi} := \max_{\boldsymbol{y},\pi} \pi \ni \text{cuts}, \boldsymbol{y} \in \mathbb{Y}. \tag{6}$$

$UB := \hat{\pi}$.

END While.

Given the example MVC model (2), $[\boldsymbol{b} - B\boldsymbol{y}]'\hat{\boldsymbol{u}}$ can be obtained as

$$[\boldsymbol{b} - B\boldsymbol{y}]'\hat{\boldsymbol{u}} = \sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{I} \hat{u}_{jts} q_{ij} y_{it}.$$

Thus, in each iteration, the newly added cut, (5), can be obtained as

$$\pi \leq \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{I} \left[\sum_{s=1}^{S} \hat{u}_{jts} - \frac{c_i^M}{[1+\gamma]^t} \right] q_{ij} y_{it}. \tag{7}$$

Let $\hat{u}_{jts}^k$ be the solution of the dual of the sub problem in the kth iteration. Then, the master problem, (6), for our example MVC model has the form of

$$\text{Maximize } \pi,$$
$$\text{s.t. } \pi \leq \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{I} \left[\sum_{s=1}^{S} \hat{u}_{jts}^k - \frac{1}{[1+\gamma]^t} c_i^M \right] q_{ij} y_{it}, \quad \forall k,$$
$$\text{(2c)-(2g)}.$$

After the decomposition, we can iteratively solve the master problem and the subproblem using different solvers. As the subproblem can be solved using common LP solvers such as CPLEX, we focus on the heuristics for the master problem herein.

3.2. Generating dynamic bench-pushbacks

From the form of the master problem, we observe that $I \times T$ binary variables are included, which is still too many to solve. Our approach is to reduce the number of binary variables by aggregating blocks to bigger scheduling units. Aggregation has been used widely in MPS optimization (e.g., Ramazan, 2007; Newman & Kuchta, 2007; Weintraub et al., 2008; Boland et al., 2009; Tabesh & Askari-Nasab, 2011; Jélvez et al., 2016, etc.).

We use a dynamic bench-pushback aggregation strategy that is generated based on the economic value of each block. The bench-pushback aggregation based on static block values is summarized in Chicoisne et al. (2012) in which the bench-pushbacks are referred as bench-phases. First, a series of nested pits are generated by gradually reducing the values of all blocks and generates a sequence of reducing *ultimate pit limits*. The pushbacks (or phases) are formed based on the obtained nested pits. Each pushback is then divided into bench-pushbacks based on the vertical levels (or 'benches'). Finally, a new mathematical model is formulated to optimize the extraction schedule of bench-pushbacks rather than blocks. The predecessors of a bench-pushback include the bench-pushbacks with an equal or higher vertical level and within the same or smaller nested pits. Figure 1 shows the definition of nested pits, pushbacks, benchphases and an example of precedence constraints. The area between any two neighboring pit shells is a pushback, and the area with the same letter is a bench-pushback. As an example, the predecessors of bench-pushback 'g' are bench-pushbacks 'a', 'b', 'c', 'd' and 'f'. However, in our mathematical formation, only the direct predecessors, 'd' and 'f', are specified in the precedence constraints. After bench-pushback aggregation, the number of binary variables is reduced and each aggregated scheduling unit has no more than two precedence constraints.

[Figure 1 about here.]

From (7), we observe that $\sum_{j=1}^{J}\left[[1+\gamma]^t\sum_{s=1}^{S}\hat{u}_{jts}-c_i^M\right]q_{ij}$ actually indicates the value created by block i if it is mined in period t, denoted by $v_{it}(\hat{\boldsymbol{u}})$. When t is unknown, the expected value created by block i, denoted by $v_i(\hat{\boldsymbol{u}})$, can be obtained as

$$v_i(\hat{\boldsymbol{u}}) = \frac{1}{T}\sum_{t=1}^{T} v_{it}(\hat{\boldsymbol{u}}).$$

Because v_i is a function of $\hat{\boldsymbol{u}}$, in our dynamic bench-pushback aggregation method, the bench-pushbacks are regenerated in each Benders iteration when $\hat{\boldsymbol{u}}$ is updated.

When the values of the blocks are available, the nested-pits can be generated by gradually reducing the block values. Suppose that N nested-pits are generated. A sequence of scalar factors, $\lambda_1 > \lambda_2 > \cdots > \lambda_N$, are introduced to reduce the block values. Let b_i^n be the binary variable that indicates if block i belongs to pit n. Then, pit n can be generated by solving the binary program

$$\text{Maximize} \sum_{i=1}^{I}\left[v_i(\hat{\boldsymbol{u}}) - \lambda_n p\sum_{j=1}^{J} g_j q_{ij}\right] b_i^n, \tag{8a}$$

$$\text{s.t. } b_\varepsilon^n \geq b_i^n, \qquad \forall \varepsilon \in \mathbb{P}_i, \tag{8b}$$

$$b_i^n \in \{0,1\}, \qquad \forall i. \tag{8c}$$

In (8a), $v_i(\hat{\boldsymbol{u}}) - \lambda_n p \sum_{j=1}^{J} g_j q_{ij}$ computes the value of block i that is used to generate pit n. (8b) is the precedence constraint.

In order to control the size of pushbacks, we set a size limit for each pushback. If the size of a pushback is lower than the size limit, the pushback is merged with the next pushback. The merge continues until the size of the size of obtained pushback reaches the size limit. When all the pushbacks are generated, the bench-pushbacks are generated as described earlier.

Note that because the bench-pushback generation is a heuristic method, the upper bound obtained by solving the reduced master problem might be less than the true upper bound (UB) obtained by solving the original master problem that is at block-level. Hence, it is possible that the obtained heuristic upper bound, denoted by (HUB), falls below the lower bound (LB) in our Benders' iteration. In the proposed BD method, we do not use the classic stop criterion of BD iteration that is based on the gap between UB and LB. Instead, we stop the iteration when LB stops increasing for a certain number of iterations.

3.3. Moving-window amelioration

When the scheduling units are aggregated from blocks to bench-pushbacks, the mining capacity in a period may not be fully utilized because some larger scheduling units might not fit in the smaller unused mining capacity. In order to resolve the issue, we perform a moving-window amelioration (MWA) on the obtained bench-pushback level schedule. We use ProcFR(t_0, t_1) to denote the procedure of rescheduling blocks from t_0 to t_1 that increase the overall NPV. ProcFR(t_0, t_1) includes two steps. First, a partially-ordered knapsack (Kolliopoulos & Steiner, 2007) model is formed to find the blocks to be rescheduled from period t_0 to period t_1. Let $\mathbb{I}_{t_0}$ denote the set of blocks that are currently scheduled in period t_0, and $\overline{y}_{t_1}$ denote the current available capacity in period t_1. Then, the model can be formed as

$$\text{Maximize } \; z, \tag{9a}$$

$$\text{s.t. } z \leq \sum_{i \in \mathbb{I}_{t_0}} \left[\frac{1}{[1+\gamma]^{t_1}} v_{it_1}(\hat{\boldsymbol{u}}^k) - \frac{1}{[1+\gamma]^{t_0}} v_{it_0}(\hat{\boldsymbol{u}}^k) \right] y_i, \quad \forall k, \tag{9b}$$

$$[t_0 - t_1] y_i \leq [t_0 - t_1] y_\varepsilon, \quad \forall i \in \mathbb{I}_{t_0}, \forall \varepsilon \in \mathbb{P}_i \cap \mathbb{I}_{t_0}, \tag{9c}$$

$$\sum_{i \in \mathbb{I}_{t_0}} y_i \leq \overline{y}_{t_1}, \tag{9d}$$

$$y_i \in \{0, 1\}, \quad \forall i \in \mathbb{I}_{t_0}. \tag{9e}$$

In the a partially-ordered knapsack model above, the objective value z is determined by the cuts generated in the iteration as in (9b) where $\hat{\boldsymbol{u}}^k$ is the solution of DualSP in the kth iteration. (9c) is the precedence constraints given t_0 and t_1. If $t_0 < t_1$, then $y_i \geq y_\varepsilon$; otherwise, if $t_0 > t_1$, then $y_i \leq y_\varepsilon$. (9d) constrains that the total number of blocks to be moved to period t_1 should not exceed the available capacity in period t_1. Note that the available capacity in t_1 should be adjusted so that the smooth constraints in (2f) are not violated. After the partially-ordered knapsack model (9) is solved, any block i with $y_i = 1$ in the optimal solution is rescheduled from period t_0 to t_1.

The steps of MWA can be summarized as

MWA:

FOR $t = 1$ TO T

  IF $t > 1$ THEN

    ProcFR$(t, t-1)$.

  END IF

  IF $t < T$ THEN

    ProcFR$(t, t+1)$.

  END IF

Next t

4. Numerical Test

In this section, three independent hypothetical cases are designed to test our proposed MVC optimization method. In the first case, the bench-pushback scheduling method with moving-window amelioration (BPMWA) proposed in Sections 3.2-3.3 is tested. We compare BPMWA against a commercial software in its capability of solving the upstream MPS when the economic value of each block is given. In the second case, the performance of the proposed BD-based simultaneous MVC optimization method (BDSimO) is tested by comparing it with a sequential MVC optimization method. In the last case, the importance of integrating market uncertainty is tested and a managerial insight is extracted from the results. Our proposed method is programmed using Matlab R2015b and CPLEX 12.51, and tested on a platform of Intel Xeon X5650 with two 2.67GHz processors and 24.0GB RAM.

4.1. The performance of BPMWA

Because BPMWA runs in each iteration to solve the master problem, if it does not generate a MPS with a good quality, the error will accumulate over iterations. Thus, we first design a hypothetical case by which the proposed BPMWA is compared with GEOVIA Whittle 4D, a commercial software that is widely used in mining industry. Because Whittle 4D is not capable of solving a MPS optimization problem with Benders cuts, it is not possible to compare BPMWA and Whittle 4D in each iteration. Hence, the comparison is based on a common MPS optimization problem without Benders cuts as

$$\text{Maximize} \sum_{i=1}^{I} \frac{1}{[1+\gamma]^t} v_i y_{it}, \tag{10}$$

Subject to (2c)-(2g).

The block value, v_i, in (10) is static as $v_i = \max\left\{\sum_{j=1}^{J}\left[p_0 g_j - c^P\right] q_{ij}, 0\right\} - c_i^M q_{ij}$, which indicates that block i is only sent for processing when the gain obtained from the product is higher than processing cost.

We use the BPMWA and Whittle 4D to optimize the MPS of a copper deposit that includes 40,044 blocks and 17 predecessors are considered for each block. 20 geological simulations of the orebody models are available as in Figure 2, where the yellow color indicates the amount of copper contained in the block. The 3D-view model shows the average grade of each block and the top-view models with contour lines show the grade of each block in different simulations. We conduct 20 comparisons, each of which is based on a particular simulated orebody scenario. A single valuable element is considered in the test and 15 bins are set on the distributed material grade to categorize the materials into $J = 15$ types. For any type j material, the grade is set to $g_j = 0.001 + 0.02 \times [j-1]$. Block i belongs to type j, i.e., q_{ij} equals to the tonnage of the block, if its grade falls in $[g_j - 0.001, g_j + 0.001)$. If a block has a grade more than 3%, it belongs to type J. The cost of mining a block is ranged as $c_i^M \in [2, 2.5]$ (\$ per tonne). The mining capacity, the unit processing cost and the discount rate are set to $\overline{y} = 6500$ (blocks per year), $c^P = 15$ (\$ per tonne) and $\gamma = 0.1$, respectively. In the current hypothetical case, we ignore the market uncertainty and set a constant commodity price as $p_t = 3,000$ (\$ per tonne) for all t.

[Figure 2 about here.]

We set Whittle 4D to generate MPS according to its default 'best-case scheduling strategy'. In this scheduling strategy, a sequence of nested pits are generated by dynmically reducing the value of each mining block. The inner pits are mined earlier and each pit is mined from top to bottom. The number of nested pits generated determines the quality of schdule. For comparison, we set the number of pits to 100, which is the maximum number allowed by Whittle 4D. For BPMWA, the size limit of pushbacks is set to 1500 blocks and the bench height is set to 5 times block height. Figure 3 shows the NPVs of the mine production schedules generated by the two methods for 20 simulations. It can be observed that BPMWA generates better schedules and creates a 1.11% higher NPV on average. Note that our method uses more computation time because Wittle 4D generates nested-pits using a customized Lerch-Grossman algorithm, which is not employed in our work. Since the MPS optimization problem is a strategic optimization problem, the difference of computation time can be ignored.

[Figure 3 about here.]

Figure 3 also shows that, before applying MWA, the best-case scheduling strategy of Whittle 4D generates better results than the bench-pushback methods. However, we cannot employ the same method because, in the best-case scheduling strategy of Whittle 4D, the mining rate is always maximized by fully utilizing the mining capacity, which is not the optimal strategy when the block values fluctuate in different periods.

The comparison results presented herein shows that BPMWA is on a par with the commercialized software in its search ability. Because, when the Benders cuts are added, the MPS optimization problem is only tightened by considering multiple scenarios of block values, the quality of MPS obtained by BPMWA will not be impacted except that the computation time will be increased.

4.2. The performance of BDSimO

The BDSimO is tested in a hypothetical case in which a new MVC is to be constructed to maximize the NPV of a newly found copper deposit as described in Section 4.1. The BDSimO is compared with a sequential optimization strategy (SeqO). In SeqO, the long-term MPS is optimized without considering the downstream MFPP and the downstream MFPP is designed after the upstream MPS is fixed. Given that the test is based on the same copper deposit used in Section 4.1, the same model, (10), is formed to generate the MPS for SeqO. Because the test focuses on the ability to coordinate the optimizations of the upstream MPS and the downstream MFPP, the geological uncertainty is ignored and the grade of each block is set to the average of 20 simulations.

Besides the parameters that have been assigned in Section 4.1, the parameters for the hypothetical case are set as follows. The investment required for building a unit processing capacity is $\tau = 50$ ($ per tonne). The lower bound and the upper bound of the material grade required for processing are $\underline{g} = 0.002$ and $\overline{g} = 0.03$, respectively. For the stockpiles, the holding cost and the rehandling cost are $c^H = 0.2$ ($ per tonne year) and $c^R = 1$ ($ per tonne), respectively.

In order to implement BDSimO, we use the MPS in SeqO as the initial solution for the master problem in the proposed BD method. In order to test the superiority of dynamic bench-pushbacks, we compare the dynamic method, namely BDSimO(D), and the static method, namely BDSimO(S). In BDSimO(S), the bench-pushbacks are not updated in each iteration of BD. The performances of BDSimO(D) and BDSimO(S) are compared in Table 2. We observe that, in the test, both BDSimO(S) and BDSimO(D) use 4 iterations and require less than an hour to converge. As the LB indicates the objective value, we find that BDSimO(D) generates a MVC plan with a 5.11% higher NPV, which is more than $69 million. The test results show the superiority of the dynamic bench-pushback generation and the efficiency of the BDSimO.

[Table 2 about here.]

In order to show the ability of BDSimO(D) in simultaneous optimization, the MVC plans obtained from

SeqO and BDSim(D) are compared. Figure 4 compares the materials extracted from the MPSs generated by SeqO and BDSim(D), respectively. We observe that, in SeqO, the upstream mining rate is maximized without considering the downstream congestion. In BDSimO(D), the upstream MPS and the downstream MFPP are coordinated, whereby the upstream mining rate is reduced to account for the processing capacity downstream. Figures 5 and 12 compare the downstream MFPPs under BDSimO(D) and SeqO. We find that the stockpile level in each year is significantly reduced after BDSimO(D) is employed, which provides evidence for the coordination of MVC. The investment in processing capacity in Figure 12 is reduced from 1.85×10^7 to 1.51×10^7, while the capacity utilization is increased in the later peiords, i.e., Years 15-20. Consequently, as shown in Figure 7, the MVC plan generated by BDSimO(D) creates a 65.42% higher NPV over the planning horizon, which is approximately $563 million .

[Figure 4 about here.]

[Figure 5 about here.]

[Figure 6 about here.]

[Figure 7 about here.]

4.3. The importance of integrating market uncertainty

We simulate 50 market scenarios where the copper price ($ per tonne) fluctuates as shown in Figure 8. The prices are adjusted to ensure that the average of simulated prices in each period equals to 3,000 ($ per tonne) so that the change in the optimal MVC plan is not caused by the prediction of a better or worse market for copper.

[Figure 8 about here.]

After implementing BDSimO(D), the stochastic MVC plan (SMVCP) that considers market uncertainty is compared with the deterministic MVC plan (DMVCP) obtained in Section 4.2. Table 3 shows the iteration process of BDSimO(D) when market uncertainty is considered. Compared with Table 2, we observe that the number of iterations and the computation time required is random and does not differ based on whether the market uncertainty is considered.

[Table 3 about here.]

Figure 9 compares the cashflows and the cumulative NPVs of DMVCP and SMVCP, respectively. The test result shows that, when the market uncertainty is considered, the optimizer estimates a 2.36% higher expected NPV, which is about \$33.6 million. This difference is caused by the underestimated value of the low grade material in the deterministic optimization when the commodity price is decreased significantly. The example in Figure 10 shows how the low-grade material is underestimated in the deterministic optimization. In the example, the horizontal axis shows the value created by extracting certain low-grade material and converting it to a commodity. In the deterministic optimization, when the material value computed from the expected commodity price is negative, the optimizer will treat the material as waste. However, because the material can be disposed of when the commodity price drops below the processing costs to prevent further loss, it actually creates a positive expected value. Hence, in Figure 10, the material is treated as waste when using deterministic commodity price and is treated as ore when market uncertainty is considered. As discussed earlier, since the computation time is not a concern, it is well worth integrating market uncertainty in long-term MVC planning to find a more reliable estimation.

[Figure 9 about here.]

[Figure 10 about here.]

Figures 11 shows the materials mined at each period in SMVCP. By comparing the materials mined in SMVCP and DMVCP, we can see that more low-grade materials are mined in SMVCP. Consequently, 29,232 blocks are mined in DMVCP and 30,094 blocks are mined in SMVCP, which means that SMVCP estimates a 2.95% larger pit at the end of planning horizon. Because the difference is mainly caused by the low-grade blocks, the difference in the expected NPV is less than the difference in pit sizes. The comparison between Figures 11 and 4b also shows that, when market uncertainty is considered, the mining rate after the second year is significantly increased and the life of mine is reduced. This change of optimal mine production schedule is for the purpose of increasing the stockpile level to increase the flexibility of the MVC to deal with market uncertainty.

[Figure 11 about here.]

The average downstream processing and stockpiling amounts expected when market uncertainty is considered is shown in Figure 12a and 12b. By comparing Figures 12a and 12b with Figures 6b and 5b, respectively, we can observe that, when market uncertainty is considered, the optimal processing capacity, which is invested at the beginning of the planning horizon, is increased and the average stockpile levels in the later periods are increased. These changes also serve to increase the flexibility of the MVC so that the mining company can adjust its commodity production with more freedom to deal with market uncertainty.

[Figure 12 about here.]

5. Conclusions

In the present work, a Bender-decomposition based mineral value chain optimization method is developed to simultaneously optimize the upstream MPS and the downstream MFPP. Based on the dual solution solved in each iteration of the proposed BD method, the blocks are aggregated to reduce the complexity of solving the upstream MPS. A moving-window amelioration method is also developed to fix the schedule generated based on the aggregated scheduling units, i.e., bench-pushbacks. By observing the results of a series of numerical tests based on the practical-scale hypothetical cases, three conclusions can be drawn. First, the proposed aggregation and amelioration methods can generate good-quality mine production schedules with reasonable computation times, and the dynamic method outperforms the static method. Second, the proposed Benders-decomposition based method can effectively synchronize the optimizations upstream and downstream in a mineral value chain. Finally, as a management insight, ignoring market uncertainty can result in the underestimation of profitability and a suboptimal mineral value chain plan. In order to adapt to market uncertainty, the stochastic optimizer suggest a higher investment in the capacity of the processing plant and a long-term mine production schedule with a higher production rate and a shorter life of mine. The suggested mine production schedule incurs a relatively higher stockpile levels but it enables a greater flexibility in changing the commodity production to accommodate market fluctuations.

6. Acknowledgements

Funding was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant 239019 and mining industry partners of the COSMO lab (AngloGold Ashanti, Barrick Gold, BHP Billiton, De Beers Canada, Kinross Gold, Newmont Mining and Vale). The first author was at COSMO from 2013-2017.

References

Blom, M. L., Burt, C. N., Pearce, A. R., & Stuckey, P. J. (2014). A decomposition-based heuristic for collaborative scheduling in a network of open-pit mines. *INFORMS Journal on Computing*, *26*, 658–676.

Blom, M. L., Pearce, A. R., & Stuckey, P. J. (2016). A decomposition-based algorithm for the scheduling of open-pit networks over multiple time periods. *Management Science*, .

Boland, N., Dumitrescu, I., Froyland, G., & Gleixner, A. M. (2009). Lp-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. *Computers & Operations Research*, *36*, 1064–1089.

Caccetta, L., & Hill, S. P. (2003). An application of branch and cut to open pit mine scheduling. *Journal of global optimization*, *27*, 349–365.

Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E., & Rubio, E. (2012). A new algorithm for the open-pit mine production scheduling problem. *Operations Research*, *60*, 517–528.

Epstein, R., Goic, M., Weintraub, A., Catalán, J., Santibáñez, P., Urrutia, R., Cancino, R., Gaete, S., Aguayo, A., & Caro, F. (2012). Optimizing long-term production plans in underground and open-pit copper mines. *Operations Research*, *60*, 4–17.

Goodfellow, R. C., & Dimitrakopoulos, R. (2016). Global optimization of open pit mining complexes with uncertainty. *Applied Soft Computing*, *40*, 292 – 304. doi:http://dx.doi.org/10.1016/j.asoc.2015.11.038.

Hochbaum, D. S., & Chen, A. (2000). Performance analysis and best implementations of old and new algorithms for the open-pit mining problem. *Operations Research*, *48*, 894–914.

Hoerger, S., Seymour, F., & Hoffman, L. (1999). Mine planning at newmont's nevada operations. *Mining engineering*, *51*, 26–30.

Jélvez, E., Morales, N., Nancel-Penard, P., Peypouquet, J., & Reyes, P. (2016). Aggregation heuristic for the open-pit block scheduling problem. *European Journal of Operational Research*, *249*, 1169–1177.

Kergosien, Y., Gendreau, M., & Billaut, J.-C. (2017). A benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints. *European Journal of Operational Research*, *262*, 287–298.

Keyvanshokooh, E., Ryan, S. M., & Kabir, E. (2016). Hybrid robust and stochastic optimization for closed-loop supply chain network design using accelerated benders decomposition. *European Journal of Operational Research*, *249*, 76–92.

Kolliopoulos, S. G., & Steiner, G. (2007). Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics*, *155*, 889–897.

Lamghari, A., & Dimitrakopoulos, R. (2012). A diversified tabu search approach for the open-pit mine production scheduling problem with metal uncertainty. *European Journal of Operational Research*, *222*, 642–652.

Mariel, K., & Minner, S. (2017). Benders decomposition for a strategic network design problem under nafta local content requirements. *Omega*, *68*, 62–75.

Montiel, L., & Dimitrakopoulos, R. (2015). Optimizing mining complexes with multiple processing and transportation alternatives: An uncertainty-based approach. *European Journal of Operational Research*, .

Newman, A. M., & Kuchta, M. (2007). Using aggregation to optimize long-term production planning at an underground mine. *European Journal of Operational Research*, *176*, 1205–1218.

Newman, A. M., Rubio, E., Caro, R., Weintraub, A., & Eurek, K. (2010). A review of operations research in mine planning. *Interfaces*, *40*, 222–245.

Osanloo, M., Gholamnejad, J., & Karimi, B. (2008). Long-term open pit mine production planning: a review of models and algorithms. *International Journal of Mining, Reclamation and Environment*, *22*, 3–35.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*, 801–817.

Ramazan, S. (2007). The new fundamental tree algorithm for production scheduling of open pit mines. *European Journal of Operational Research*, *177*, 1153–1166.

de Sá, E. M., Morabito, R., & de Camargo, R. S. (2017). Benders decomposition applied to a robust multiple allocation incomplete hub location problem. *Computers & Operations Research*, .

Soares, J., Canizes, B., Gazvhini, M. A. F., Vale, Z., & Venayagamoorthy, G. (2017). Two-stage stochastic model using benders' decomposition for large-scale energy resources management in smart grids. *IEEE Transactions on Industry Applications*, .

Tabesh, M., & Askari-Nasab, H. (2011). Two-stage clustering algorithm for block aggregation in open pit mines. *Mining Technology*, *120*, 158–169.

Weintraub, A., Pereira, M., & Schultz, X. (2008). A priori and a posteriori aggregation procedures to reduce model size in mip mine planning models. *Electronic Notes in Discrete Mathematics*, *30*, 297–302.

Zhang, J., & Dimitrakopoulos, R. G. (2017a). A dynamic-material-value-based decomposition method for optimizing a mineral value chain with uncertainty. *European Journal of Operational Research*, *258*, 617 – 625. doi:http://dx.doi.org/10.1016/j.ejor.2016.08.071.

Zhang, J., & Dimitrakopoulos, R. G. (2017b). Stochastic optimization for a mineral value chain with nonlinear recovery and forward contracts. *Journal of the Operational Research Society*, . doi:10.1057/s41274-017-0269-5.

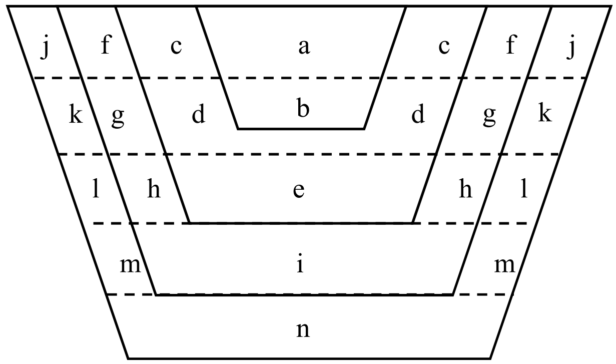


Figure 1: Bench-pushbacks of an open-pit mine.

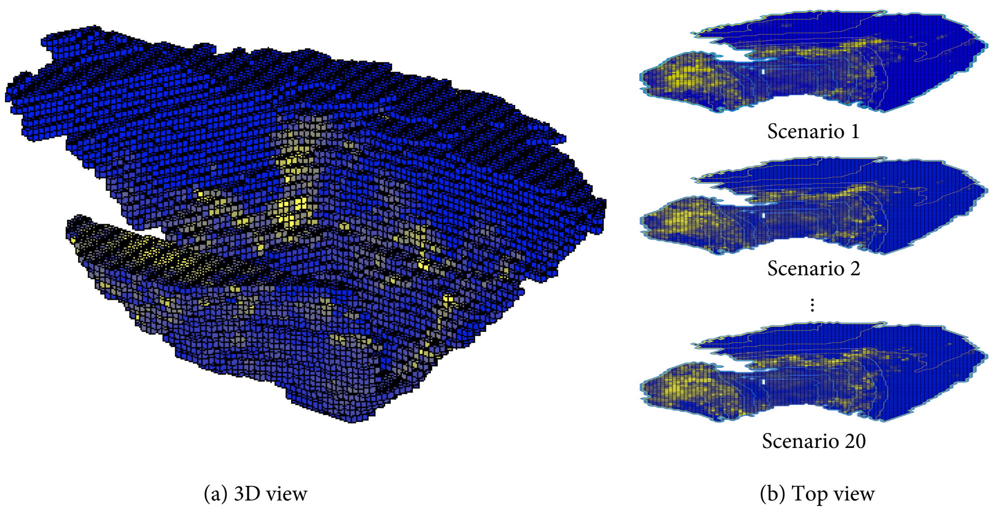


Figure 2: The 3D view of the orebody with average grade and the top view of the orebodies with random grades.

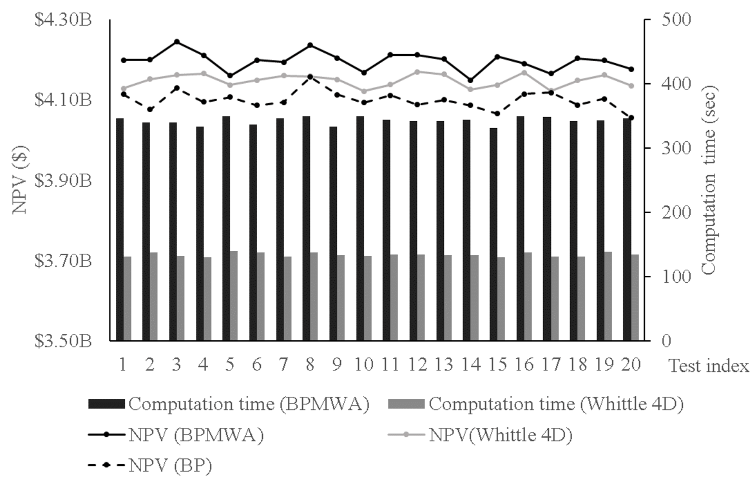
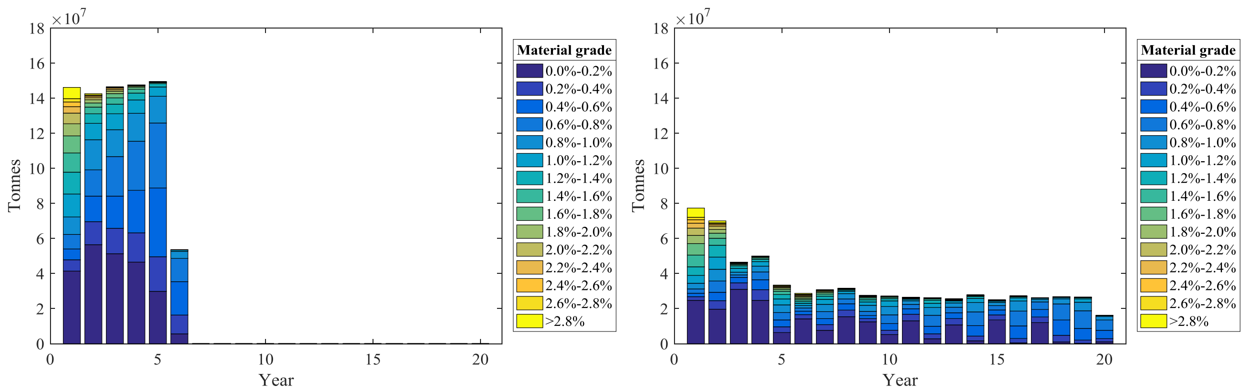


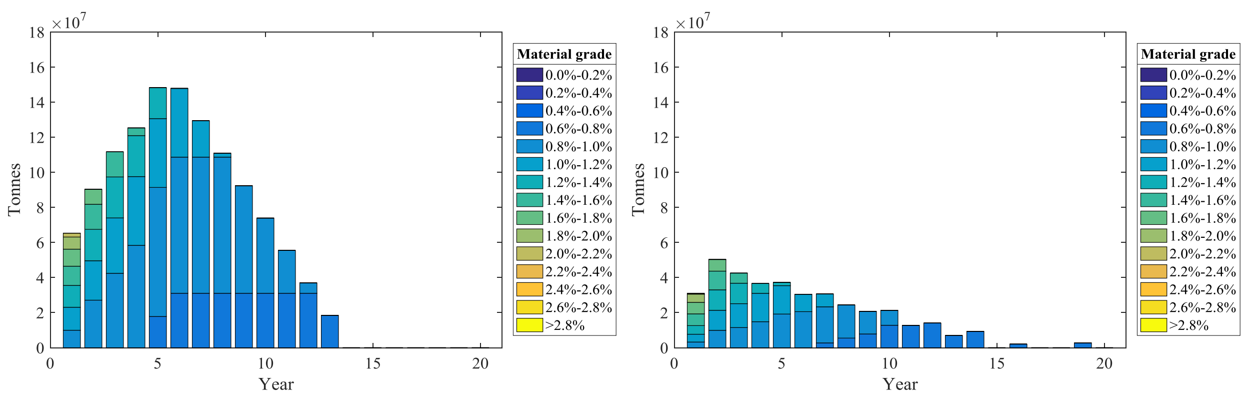
Figure 3: Comparing Whittle 4D and BPMWA.



(a) SeqO

(b) BDSimO(D)

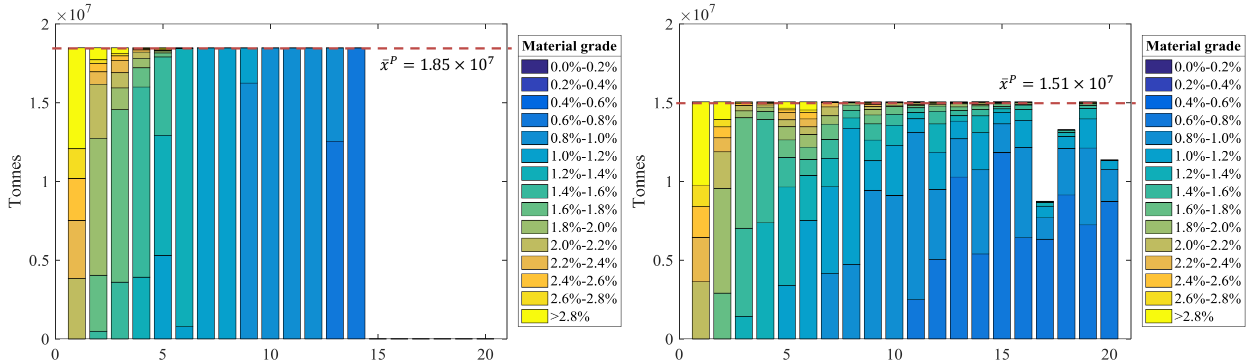
Figure 4: The raw materials extracted from the MPSs generated by both methods.



(a) SeqO

(b) BDSimO(D)

Figure 5: The optimal stockpile levels incurred by the MPSs generated by both methods.



(a) SeqO

(b) BDSimO(D)

Figure 6: The optimal processing rates based on the MPSs generated by both methods.

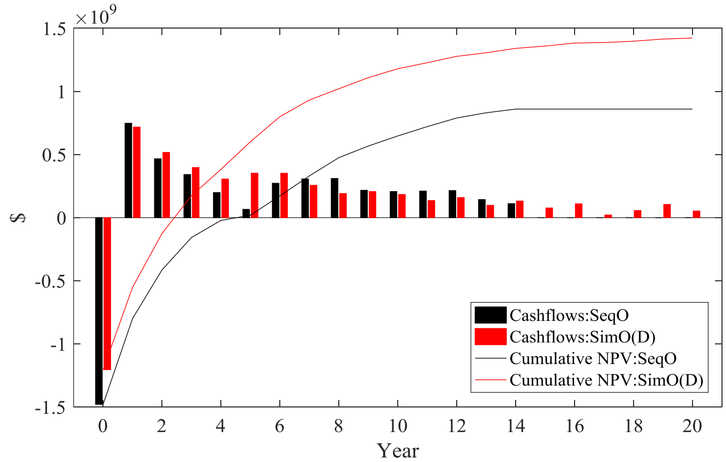


Figure 7: The cashflows and the cumulative NPVs of the MVC plans generated by SeqO and BDSimO(D).

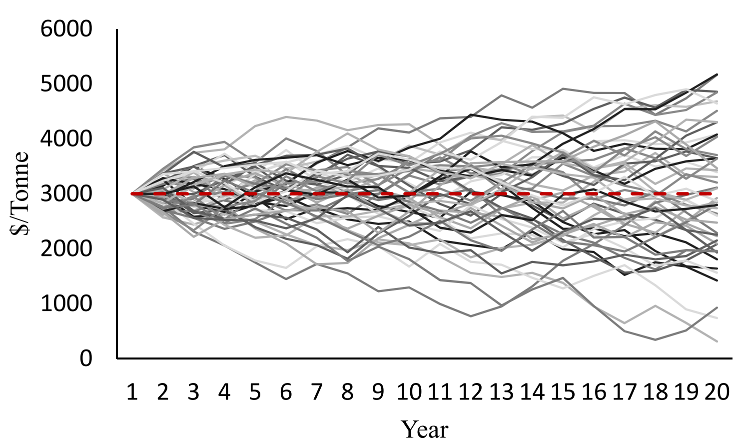


Figure 8: 50 simulations of copper price fluctuation over 20 years.

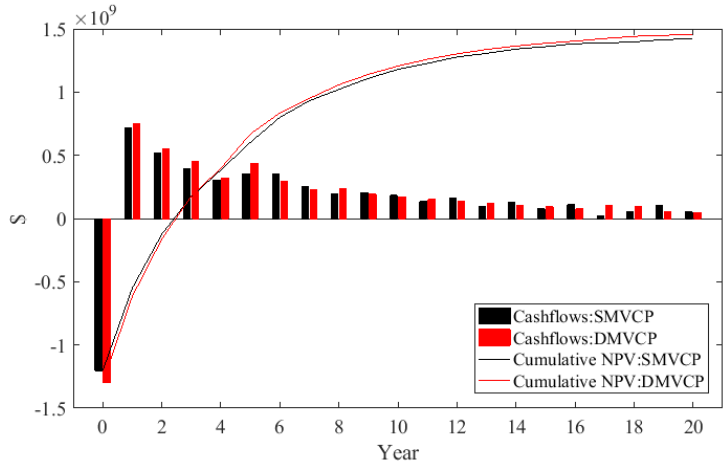


Figure 9: The cashflows and the cumulative NPVs of DMVCP and SMVCP.

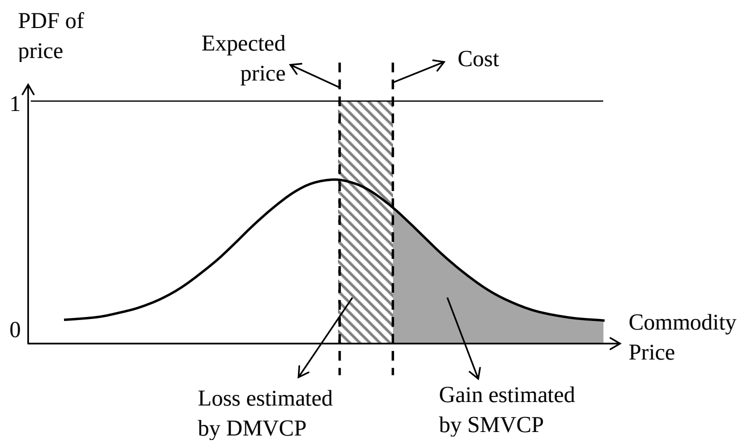


Figure 10: How value of the low-grade material is underestimated in deterministic optimization.

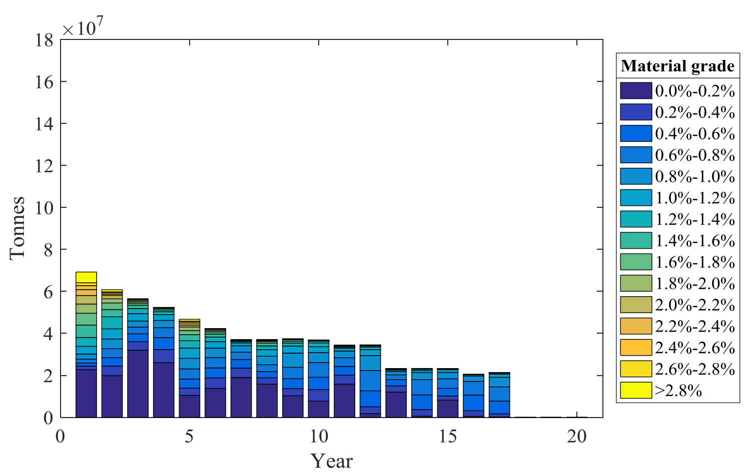
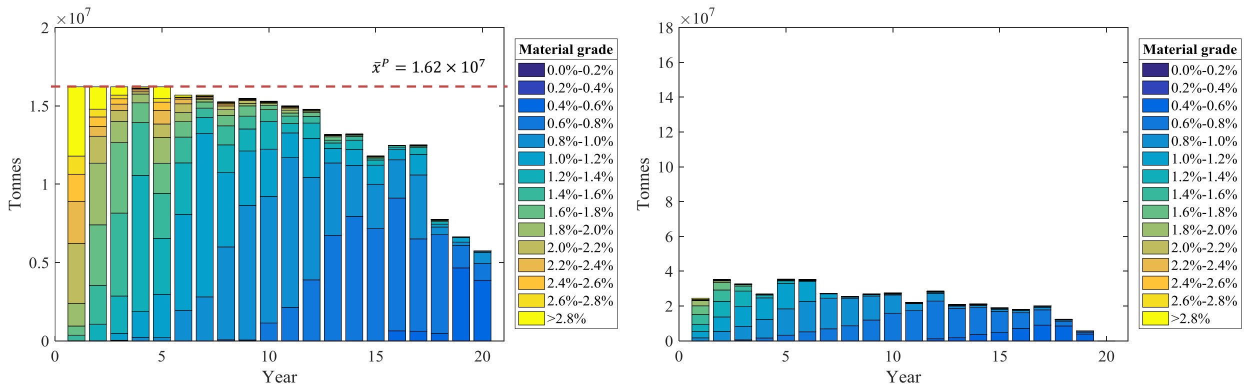


Figure 11: The materials mined in SMVCP.



(a) Processing amounts

(b) Stockpiling amounts

Figure 12: The average downstream stockpiling and processing amounts expected in each year when the market uncertainty is considered.

Symbol	Description*
General	
$T \in \mathbb{Z}^+$	The planning horizon.
$S \in \mathbb{Z}^+$	The number of orebody scenarios representing the geological uncertainty.
$I \in \mathbb{Z}^+$	The number of blocks considered for scheduling.
$J \in \mathbb{Z}^+$	The number of material types.
Indices	
$t \in \{1, \dots, T\}$	The index of a planning period.
$s \in \{1, \dots, S\}$	The index of a scenario in the stochastic model.
$i \in \{1, \dots, I\}$	The index of a block.
Parameters	
$\gamma \in \mathbb{R}^+$	The discount rate used for computing NPV.
$q_{ij} \in \mathbb{R}^+$	The parameter indicating the amount (in tonnes) of type j material in block i.
$c_i^M \in \mathbb{R}^+$	The cost of mining a ton of block i. The mining cost varies between blocks because it is determined by a block's characteristics such as hardness, locations, and so on.
$p_{ts} \in \mathbb{R}^+$	The price of a ton of commodity in period t.
$g_j \in (0, 1)$	The expected grade, i.e., the percentage of the valuable element contained, of type j material. In our example MVC model, only a single valuable element is considered.
$c^H \in \mathbb{R}^+$	The holding cost (per ton) for keeping the material in stockpiles.
$c^R \in \mathbb{R}^+$	The cost (per ton) for rehandling the material sent from stockpiles to the processing plant.
$\overline{y} \in \mathbb{R}^+$	Mining capacity, i.e., the maximum number of blocks that can be extracted in a period.
$\xi \in \mathbb{R}^+$	The smoothing parameter that constrains the fluctuation of the production rate at each planning period.
$\mathbb{P}_i \subset \{1, \dots, I\}$	The set of the direct predecessors of block i.
Variables	
$y_{it} \in \{0, 1\}$	Binary variable which equals to 1 if and only if block i is mined in period t.
$x_{jts}^{MP} \in \mathbb{R}^+$	The amount of type j material sent from mine to processing plant in period t.
$x_{jts}^{MH} \in \mathbb{R}^+$	The amount of type j material sent from mine to stockpiles in period t.
$x_{jts}^{HP} \in \mathbb{R}^+$	The amount of type j material sent from stockpiles to processing plant in period t.
$x_{jts}^{H} \in \mathbb{R}^+$	The stockpile level of type j material in period t.
$x^P \in \mathbb{R}^+$	The processing capacity of processing plant in a single period.

*The descriptions for subscript s, representing scenario-dependant, are omitted.

Table 1: List of notation

	Iteration number	Time used (s)	LB (Billion $)	HUB (Billion $)	Gap ([HUB-LB]/LB)
BDSimO(S)	1	296	$0.8599B	$4.1837B	386.56%
	2	702	$1.3532B	$1.4645B	8.36%
	3	1168	$1.3533B	$1.3533B	0.04%
	4	1352	$1.3533B	$1.3533B	0.00%
	5	1536	$1.3533B	$1.3533B	0.00%
	6	1717	$1.3533B	$1.3533B	0.00%
BDSimO(D)	1	294	$0.8599B	$4.1837B	386.56%
	2	889	$1.3138B	$1.4653B	11.53%
	3	1281	$1.4042B	$1.4123B	0.58%
	4	1869	$1.4042B	$1.3811B	-1.64%
	5	2250	$1.4224B	$1.4278B	0.38%
	6	2684	$1.4224B	$1.3462B	-5.36%
	7	3158	$1.4224B	$1.4029B	-1.37%
	8	3642	$1.4224B	$1.4084B	-0.99%

Table 2: Comparing BDSimO(S) and BDSimO(D).

Iteration number	Time used (s)	LB (Billion $)	HUB (Billion $)	Gap ([HUB-LB]/LB)
1	304.8916	$0.9306B	$4.1837B	349.58%
2	774.96	$1.4082B	$1.5322B	8.80%
3	1143.693	$1.4560B	$1.4619B	0.41%
4	1529.466	$1.4560B	$1.4472B	-0.60%
5	1966.712	$1.4560B	$1.3782B	-5.34%
6	2520.673	$1.4560B	$1.4066B	-3.39%

Table 3: The iteration process of BDSimO(D) when the market uncertainty is considered.