Optimization and Engineering
https://doi.org/10.1007/s11081-021-09629-9

RESEARCH ARTICLE

A matheuristic approach for optimizing mineral value chains under uncertainty

Amina Lamghari[1,2] · **Roussos Dimitrakopoulos**[1] · **Renaud Senécal**[1]

Received: 28 July 2020 / Revised: 5 April 2021 / Accepted: 5 April 2021

Abstract
Mineral value chains, also known as mining complexes, involve mining, processing, stockpiling, waste management and transportation activities. Their optimization is typically partitioned into separate stages, considered sequentially. An integrated stochastic optimization of these stages has been shown to increase the net present value of the related mining projects and operations, reduce risk in meeting production targets, and lead to more robust and coordinated schedules. However, it entails solving a larger and more complex stochastic optimization problem than separately optimizing individual components of a mineral value chain does. To tackle this complex optimization problem, a new matheuristic that integrates components from exact algorithms (relaxation and decomposition), machine learning techniques (reinforcement learning and artificial neural networks), and heuristics (local improvement and randomized search) is proposed. A general mathematical formulation that serves as the basis for the proposed methodology is also introduced, and results of computational experiments are presented.

Keywords Mineral value chains · Mining complexes · Stochastic simultaneous optimization · Large-scale optimization · Matheuristics · Decomposition

1 Introduction

Production scheduling (PS) and downstream optimization (DO) are two classical problems in mine planning. Production scheduling concerns strategic aspects, such as designing a mining sequence over the life-of-the-mine, while downstream

✉ Amina Lamghari
amina.lamghari@uqtr.ca

[1] COSMO - Stochastic Mine Planning Laboratory, Department of Mining and Materials Engineering, McGill University, Montreal, QC, Canada

[2] École de Gestion, Département de Management, Université du Québec À Trois-Rivières, Trois-Rivières, QC, Canada

Published online: 27 April 2021

optimization deals with the material flow aspect and assumes that strategic decisions have already been made. These two problems have been intensively studied in the mining literature. However, although the two problems are closely related, most studies focus on only one of them, leaving behind the benefits of coordinating and integrating mine planning decisions. The joint optimization of PS and DO while explicitly accounting for uncertainty (geological and/or financial), henceforth referred to as the simultaneous stochastic optimization of mining complexes (SSOMC), has received little attention despite its practical relevance, one main reason being computational complexity. Recently, there has been some work done, but the literature is still sparse. This paper aims to fill this gap.

As for what has been done, Goodfellow and Dimitrakopoulos (2016) introduced a mathematical model that specifically addresses the optimization of mining complexes under geological uncertainty. The authors employed a multi-neighborhood simulated annealing algorithm to solve the problem. The algorithm uses three neighborhoods. The first neighborhood changes the period in which a block is extracted, the second changes the destination of a cluster of blocks, and the third one modifies the amount of material sent from one processor to another. The authors compared the proposed algorithm to two other methods that combine multi-neighborhood simulated annealing with particle swarm and differential evolution, respectively. The latter proved to be more efficient, but also more computationally expensive. Multi-neighborhood simulated annealing was further studied in Montiel and Dimitrakopoulos (2015), Montiel et al. (2016), and Montiel and Dimitrakopoulos (2018). In the first paper, the authors simultaneously optimize mining, processing, and transportation decisions. The initial solution is improved by shifting a single block to another period, changing the operating mode in a processor, and changing the transportation arrangement in a processor. In the other two papers, the model and the algorithms are designed to handle mining complexes with multiple mines. Del Castillo and Dimitrakopoulos (2019) investigated a similar problem but in which capital expenditure decisions are included. There are some studies that considered exact methods. Zhang et al. (2019) studied mining complexes under market uncertainty. They combined classical Benders decomposition with aggregation techniques to reduce the size of the problem so that the resulting model is of tractable size and can be solved by the proposed method.

Nearly all papers on SSOMC, except the one above by Zhang et al. (2019), employ metaheuristics, mainly multi-neighborhood simulated annealing, to meet the challenges of scale, complexity, and uncertainty in optimizing mining complexes. In this paper, a different approach, namely a new matheuristic, is introduced. The proposed methodology capitalizes on the synergies between artificial intelligence and optimization techniques and integrates components from exact algorithms (relaxation and decomposition), machine learning techniques (reinforcement learning and artificial neural networks), and heuristics (local improvement and randomized search).

The remainder of the paper is divided into four sections. Section 2 gives a formal mathematical description of the problem studied. The proposed solution procedure

is described in Sect. 3. Numerical results are reported in Sect. 4. Section 5 provides conclusions and directions for future research.

2 Problem description and formulation

2.1 Problem description

A typical mining complex (mineral value chain) consists of multiple mines that provide material to multiple downstream facilities where the supply is blended and transformed into final sellable products. A downstream facility could be a stockpile, a crusher, a mill, a concentrator, a leach pad, a refinery, a waste dump, etc. and the output from one facility may be used as an input for another facility. In this general context, three kinds of decisions must be made in each period of the planning horizon: whether or not to mine a block (extraction decisions), where to send extracted blocks (first-stage destination decisions), and how to route the flow from the facilities fed by the mines to the facilities where final sellable products are produced (routing decisions). The constraints to be taken into account can be grouped into four main classes: precedence constraints, resource constraints, blending constraints, and flow conservation constraints. The information about blocks' mineral properties required for decision making is not known with certainty a priori, but only approximately in the form of equiprobable geological realizations (scenarios) of the orebodies. Such realizations are used to derive a robust solution that is well-hedged against uncertainty and where the risk of not meeting production targets is minimized.

In this paper, while extraction decisions are made at the block level, first-stage destination decisions are made at an aggregate level based on a clustering method similar to that proposed in Goodfellow and Dimitrakopoulos (2016). More specifically, in a pre-processing step, blocks' mineral properties are used to measure blocks' similarities and partition them into $\mathcal{G}$ subsets, henceforth called *groups*. We then refer to block i being in group g under geological scenario s when its mineral properties under that scenario fall into that group. Grouping blocks to define first-stage destinations is interesting from a practical point of view as it provides the mine planners with guidelines for what cut-off grade they should operate with for each year. It is also useful computationally, as it significantly reduces the size of the optimization problem and consequently the computational burden of solving it but at the cost of a loss in selectivity. The selectivity drawback can be overcome by setting $\mathcal{G}$ to high values close to the number of blocks.

A mathematical formulation of the optimization problem described above is presented in the next section.

2.2 Problem formulation

The following notation is used to formulate the problem.

Sets

- $\mathcal{T}$: Set of time periods, indexed by t.
- $\mathcal{B}$: Set of blocks, considering all mines in the mining complex (mineral value chain), indexed by b.
- $\mathcal{P}_b \subset \mathcal{B}$: Set of immediate predecessors of block b, indexed by b'.
- $\mathcal{S}$: Set of scenarios modelling geological uncertainty, indexed by s. Scenarios are multiple equiprobable geological realizations of the orebody, describing the uncertain geology and accurately reproducing the spatial statistics of the drillhole data.
- $\mathcal{G}$: Set of groups, indexed by g. Recall that the groups are defined based on the mineral properties of all blocks in the different mines, considering all geological scenarios.
- $\mathcal{A}$: Set of attributes or specificities to be tracked across the mineral value chain, indexed by a.
- $\mathcal{M} \subset \mathcal{A}$: Set of final sellable products (minerals), indexed by m.
- $\mathcal{R}_a \subset \mathcal{A}$: Set of attributes whose proportion relatively to the proportion of attribute a has to be controlled, indexed by a'.

To characterize the downstream part of the mining complex that goes from the first-stage facilities fed from the mines where blocks are first mixed, to intermediate facilities where material is further mixed and processed, to last-stage facilities where final sellable products are recovered, an acyclic direct graph is used. Nodes represent any downstream facility, and arcs between two nodes define an existing flow. The following notation is used to define the graph.

- $\mathcal{I}$: Set of nodes associated with first-stage facilities, indexed by i.
- $\mathcal{L}$: Set of nodes associated with last-stage facilities, indexed by l.
- $\mathcal{N} = \mathcal{I} \cup \mathcal{J} \cup \mathcal{L}$: Set of all nodes, indexed by n. Therefore, a node $j \in \mathcal{J}$ is associated with neither a first-stage facility nor a last-stage facility. It is associated with an intermediate facility.
- A: Set of arcs. An arc (i,j) exists only if facility i can potentially send material to facility j. Without loss of generality, we assume that $A \subseteq (\mathcal{I} \times \mathcal{J}) \cup (\mathcal{J} \times \mathcal{J}) \cup (\mathcal{J} \times \mathcal{L}) \cup (\mathcal{I} \times \mathcal{L})$

Parameters

- w_b: Weight of block b(tonnage).
- α^s_{ab}: Grade of attribute a in block b under scenario s. The grade is defined to be the proportion of attribute to rock.
- $\beta^s_{bg} = \begin{cases} 1 & \text{if block } b \text{ belongs to group g under scenario } s \\ 0 & \text{otherwise} \end{cases}$

Recall that groups $g \in \mathcal{G}$ are disjoint and are defined based on the blocks' mineral properties. Therefore, for a given scenario, a block cannot appear in different groups; that is, $\sum_{g \in \mathcal{G}} \beta^s_{bg} = 1 \forall b \in \mathcal{B},\ s \in S$. However, if we consider all scenarios, a block might appear in multiple groups; that is,

$$\sum_{s\in S}\sum_{g\in\mathcal{G}}\beta_{bg}^{s}\geq 1\quad\forall b\in\mathcal{B}.$$

- $\epsilon_{gi}=\begin{cases}1 & \text{if group } g \text{ is eligible for node } i\in I\\ 0 & \text{otherwise.}\end{cases}$

Recall that each $i\in\mathcal{I}$ is associated with one first-stage facility. An available (extracted) block cannot be sent to just any first-stage facility. It can go only to a predetermined subset of facilities depending on the block's material type and also on the mine from which the block has been extracted. The parameter ϵ_{gi} is used to identify such subset.

- $\overline{W^t}$: Maximum amount of material that can be mined in period t.
- $\overline{C_n^t}$: Maximum amount of material that can be processed at node $n\in\mathcal{N}=\mathcal{I}\cup\mathcal{J}\cup\mathcal{L}$ in period t (capacity of node n).
- $\overline{P_{al}^t}$: Upper limit (bound requirement) on the proportion (level) of attribute a at node $l\in\mathcal{L}$ in period t.
- $\overline{R_{aa'l}^t}$: Upper limit on the proportion of attribute a relative to attribute a' at node $l\in\mathcal{L}$ in period t.
- r_{ml}^t: Recovery of final sellable product $m\in\mathcal{M}$ at node $l\in\mathcal{L}$ during period $t\left(r_{ml}^t\in[0,1]\right)$. Recall that the set of final sellable products $\mathcal{M}$ is a subset of the set of attributes $\mathcal{A}$. The recovery represents the amount of output of attribute m obtained for each ton of input of m.
- c_b^t: Per-unit cost of mining block b in period t.
- c_{ij}^t: Per-unit cost of processing material at node j and/or transporting material on arc $(i,j)\in A$.
- σ_n^{ts+}: Unit surplus cost incurred if the amount of material received at node $n\in\mathcal{N}=\mathcal{I}\cup\mathcal{J}\cup\mathcal{L}$ during period t exceeds the capacity $\overline{C_n^t}$ of that node during that period.
- c_{al}^{t+}: Unit surplus cost incurred if the proportion of attribute a in node $l\in\mathcal{L}$ exceeds $\overline{P_{al}^t}$ during period t.
- $c_{aa'l}^{t+}$: Unit surplus cost incurred if the proportion of attribute a relative to that of a' in node $l\in\mathcal{L}$ exceeds $\overline{P_{aa'l}^t}$ during period t.
- π_m^t: Per-unit profit from selling final sellable product (mineral) m in period t, defined as being the revenue minus the selling cost.

Decision variables

- For each block $b\in\mathcal{B}$ and each period $t\in\mathcal{T}$, we define the following binary variables:

$$x_b^t=\begin{cases}1 & \text{if block } b \text{ is mined in period } t\\ 0 & \text{otherwise.}\end{cases}$$

- For each group $g\in\mathcal{G}$, each node $i\in\mathcal{I}$ (associated with a first-stage facility) and each period $t\in\mathcal{T}$, we define the following binary variables:

$$\kappa_{gi}^{t} = \begin{cases} 1 \text{ if } blocks \text{ in group } g \text{ are sent to destination } i \text{ in period } t \\ 0 \text{ otherwise.} \end{cases}$$

As discussed in Sect. 2.1, variables κ_{gi}^{t} control the cut-off grade applied at i during period t.

Other decision variables in the formulation are the following:

- y_{ij}^{ts}: Flow of material on arc (i, j) in period t under scenario s.
- Δ_{n}^{ts+}: Surplus in the amount of material received at node $n \in \mathcal{N}$ in period t under scenario s. Recall that the set of nodes $\mathcal{N}$ comprises nodes associated with first-stage facilities $i \in \mathcal{I}$, with intermediate facilities node $j \in \mathcal{J}$, and with last-stage facilities $l \in \mathcal{L}$.
- p_{aj}^{ts}: Proportion value (grade) of attribute $a \in \mathcal{A}$ in material leaving node $j \in \mathcal{I} \bigcup \mathcal{J}$ at period t under scenario s.
- δ_{al}^{ts+}: Surplus in quantity of attribute a in node $l \in \mathcal{L}$ in period t under scenario s (before considering the arriving material).
- $d_{aa'l}^{ts+}$: Surplus in the proportion of quantity of attribute $a \in \mathcal{A}$ compared to the quantity of attribute $a' \in \mathcal{R}_a$ in node $l \in \mathcal{L}$ in period t under scenario s (before considering the arriving material).

To simplify the presentation, we introduce the following three variables, although not necessary:

- μ_{i}^{ts}: Quantity of material (rock) supplied from the mines to node $i \in \mathcal{I}$ at period t under scenario s.
- ϑ_{ai}^{ts}: Proportion value (grade) of attribute $a \in \mathcal{A}$ in material supplied from the mines to node $i \in \mathcal{I}$ during period t under scenario s.
- o_{m}^{ts}: Quantity produced of sellable product (mineral) $m \in \mathcal{M}$ in period t under scenario s.

Denote by ζ the random data vector (uncertain parameters) and by $\zeta(s)$ one of its particular realizations. Assuming that all the scenarios are equiprobable, the two-stage stochastic formulation of the problem can be written as follows:

$$\max f(x, \kappa) = -\sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}} c_b^t x_b^t + \frac{1}{S} \sum_{s \in S} Q(x, \kappa, \zeta(s)) \tag{1}$$

Subject to

$$\sum_{t \in \mathcal{T}} x_b^t \leq 1 \quad \forall b \in \mathcal{B} \tag{2}$$

$$x_b^t \leq \sum_{\tau=1}^{t} x_{b'}^{\tau} \quad \forall b \in \mathcal{B}, b' \in \mathcal{P}_b, t \in \mathcal{T} \tag{3}$$

$$\sum_{b\in\mathcal{B}} w_b x_b^t \le \overline{W^t} \quad \forall t \in \mathcal{T} \tag{4}$$

$$\sum_{i\in\mathcal{I}} \kappa_{gi}^t = 1 \quad \forall g \in \mathcal{G},\ t \in \mathcal{T} \tag{5}$$

$$\kappa_{gi}^t \le \epsilon_{gi} \quad \forall g \in \mathcal{G},\ i \in \mathcal{I},\ t \in \mathcal{T} \tag{6}$$

$$x_b^t \in \{0,1\} \quad \forall b \in \mathcal{B},\ t \in \mathcal{T} \tag{7}$$

$$\kappa_{gi}^t \in \{0,1\} \quad \forall g \in \mathcal{G},\ i \in \mathcal{I},\ t \in \mathcal{T} \tag{8}$$

where $Q(x,\kappa,\zeta(s))$ is the optimal value of the following problem (second-stage problem):

$$Q(x,\kappa,\zeta(s)) = \max \sum_{t\in\mathcal{T}} \left\{ -\sum_{(i,j)\in A} c_{ij}^t y_{ij}^{ts} - \sum_{n\in N} \sigma_n^{t+}\Delta_n^{ts+} - \sum_{a\in\mathcal{A}}\sum_{l\in\mathcal{L}} c_{al}^{t+}\delta_{al}^{ts+} - \sum_{a\in\mathcal{A}}\sum_{a'\in\mathcal{R}_a}\sum_{l\in\mathcal{L}} c_{aa'l}^{t+} d_{aa'l}^{ts+} + \sum_{m\in\mathcal{M}} \pi_m^t o_m^{ts} \right\} \tag{9}$$

Subject to

$$\sum_{g\in\mathcal{G}}\sum_{b\in\mathcal{B}} \beta_{bg}^s w_b x_b^t \kappa_{gi}^t = \mu_i^{ts} \quad \forall i \in \mathcal{I},\ t \in \mathcal{T} \tag{10}$$

$$\mu_i^{ts} = \sum_{j\in\mathcal{J}\cup\mathcal{L}} y_{ij}^{ts} \quad \forall i \in \mathcal{I},\ t \in \mathcal{T} \tag{11}$$

$$\sum_{i\in\mathcal{I}\cup\mathcal{J}} y_{ij}^{ts} = \sum_{l\in\mathcal{J}\cup\mathcal{L}} y_{jl}^{ts} \quad \forall j \in \mathcal{J},\ t \in \mathcal{T} \tag{12}$$

$$\mu_i^{ts} - \Delta_i^{ts+} \le \overline{C_i^t} \quad \forall i \in \mathcal{I},\ s \in S \tag{13}$$

$$\sum_{i\in\mathcal{I}\cup\mathcal{J}} y_{ij}^{ts} \le \overline{C_j^t} \quad \forall j \in \mathcal{J}\cup\mathcal{L}, t \in \mathcal{T} \tag{14}$$

$$\sum_{g\in\mathcal{G}}\sum_{b\in\mathcal{B}} \beta_{bg}^s \alpha_{ab}^s w_b x_b^t \kappa_{gi}^t = \vartheta_{ai}^{ts}\mu_i^{ts} \quad \forall a \in \mathcal{A}, i \in \mathcal{I},\ t \in \mathcal{T} \tag{15}$$

$$\sum_{i\in\mathcal{I}} \vartheta_{ai}^{ts} y_{ij}^{ts} + \sum_{j'\in\mathcal{J}} p_{aj'}^{ts} y_{j'j}^{ts} = p_{aj}^{ts} \sum_{i\in\mathcal{I}\cup\mathcal{J}} y_{ij}^{ts} \quad \forall a \in \mathcal{A},\ j \in \mathcal{J}\cup\mathcal{L}, t \in \mathcal{T} \tag{16}$$

$$p_{al}^{ts} - \delta_{al}^{ts+} \le \overline{P_{al}^t} \quad \forall a \in \mathcal{A},\ l \in \mathcal{L}, t \in \mathcal{T} \tag{17}$$

$$p^{ts}_{al} - d^{ts+}_{aa'l} \leq \overline{R^{t}_{aa'l}} p^{ts}_{a'l} \quad \forall a \in \mathcal{A},\ a' \in \mathcal{R}_a, l \in \mathcal{L}, t \in \mathcal{T} \tag{18}$$

$$o^{ts}_{m} = \sum_{l \in \mathcal{L}} p^{ts}_{ml} \sum_{i \in \mathcal{I} \cup \mathcal{J}} y^{ts}_{il} \quad \forall m \in \mathcal{M}, l \in \mathcal{L},\ t \in \mathcal{T} \tag{19}$$

$$y, p, \mu, \vartheta, o, \Delta^{+}, \delta^{+}, d^{+} \geq 0 \tag{20}$$

The objective function (1) minimizes the first-stage mining costs and maximizes the expected second-stage profit. The objective function of the second-stage (9) consists of five parts: the processing and transportation costs (first term), the penalty costs (second, third, and fourth terms), and the revenues from the minerals (final products) produced (fifth term). It is assumed that all minerals produced in period t can be sold in the same period. Penalties are incurred for failing to meet the requirements of the downstream facilities. Indeed, given that the blocks are sent as groups to first-stage facilities, it might be impossible to respect the facilities' capacity for certain realizations. Therefore, an additional cost (second term) is included to penalize surplus. Similarly, given that it might be impossible to meet upper bounds on the grades and proportions for certain realizations of the uncertain parameters, additional cost terms are included to penalize surplus in the attributes that need to be controlled (third and fourth term).

Constraints (2–8) define the feasible set of the first-stage variables. More specifically, constraints (2) ensure that a block is mined at most once while constraints (3) prevent a block from being mined before its predecessors. Constraints (4) impose maximal mining level at each period. Constraints (5) allow a group of blocks to be sent to one first-stage facility only if it is *eligible* for this facility [constraints (6)].

Constraints (10) link the first-stage variables to the second-stage variables. More specifically, they define the total tonnage available at each first-stage facility under each scenario. Constraints (11) ensure that the supply from the mines is transported to a downstream facility. Constraints (12) are the flow balance at each intermediate facility. The total quantity in a facility at each period can be at most the facility capacity as specified by constraints (13) and (14). Surplus is allowed but at the expense of an additional cost.

Constraints (15) define the grade of each attribute at each first-stage facility; that is, in the material supplied from the mines. Constraints (16) are similar and apply to the intermediate and last-stage facilities. Recall that the grade is the proportion of the attribute to rock. The grade p^{ts}_{aj} of attribute a in node j under scenario s at the end of period t is thus a variable that depends on the amount of this attribute and the amount of rock sent to this node from the nodes of the previous stage. Also, recall that for each node associated with a first-stage or intermediate facility, the amount of input attribute and the output attribute are equal. Given these considerations, p^{ts}_{aj} of the material leaving node j can be calculated as follows:

$$p_{aj}^{ts} = \frac{\sum_{i \in \mathcal{I}} \vartheta_{ai}^{ts} y_{ij}^{ts} + \sum_{j' \in \mathcal{J}} p_{aj'}^{ts} y_{j'j}^{ts}}{\sum_{i \in \mathcal{I} \bigcup \mathcal{J}} y_{ij}^{ts}} \tag{21}$$

These equations simplify to the bilinear constraints (16).

Constraints (17) ensure that the proportion of each attribute in the combined material in each last-stage facility lies below a specific level, and if not, a penalty cost is incurred. Note that an attribute can be a pollutant, and thus these constraints handle sustainability constraints at waste dumps, tailings and slags, for example. Constraints (18) are similar and deal with the proportion of an attribute a relative to another attribute $a' \in \mathcal{R}_a$. Constraints (19) define the amount of each mineral produced at each period under each scenario at the final nodes. Finally, constraints (20) are the non-negativity constraints. Indices are omitted. Although in this paper we assume that the arcs have unlimited capacities, limits on the flows can be easily accommodated without any added conceptual difficulties.

In the next section, the algorithmic strategy for addressing the intrinsic difficulties of problem (1)–(20); i.e., non-linear constraints, complex objective function, and large size, is detailed.

3 Matheuristic

3.1 General structure of the algorithm

As mentioned in Sect. 1, to tackle the SSOMC, a matheuristic that exploits mathematical programming techniques and machine learning techniques in a heuristic framework is used. Within this algorithmic framework, the problem is separated into two sub-problems. The first one, referred to as the upstream sub-problem, determines a mining sequence; that is, which blocks to extract at each period of the life-of-the-mine. Given a mining sequence, the second sub-problem, referred to as the downstream sub-problem, determines where to send the extracted blocks and the most profitable way to mix them so as to meet the different processing facilities' capacities and blending requirements, as well as production targets. The two sub-problems are iteratively solved until a stopping criterion is met.

Solving the upstream sub-problem involves modifying the mining sequence to explore different parts of the solution space. This is done using the hyper-heuristic proposed in Lamghari and Dimitrakopoulos (2020) that uses a set of 27 simple perturbative low-level heuristics and a score-based learning mechanism along with a tabu list to select the low-level heuristic to apply at each step of the solution process. To evaluate the mining sequences generated by the selected low-level heuristic, the downstream sub-problem, whose formulation involves binary variables and bilinear terms, must be solved. This is done by using an exact algorithm inspired by Benders decomposition (Benders 1962). The original downstream sub-problem is reformulated into a master problem and a series of blending sub-problems. The master is a relaxation of the original problem in which McCormick envelopes are used to relax the bilinear terms and valid inequalities are used to strengthen the resulting linear

formulation. At each iteration of this decomposition algorithm, the master problem is solved to obtain an upper bound on the optimal objective function value; then the blending sub-problems are solved by holding the master binary variables fixed to compute a lower bound and generate an optimality cut. This cut is added to the master, and the process is repeated until a threshold optimality gap is reached. As for the upstream sub-problem, machine learning techniques are also exploited when solving the downstream sub-problem. More specifically, in order to speed up the computational times, an artificial neural network surrogate is used instead of the decomposition algorithm to approximate the objective function values and identify the best neighbor solutions generated by the upstream low-level heuristic. The decomposition algorithm is then employed only for these solutions, and its output is used to update the scores of the low-level heuristics and the training set of the neural network for the next iteration.

A more detailed description of the procedures proposed to solve the downstream sub-problem is provided in the following section. Details about the algorithm for solving the upstream sub-problem can be found in Lamghari and Dimitrakopoulos (2020).

3.2 Solving the downstream sub-problem

Let $\bar{x} = (\bar{x}_b^t)$ be a mining sequence generated by the hyper-heuristic. Denote by:

- $w_g^{ts} = \sum_{b \in \mathcal{B}} \beta_{bg}^s w_b \bar{x}_b^t$ the total tonnage of material that belongs to group g under scenario s, available at period t
- $\lambda_{ag}^{ts} = \frac{\sum_{b \in \mathcal{B}} \beta_{bg}^s w_b \bar{x}_b^t}{w_g^{ts}}$, the grade of attribute a in the material that belongs to group g under scenario s, available at period t.

Evaluating the objective function $f(\bar{x})$ involves identifying the flow of the extracted material through the downstream part of the mining complex; i.e., solving the downstream sub-problem. Assuming that there are no stockpiles that link the periods to each other, the sub-problems associated with the periods can be solved independently. Therefore, for notational simplicity, we drop the subscript designating the period in the rest of this section. All variables and parameters should be understood as related to a particular period t.

The sub-problem associated with a period t, which we will refer to as DP, can informally be defined as follows: Given a set of groups (raw material with different properties) characterized by scenario-dependent tonnages and attribute levels (w_g^s and λ_{ag}^s, respectively) that have to be first mixed in first-stage facilities ($i \in \mathcal{I}$), then in intermediate facilities ($j \in \mathcal{J}$), and finally sent forth from the intermediate facilities to be mixed again at last-stage facilities ($l \in \mathcal{L}$) to recover final sellable products, what is the most profitable way to mix these groups so as to meet to the best extent possible the different facilities' capacities as well as the attribute requirements at the last-stage facilities?

The DP can be viewed as a stochastic variant of the well-known pooling problem, which is a generalization of the blending problem (Gupte et al. 2017).

Different formulations have been proposed in the literature for the pooling problem, one of which is extended below to account for the specificities and handle the features of the problem addressed in this paper. It is based on that proposed by Haverly (1978) and uses the so-called *quality* variables that control the concentration of each attribute in the material leaving each facility, which lead to a non-linear formulation. In what follows, we first provide the mathematical model and then its linear relaxation. In addition to the pooling problem requirements and restrictions, binary variables are required to model the flow of materials in first-stage facilities and scenario-dependent variables are required to account for the stochastic nature (uncertainty) of supply.

3.2.1 Mathematical formulation

We seek to maximize the expected profit; that is, the expected revenue from the final sellable products minus the total expected cost, which is the sum of the expected transportation and processing costs at the first-stage, intermediate and last-stage facilities and of the expected penalty costs incurred whenever the capacity, blending, and ratio constraints are violated. Using the same notation as in the previous section (Sect. 2) omitting the t subscript, the objective function can be expressed as follows:

$$\max \frac{1}{S} \sum_{s \in S} \left\{ \sum_{m \in \mathcal{M}} \pi_m o_m^s - \sum_{(i,j) \in A} c_{ij} y_{ij}^s - \sum_{j \in \mathcal{N}} \sigma_j^+ \Delta_j^{s+} - \sum_{a \in \mathcal{A}} \sum_{l \in \mathcal{L}} c_{al}^+ \delta_{al}^{s+} - \sum_{a \in \mathcal{A}} \sum_{a' \in \mathcal{R}_a} \sum_{l \in \mathcal{L}} c_{aa'l}^+ d_{aa'l}^{s+} \right\} \tag{22}$$

The constraints are as follows:

1. *First-stage facilities constraints*: This set of constraints ensures that each group $g \in \mathcal{G}$ is sent to exactly one single first-facility for which it is *eligible*:

$$\sum_{i \in \mathcal{I}} \kappa_{gi} = 1 \quad \forall g \in \mathcal{G} \tag{23}$$

$$\kappa_{gi} \leq \epsilon_{gi} \quad \forall g \in \mathcal{G},\ i \in \mathcal{I} \tag{24}$$

2. *Flow conservation constraints*: These constraints ensure that the amount entering and leaving each first stage facility $i \in \mathcal{I}$ and each intermediate facility $j \in \mathcal{J}$ under each scenario $s \in S$ are the same:

$$\sum_{g \in \mathcal{G}} w_g^s \kappa_{gi} = \sum_{j \in \mathcal{J} \cup \mathcal{L}} y_{ij}^s \quad \forall i \in \mathcal{I},\ s \in S \tag{25}$$

$$\sum_{i \in \mathcal{I} \cup \mathcal{J}} y_{ij}^s = \sum_{l \in \mathcal{J} \cup \mathcal{L}} y_{jl}^s \quad \forall j \in \mathcal{J}, s \in S \tag{26}$$

3. *Capacity constraints*: These constraints limit the quantity received at each facility $n \in \mathcal{N}$ under each scenario $s \in S$ to the facility's capacity; otherwise, an extra cost $\sigma_n^+ \Delta_n^{s+}$ is incurred:

$$\sum_{j \in \mathcal{J} \cup \mathcal{L}} y_{ij}^s - \Delta_i^{s+} \leq \overline{C_i} \quad \forall i \in \mathcal{I},\ s \in S \tag{27}$$

$$\sum_{l \in \mathcal{J} \cup \mathcal{L}} y_{jl}^s - \Delta_j^{s+} \leq \overline{C_j} \quad \forall j \in \mathcal{J}, s \in S \tag{28}$$

$$\sum_{i \in \mathcal{I} \cup \mathcal{J}} y_{il}^s - \Delta_l^{s+} \leq \overline{C_l} \quad \forall l \in \mathcal{L}, s \in S \tag{29}$$

4. *Grade definition at the first-stage and intermediate facilities*: Assuming that at least one group whose tonnage is non-null has been sent to first-stage facility $i \in \mathcal{I}$ (i.e., that $\sum_{g \in \mathcal{G}} w_g^s \kappa_{gi} \neq 0$) and that the mixing process follows a linear blending, the grade of attribute $a \in \mathcal{A}$ in the material leaving node i under scenario s is given by[1]:

$$p_{ai}^s = \frac{\sum_{g \in \mathcal{G}} \lambda_{ag}^s w_g^s \kappa_{gi}}{\sum_{g \in \mathcal{G}} w_g^s \kappa_{gi}}.$$

This expression can be equivalently rewritten in a bilinear form as:

$$\sum_{g \in \mathcal{G}} \lambda_{ag}^s w_g^s \kappa_{gi} = p_{ai}^s \sum_{g \in \mathcal{G}} w_g^s \kappa_{gi} \quad \forall a \in \mathcal{A},\ i \in \mathcal{I},\ s \in S \tag{30-a}$$

which can also be rewritten, given constraints (25), as follows:

$$\sum_{g \in \mathcal{G}} \lambda_{ag}^s w_g^s \kappa_{gi} = p_{ai}^s \sum_{j \in \mathcal{J} \cup \mathcal{L}} y_{ij}^s \quad \forall a \in \mathcal{A},\ i \in \mathcal{I},\ s \in S \tag{30-b}$$

Similarly, at the intermediate facilities $j \in \mathcal{J}$, we have:

$$\sum_{i \in \mathcal{I}} p_{ai}^s y_{ij}^s + \sum_{j' \in \mathcal{J}} p_{aj'}^s y_{j'j}^s = p_{aj}^s \sum_{i \in \mathcal{I} \cup \mathcal{J}} y_{ij}^s \quad \forall a \in \mathcal{A},\ j \in \mathcal{J}, s \in S \tag{31-a}$$

which can also be rewritten as follows, considering constraints (26):

$$\sum_{i \in \mathcal{I}} p_{ai}^s y_{ij}^s + \sum_{j' \in \mathcal{J}} p_{aj'}^s y_{j'j}^s = p_{aj}^s \sum_{l \in \mathcal{J} \cup \mathcal{L}} y_{jl}^s \quad \forall a \in \mathcal{A},\ j \in \mathcal{J}, s \in S \tag{31-b}$$

[1] Note that, in this paper, it is assumed that the recovery factor at the first-stage and intermediate facilities is equal to 1, while it is smaller than 1 in the last-stage facilities. Different values for the recovery factor can be easily accommodated without any added conceptual difficulties.

5. *Blending constraints*: These constraints model the mixing process requirements at each last-stage facility $l \in \mathcal{L}$. Constraints (32) limit the grade of attribute $a \in \mathcal{A}$ in the material received at l to the facility's upper bound, $\overline{P_{al}}$, making it impossible to have more than what would exceed this upper bound unless a unit cost of c^{+}_{al} is paid. Constraints (33) model the ratio requirements, ensuring that in the material processed at l, the ratio between attribute $a \in \mathcal{A}$ to related attribute $a' \in \mathcal{R}_a$ does not exceed the upper bound $\overline{R_{aa'l}}$; otherwise, a cost $c^{+}_{aa'l}$ has to be paid for each unit in surplus:

$$\sum_{i \in \mathcal{I}} p^s_{ai} y^s_{il} + \sum_{j \in \mathcal{J}} p^s_{aj} y^s_{jl} - \delta^{s+}_{al} \leq \overline{P_{al}} \sum_{i \in \mathcal{I} \cup \mathcal{J}} y^s_{il} \quad \forall a \in \mathcal{A}, l \in \mathcal{L}, s \in S \tag{32}$$

$$\sum_{i \in \mathcal{I}} p^s_{ai} y^s_{il} + \sum_{j \in \mathcal{J}} p^s_{aj} y^s_{jl} - d^{s+}_{aa'l} \leq \overline{R_{aa'l}} \left(\sum_{i \in \mathcal{I}} p^s_{a'i} y^s_{il} + \sum_{j \in \mathcal{J}} p^s_{a'j} y^s_{jl} \right) \quad \forall a \in \mathcal{A}, a' \in \mathcal{R}_a, l \in \mathcal{L}, s \in S \tag{33}$$

6. *Quantities produced*: The amount produced of each sellable final product $m \in \mathcal{M} \subset \mathcal{A}$ under each scenario $s \in S$ is given by the total amount recovered in the different last-stage facilities:

$$o^s_m = \sum_{l \in \mathcal{L}} r_{ml} \left(\sum_{i \in \mathcal{I}} p^s_{mi} y^s_{il} + \sum_{j \in \mathcal{J}} p^s_{mj} y^s_{jl} \right) \quad \forall m \in \mathcal{M}, s \in S \tag{34}$$

7. Integrality and non-negativity constraints:

$$\kappa_{gi} \in \{0, 1\} \text{ and } y, p, o, \Delta^{+}, \delta^{+}, d^{+} \geq 0. \tag{35}$$

3.2.2 Linear relaxation of the bilinear terms

The problem described in Sect. 3.2.1 is a stochastic variant of the pooling problem, which is a nonconvex, strongly NP-hard bilinear problem (Gupte et al. 2017). It can be relaxed by linearizing the bilinear terms in constraints (30)–(34). Recall that the bilinear constraints defining the grade at first-stage and intermediate facilities can be written in two different forms. For first-stage facilities i, constraints (30-a) will be used instead of constraints (30-b) as the former allow an exact linearization as explained next.[2] For intermediate facilities j, constraints (31-b) will be used instead of (31-a) in order to have all the non-linearities in the model in the form $p^s_{ai} y^s_{ij}$ (using constraints (31-a) would introduce bilinear terms of the form $p^s_{aj} y^s_{ij}$). Before describing the proposed linear relaxation techniques, some extra notation and definitions that will be used through this section to facilitate further discussions must first be introduced.

[2] Recall that the bilinear terms in (30-a) are the product of a binary and a continuous variable as opposed to the product of two continuous variables in (30-b).

For every facility $j \in \mathcal{I} \bigcup \mathcal{J}$, let $\mathcal{G}_j$ be the subset of groups that might end up being processed at j. This means that for intermediate facilities $j \in \mathcal{J}$,$\mathcal{G}_j = \{g \in \mathcal{G} :$ there exists at least a first-stage facility $i \in \mathcal{I}$ to which g is *eligible* and from which there exists a path in the graph to $j\}$, while for first-stage facilities $i \in \mathcal{I}$, $\mathcal{G}_j$ reduces to the subset of groups *eligible* to i ($\mathcal{G}_j = \{g \in \mathcal{G} : \epsilon_{gi} = 1\}$). Also, for every attribute $a \in \mathcal{A}$, every facility $j \in \mathcal{I} \bigcup \mathcal{J}$ and every scenario $s \in S$, let $\Lambda^s_{aj} = \min_{g \in \mathcal{G}_j} \lambda^s_{ag}$ be a lower bound on the grade of attribute a in the material leaving j under scenario s. Note that this lower bound is valid since the flow leaving j originates at the groups $g \in \mathcal{G}_j$. The upper bound is denoted by $\overline{\Lambda^s_{aj}} = \max_{g \in \mathcal{G}_j} \lambda^s_{ag}$. Finally, for each arc $(i,j) \in A$ and each scenario $s \in S$, define $\overline{Y^s_{ij}} = \sum_{g \in \mathcal{G}_i} w^s_g$ to be an upper bound on the flow along arc (i,j). [3] We assume the lower bound on flow on arc (i,j) to be zero (i.e., $\underline{Y^s_{ij}} = 0 \forall (i,j) \in A, s \in S$).

Using the notation above, constraints (30-a) can be linearized by substituting every bilinear term $p^s_{ai}\kappa_{gi}$ with a new variable u^s_{agi} and adding the following constraints:

$$u^s_{agi} \leq \overline{\Lambda^s_{ai}}\kappa_{gi} \tag{36-a}$$

$$u^s_{agi} \leq p^s_{ai} \tag{36-b}$$

$$u^s_{agi} \geq p^s_{ai} - (1 - \kappa_{gi})\overline{\Lambda^s_{ai}} \tag{36-c}$$

$$u^s_{agi} \geq 0 \tag{36-d}$$

This way, if κ_{gi} is zero, then inequalities (36-a) and (36-d) ensure that u^s_{agi} will be zero as well, and the other inequalities are redundant. On the other hand, if κ_{gi} is equal to 1, inequalities (36-b) and (36-c) imply that u^s_{agi} will be p^s_{ai}, exactly as required. Note that in this case, inequality (36-a) is redundant since it ensures that $u^s_{agi} = p^s_{ai}$ is less than $\overline{\Lambda^s_{ai}} = \max_{g \in \mathcal{G}_i} \lambda^s_{ag}$, which is true by definition.

The standard McCormick approximation (McCormick 1976) can be used to relax the bilinear terms in constraints (31-b) and (32)–(34), therefore obtaining a Mixed Integer Linear Programming Problem. This approach consists of introducing new variables and adding linear constraints that tie these variables close to the bilinear terms that they replace. More precisely, any bilinear term xy, where $x \in [\underline{x}, \overline{x}]$ and $y \in [\underline{y}, \overline{y}]$, is replaced by a new variable v. This new variable is linked to the variables x and y via the following set of constraints:

[3] Note that upper bound on the flow from i to j cannot be defined as $\overline{Y^s_{ij}} = min\{\sum_{g \in \mathcal{G}_i} w^s_g, \overline{C_i}, \overline{C_j}\}$ because the capacity constraints are modelled as soft constraints (cf. constraints (27)–(29)), and thus the flow on arc (i,j) might exceed the facilities' capacity.

$$v \geq \underline{x}y + \underline{y}x - \underline{x}\underline{y} \tag{37-a}$$

$$v \geq \overline{x}y + \overline{y}x - \overline{x}\overline{y} \tag{37-b}$$

$$v \leq \overline{x}y + \underline{y}x - \overline{x}\underline{y} \tag{37-c}$$

$$v \leq \underline{x}y + \overline{y}x - \underline{x}\overline{y} \tag{37-d}$$

It is worth noting that the tightness of the relaxation depends on the tightness of the bounds of the variables. It has been shown in Androulakis et al. (1995) that the maximum difference between the variable v and the bilinear term xy is $d_{max} = \frac{1}{4}(\overline{x} - \underline{x})(\overline{y} - \underline{y})$. This explains why constraints (30-a) were chosen over constraints (30-b) to model the grade definition constraints at first-stage facilities.

Without loss of generality, consider the bilinear terms $p^s_{ai} y^s_{ij}$ in constraints (31-b). Using the notation introduced at the beginning of this section, the McCormick inequalities reduce to the following:

$$v^s_{aij} \geq \underline{\Lambda^s_{ai}} y^s_{ij} \tag{38-a}$$

$$v^s_{aij} \geq \overline{\Lambda^s_{ai}} y^s_{ij} + \overline{Y^s_{ij}} p^s_{ai} - \overline{\Lambda^s_{ai}}\ \overline{Y^s_{ij}} \tag{38-b}$$

$$v^s_{aij} \leq \overline{\Lambda^s_{ai}} y^s_{ij} \tag{38-c}$$

$$v^s_{aij} \leq \underline{\Lambda^s_{ai}} y^s_{ij} + \overline{Y^s_{ij}} p^s_{ai} - \underline{\Lambda^s_{ai}}\,\overline{Y^s_{ij}} \tag{38-d}$$

where v^s_{aij} is defined as:

$$v^s_{aij} = p^s_{ai} y^s_{ij}. \tag{38-e}$$

Note that the variable v^s_{aij} can be interpreted as the total amount of attribute a in the flow along arc (i, j) under scenario s. With this definition, constraints (31-b), defining the grade at intermediate facilities $j \in \mathcal{J}$, simply state that the amounts of attribute a entering and leaving node (facility) j are equal.

3.2.3 Decomposition algorithm

As mentioned in Sect. 3.1, to solve the DP (the downstream sub-problem), an exact algorithm, inspired by Benders decomposition (Benders 1962), is proposed. Benders decomposition can be summarized as follows: The original problem is reformulated into a master and a number of sub-problems. At each iteration, the values of the master problem variables are first determined, and the sub-problems are solved by holding these variables fixed. If all the sub-problems are feasible and bounded, an

optimal cut is added to the master problem; otherwise, a feasibility cut is added. A lower bound can be computed from the sub-problems and an upper bound is obtained if the master problem is solved to optimality. The process continues until an optimal solution is found or until the optimality gap is smaller than a given threshold value. In the context of stochastic integer programming literature, this algorithmic scheme is also known as the L-shaped method (Birge and Louveaux 2011).

In general, there are two decomposition strategies: one that projects out all variables of the subproblems from the master, and one that does not project out any variables. In this paper, the second strategy is used. Namely, the master formulation includes all variables. It also includes linear approximations of the non-linear constraints (all bilinear terms are replaced with the corresponding McCormick envelopes or exact linearization, described in the previous section). The master is thus a mixed-integer linear programming relaxation of the original problem. During the course of the algorithm, additional constraints (Benders cuts) are also included in the formulation of the master problem to obtain a better approximation of the objective function value as discussed in detail below.

Define:

- θ: an approximation of the objective function value of the original problem DP (the non-linear stochastic mixed integer program).
- U: an upper bound on the value of the objective function value of the original problem DP described in Sect. 3.2.1. Since DP is bounded, U is finite.
- $\mathcal{R}$: the set of first-stage feasible solutions, indexed by r. Note that since first-stage decision variables (κ_{gi}) are binary, the set $\mathcal{R}$ is finite.

Moreover, for every r, let:

- θ_r: a constant equal to the expected second stage value of the r^{th} solution. To compute θ_r, the κ_{gi} are fixed as in the r^{th} solution and the so-obtained $|S|$ blending problems are solved to get the values of the other variables. Let $(y_{ij}^{s*}, o_m^{s*}, \Delta_j^{s+*}, \delta_{al}^{s+*}, d_{aa'l}^{s+*})$ be the optimal solution of the blending problem associated with scenario s. Then θ_r is calculated as follows:

$$\theta_r = \frac{1}{S}\sum_{s\in S}\left\{\sum_{m\in\mathcal{M}}\pi_m o_m^{s*} - \sum_{(i,j)\in A} c_{ij}y_{ij}^{s*} - \sum_{j\in\mathcal{N}}\sigma_j^+\Delta_j^{s+*} - \sum_{a\in\mathcal{A}}\sum_{l\in\mathcal{L}} c_{al}^+\delta_{al}^{s+*} - \sum_{a\in\mathcal{A}}\sum_{a'\in\mathcal{R}_a}\sum_{l\in\mathcal{L}} c_{aa'l}^+ d_{aa'l}^{s+*}\right\} \tag{39}$$

- S_r: the subset of indices (g,i) such as $\kappa_{gi} = 1$ in the r^{th} feasible solution, in other words, $S_r = \{(g,i) : g \in \mathcal{G}, i \in \mathcal{I}$ and g has been sent to first-stage facility i in the r^{th} feasible solution$\}$.

Using this notation, the master problem can be formally written as:

$$\max\theta \tag{40}$$

$$\theta \le \frac{1}{S}\sum_{s\in S}\left\{\sum_{m\in\mathcal{M}}\pi_m o_m^s - \sum_{(i,j)\in A} c_{ij}y_{ij}^s - \sum_{j\in\mathcal{N}}\sigma_j^+\Delta_j^{s+} - \sum_{a\in\mathcal{A}}\sum_{l\in\mathcal{L}} c_{al}^+\delta_{al}^{s+} - \sum_{a\in\mathcal{A}}\sum_{a'\in\mathcal{R}_a}\sum_{l\in\mathcal{L}} c_{aa'l}^+ d_{aa'l}^{s+}\right\} \tag{41}$$

$$\theta \le -\left(U-\theta_r\right)\left(\sum_{(g,i)\in S_r}\kappa_{gi} - \sum_{(g,i)\notin S_r}\kappa_{gi}\right) + \left(U-\theta_r\right)\left(|S_r|-1\right) + U \quad \forall r\in\mathcal{R} \tag{42}$$

$$\sum_{i\in\mathcal{I}}\kappa_{gi} = 1 \quad \forall g\in\mathcal{G} \tag{43}$$

$$\kappa_{gi} \le \epsilon_{gi} \quad \forall g\in\mathcal{G},\ i\in\mathcal{I} \tag{44}$$

$$\sum_{g\in\mathcal{G}} w_g^s\kappa_{gi} = \sum_{l\in\mathcal{L}} y_{il}^s \quad \forall i\in\mathcal{I},\ s\in S \tag{45}$$

$$\sum_{l\in\mathcal{L}} y_{il}^s - \Delta_i^{s+} \le \overline{C_i} \quad \forall i\in\mathcal{I},\ s\in S \tag{46}$$

$$\sum_{i\in\mathcal{I}} y_{il}^s - \Delta_l^{s+} \le \overline{C_l} \quad \forall l\in\mathcal{L}, s\in S \tag{47}$$

$$\sum_{g\in\mathcal{G}} \lambda_{ag}^s w_g^s\kappa_{gi} = \sum_{g\in\mathcal{G}} w_g^s u_{agi} \quad \forall a\in\mathcal{A}, i\in\mathcal{I},\ s\in S \tag{48}$$

$$\sum_{i\in\mathcal{I}} v_{ail}^s - \delta_{al}^{s+} \le \overline{P_{al}}\sum_{i\in\mathcal{I}} y_{il}^s \quad \forall a\in\mathcal{A},\ l\in\mathcal{L}, s\in S \tag{49}$$

$$\sum_{i\in\mathcal{I}} v_{ail}^s - d_{aa'l}^{s+} \le \overline{R_{aa'l}}\sum_{i\in\mathcal{I}} v_{a'il}^s \quad \forall a\in\mathcal{A}, a'\in\mathcal{R}_a, l\in\mathcal{L}, s\in S \tag{50}$$

$$o_m^s = \sum_{l\in\mathcal{L}} r_{ml}\sum_{i\in\mathcal{I}} v_{mil}^s \quad \forall m\in\mathcal{M}, s\in S \tag{51}$$

$$u_{agi}^s \le \overline{\Lambda_{ai}^s}\kappa_{gi} \quad \forall a\in\mathcal{A},\ g\in\mathcal{G},\ i\in\mathcal{I},\ s\in S \tag{52}$$

$$u_{agi}^s \le p_{ai}^s \quad \forall a\in\mathcal{A},\ g\in\mathcal{G},\ i\in\mathcal{I},\ s\in S \tag{53}$$

$$u_{agi}^s \ge p_{ai}^s - \left(1-\kappa_{gi}\right)\overline{\Lambda_{ai}^s} \quad \forall a\in\mathcal{A},\ g\in\mathcal{G},\ i\in\mathcal{I},\ s\in S \tag{54}$$

$$v^s_{ail} \geq \underline{\Lambda^s_{ai}} y^s_{il} \quad \forall a \in \mathcal{A},\ (i,l) \in A, s \in S \tag{55}$$

$$v^s_{ail} \geq \overline{\Lambda^s_{ai}} y^s_{il} + \overline{Y^s_{il}} p^s_{ai} - \overline{\Lambda^s_{ai} Y^s_{il}} \quad \forall a \in \mathcal{A},\ (i,l) \in A, s \in S \tag{56}$$

$$v^s_{ail} \leq \overline{\Lambda^s_{ai}} y^s_{il} \quad \forall a \in \mathcal{A},\ (i,l) \in A, s \in S \tag{57}$$

$$v^s_{ail} \leq \underline{\Lambda^s_{ai}} y^s_{il} + \overline{Y^s_{il}} p^s_{ai} - \underline{\Lambda^s_{ai}} \overline{Y^s_{il}} \quad \forall a \in \mathcal{A},\ (i,l) \in A, s \in S \tag{58}$$

$$\kappa_{gi} \in \{0,1\} \text{ and } y, p, o, \Delta^+, \delta^+, d^+, u, v, \theta \geq 0. \tag{59}$$

The constraints of the master can be divided into five categories: upper bounds on the objective function, capacity constraints, upper bounds on the grade and ratio, approximation of the bilinear terms, and integrality and non-negativity constraints. While the last four categories are self-explanatory given the discussion in the previous sections, the constraints in the first category [constraints (41) and (42)] deserve some discussion.

Recall that variable θ gives an approximation of the objective function value of DP (the non-linear stochastic mixed integer program). Constraints (41) simply state that θ can be at most the objective function value; that is, it has to be chosen in a way that is coherent with the values of the other variables of the master. Constraints (42) are similar to those introduced by Laporte and Louveaux (1993) in the context of stochastic integer programs with complete recourse, and they are gradually added to the master during the solution procedure to ensure that if an already visited first-stage solution r is selected, then a reward equal to θ_r is incurred. This reward is nothing else but the exact value of this solution, which is known since it has been already calculated during the previous iterations of the algorithm. Otherwise, there are no restrictions on θ except (41) and non-negativity. Note that θ is non-negative because it is always possible to send all groups to the waste dump and get a solution of value 0 (no costs, no revenues, no constraints to be violated since it is assumed that the waste dump has an unlimited capacity).

As discussed above, the master provides an upper bound. Moreover, once solved, a feasible solution to the original non-linear formulation can be determined. This is done as follows: Variables κ_{gi} are fixed, which allows setting and fixing variables p^s_{ai} as well, and in turn eliminating all bilinearities from the model. The resulting Benders sub-problem can be separated by scenario and reduces to independent problems that have the structure of a blending problem and can be solved as linear programs (LPs). Due to the presence of the variables Δ^{s+}_i, Δ^{s+}_l, δ^{s+}_{al}, and $d^{s+}_{aa'l}$ the sub-problems are always feasible. Furthermore, since the objective function coefficients (costs and profit parameters) are finite, any feasible solution must be bounded.

A formal statement of the proposed decomposition algorithm is given below.

Step 0 (Initialization) Set $r := 0, R := \emptyset, \overline{Z} = UB := \infty$ or an appropriate upper bound, $\underline{Z} = LB := -\infty$ or the value of the best-known feasible solution. Specify an optimality gap or tolerance ϵ

Step 1 (Solving the master problem) Set $r := r + 1$ and solve the current master problem. Let $(\kappa^r, y^r, p^r, o^r, \Delta^{+r}, \delta^{+r}, d^{+r}, u^r, v^r, \theta^r)$ be the so-obtained optimal solution and U^r be the value of this solution. Set $\overline{Z} := U^r$

Step 2 (Solving the Benders sub-problem) Fix the first-stage variables κ to κ^r, solve the resulting $|S|$ blending sub-problems, and compute the expected true value θ^r of this r^{th} feasible solution.

Step 3 (Updating the bounds and the incumbent) If $\theta^r > \underline{Z}$, then set $\underline{Z} := \theta^r$ and update the best-known feasible solution (the incumbent).

Step 4 (Verifying the stopping criterion and updating the master) If $\frac{\overline{Z}-\underline{Z}}{\underline{Z}} \leq \epsilon$, stop and return the best-known feasible solution. Otherwise, include r in$\mathcal{R}$, add to the master the optimality cut (42) and return to Step 1.

The algorithm summarized above yields a global optimal solution, within ϵ optimality gap, in a finite number of iterations. The convergence of the algorithm is guaranteed by the finite number of feasible first-stage solutions.

3.2.4 Using a surrogate

As can be seen from the results in the next section (Sect. 4), the proposed decomposition algorithm converges rapidly. However, using it to evaluate each mining sequence (neighbor solution) generated using the hyper-heuristic might not be the most suitable approach, as it would lead to prohibitive computational times (recall that computing the objective function of any neighbor solution entails solving the associated downstream sub-problem). Instead, a surrogate model can be used to provide fast approximations of the objective function and thus speed up the search. We propose using machine learning techniques, namely an artificial neural network as a surrogate model (surrogate evaluation function). The most promising neighbor solutions; that is, those achieving the best value for the surrogate, are subsequently evaluated using the decomposition algorithm.

The performance of a neural network often depends on the choice of the input features. We thus need to determine which features are the most appropriate for the problem under consideration. In this paper, we look at five different ways to define input features and we compare their impact on the prediction accuracy. In the first, referred to as **FULL**, the inputs are the grade of each attribute in each group under each scenario ($\lambda^s_{ag} \forall a \in \mathcal{A}, g \in \mathcal{G}, s \in S$) and the tonnage of each group under each scenario ($w^s_g \forall g \in \mathcal{G}, s \in S$). This means that the neural network is fed a very large data set consisting of $|\mathcal{G}||\mathcal{A}|\,|S| + |\mathcal{G}||S|$ input features. In the second, referred to as **TGPS**, tonnages and attribute grades are also used. However, rather than using data about each individual group as **FULL** does, aggregated data is used. Specifically, let $\overline{\mathcal{B}}$ be the set of blocks mined in the current solution. The inputs to be fed to the neural network are the total tonnage of these blocks ($\sum_{b \in \overline{\mathcal{B}}} w_b$) and the grade of each

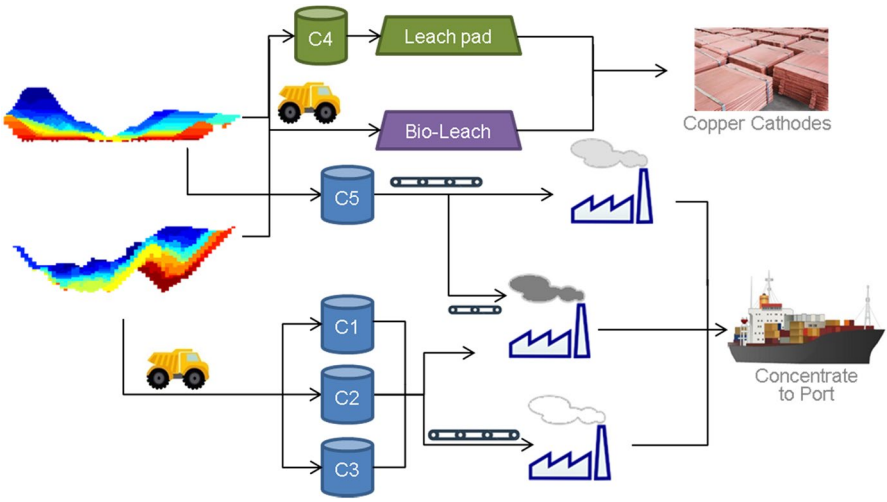

Fig. 1 Schematic representation of the copper–gold mining complex

attribute under each scenario ($\frac{\sum_{b\in\overline{B}}\alpha^s_{ab}w_b}{\sum_{b\in\overline{B}}w_b}\forall a\in\mathcal{A}, s\in S$). The third way, referred to as **TGA**, is very similar to **TGPS** except that the attribute grades are averaged over the scenarios. Thus, a total of $|\mathcal{A}|+1$ input features are fed to the neural network instead of $|\mathcal{A}|\,|S|+1$ in **TGPS**. The fourth way, referred to as **ENPS**, accounts for eligibility constraints. Recall that groups cannot be sent to just any first-stage facility $i\in\mathcal{I}$ and that the parameter ϵ_{gi} identifies the subset of facilities to which each group is *eligible* ($\epsilon_{gi}=1$ if group g is *eligible* for node $i\in\mathcal{I}$; 0, otherwise). In **ENPS**, the inputs are the total tonnage *eligible* for each first-stage facility under each scenario ($\sum_{g\in\mathcal{G}}\epsilon_{gi}w^s_g\forall i\in\mathcal{I}, s\in S$) and the grade of each attribute in the material *eligible* for each first-stage facility under each scenario ($\frac{\sum_{g\in\mathcal{G}}\epsilon_{gi}\lambda^s_{ag}w^s_g}{\sum_{g\in\mathcal{G}}\epsilon_{gi}w^s_g}\forall a\in\mathcal{A}, i\in\mathcal{I}, s\in S$). Finally, the fifth way, referred to as **ENA**, is similar to **ENPS** except that the tonnages and the attribute grades are averaged over the scenarios, which leads to $|\mathcal{A}|\,|\mathcal{I}|+|\mathcal{I}|$ input features as opposed to $|\mathcal{A}||\mathcal{I}|\,|S|+|\mathcal{I}|\,|S|$ in **ENPS**.

4 Computational results

The computational experiments aim to assess the efficiency of the decomposition algorithm (DA), the prediction accuracy of the neural network-based surrogate model (NN), and the effectiveness of using NN instead of DA when solving large-scale instances of the SSOMC. In what follows, we first describe the test instance. We then present the computational results. All tests were performed on an Intel(R) Xeon(R) CPU X5675 computer (3.07 GHz) with 96 Go of RAM running under Linux. The decomposition algorithm was coded in C++using IBM ILOG CPLEX 12.6.1.

Table 1 Summary of the computational times in seconds obtained when considering each of the two mines individually and the two mines simultaneously

Period	CPU time (in seconds)		
	Mine 1	Mine 2	Mine 1 and 2
1	1.91	2.48	5.4
2	1.73	2.96	4.43
3	1.26	3.66	7.99
4	1.73	2.59	3.94
5	1.68	3.18	8.47
6	1.21	3.29	4.77
7	0.37	3.84	8.21
8	1.02	10.09	5.71

Table 2 Number of cuts generated when considering blocks extracted from both mines

Period	Number of cuts
1	36
2	27
3	59
4	21
5	58
6	25
7	46
8	44

4.1 Dataset

The test instance consists of a copper–gold mining complex that comprises two mines, five crushers, three mills, one leach pad, one bioleach, and one waste dump. As can be seen from Fig. 1, material comes out of the mines and goes to one of the five crushers or to the bioleach or the waste dump. It gets further processed in one of the three mills or in a leach pad and ends up at the port or as copper cathodes. Note that the main sellable final product is copper, but the two mines also produce gold, iron, and molybdenum.

There are six main material types (sulphide high grade, sulphide low grade, oxide high grade, oxide low grade, mixed, and waste) and seven attributes of interest (copper total, copper soluble, gold, silver, arsenic, iron, and molybdenum). A set of 15 equiprobable scenarios is used to model geological uncertainty (grades and material types).

Of the mining complex's two mines, the first one (mine 1) has 68,450 blocks, while the second one (mine 2) has 103,390 blocks. In total, there are 171,840 blocks to be scheduled over 8 years. In addition to the slope, eligibility, mining and processing capacity constraints, there are also blending and ratio constraints that must be taken into account. Namely, at the mills, the level of arsenic, iron and the ratio of

Table 3 Number of input features fed to the neural network

Method	FULL	TGPS	TGA	ENPS	ENA
Number of inputs (I)	14,400	106	8	840	56

Table 4 Impact of the input features on the performance of the neural network

Method	FULL	TGPS	TGA	ENPS	ENA
Number of inputs (I)	14,400	106	8	840	56
r^2 training score	–	0.999815	0.997350	0.999741	0.999721
r^2 test score	–	0.999813	0.997505	0.999754	0.999713
Training time (in minutes)	–	2.97	5.69	7.98	4.62

copper total to iron need to be controlled, while at the bioleach, the ratio of copper soluble to copper total must be controlled.

4.2 Evaluation of the decomposition algorithm

In this section, results obtained with the decomposition algorithm (DA) are reported. For each mine, a mining sequence was generated using the random heuristic proposed in Lamghari and Dimitrakopoulos (2012). DA was applied to determine the flow of the extracted material.

As can be seen from the results in Table 1, DA converges quite rapidly. It obtains an optimal solution of the downstream sub-problem in just a few seconds. A comparison of the results in columns 2, 3, and 4 of this table show that the algorithm scales well with size. The difference in solution time can be explained by the number of major iterations, or equivalently, the number of cuts added to the master problem during the course of the algorithm. The number of cuts generated when solving the instance with two mines is given in Table 2. Note that in our implementation, instead of solving one master at each iteration, a *branch-and-Benders-cut method*, also known as *Benders-based branch-and-cut method*, was used. In this method, a single branch-and-cut tree is constructed, and the cuts are added to this tree during its exploration.

4.3 Evaluation of the neural network-based surrogate model

In this section, we evaluate the prediction accuracy of the neural network-based surrogate model (NN) and investigate the impact of input features on its performance. Details about the neural network are first presented, followed by a summary of computational results.

We used a standard neural network architecture, namely a two-layer perceptron network. The network consists of I input variables that provide information about the solution (mining sequence), two fully connected hidden layers of 50 neurons,

Table 5 Performance of the neural network when less data is used for training and testing

Method	FULL	TGPS	TGA	ENPS	ENA
Number of inputs (I)	14,400	106	8	840	56
r^2 training score	–	0.999051	0.994065	0.999506	0.995103
r^2 test score	–	0.999779	0.999359	0.999641	0.996395

and a single output neuron that provides an estimate of the objective function value given the inputs. Table 3 summarizes the values of I for each of the five ways of defining the input features discussed in Sect. 3.2.4. The logistic function was used as the activation function, and the Adam algorithm (adaptive moment estimation), proposed by Kingma and Ba (2014), was used to update the weights. The adaptive learning rate and the L2 regularization penalty were set to 10^{-5}, while the convergence tolerance value was set to 10^{-7}.

To obtain training and testing data, 100,000 different solutions (mining sequences) were generated using the low-level heuristics introduced in Sect. 3.1, and each solution was evaluated using the decomposition algorithm. The available data was then divided into two groups. The first, consisting of 95% of available data, was used to train the neural network. The remaining data was used for testing. Before training, the input features were individually standardized, and the output value was normalized.

Table 4 summarizes the results of the experiments. For each method used to define the input features, the training scores and the test scores are shown. The training time (in minutes) is also reported. The results indicate that method **FULL** is inappropriate as the neural network failed to converge. By considering fewer inputs (methods **TGPS**, **TGA**, **ENPS,** and **ENA**), the neural network was capable of estimating the objective function value with high accuracy. The results also show that introducing information about the individual scenarios leads to higher accuracy (**TGPS** outperforms **TGA** and **ENPS** slightly outperforms **ENA**). So overall, **TGPS** would rank first, **ENPS** second, **ENA** third, and **TGA**, where only 8 input features are fed to the neural network, would rank last.

While training the neural network took only a few minutes, as can be seen from the last row of Table 4, generating 100,000 solutions (mining sequences) and evaluating each solution using the decomposition algorithm to collect data for training and testing was time consuming and took approximately 3 weeks. The next experiments address the question whether considering less data can deteriorate the neural network performance.

Only 100 solutions were generated and evaluated using the decomposition algorithm. Again, the available data was divided into two groups: 95% for training the neural network and 5% for testing. The results are presented in Table 5.

If we compare the scores in Table 5 to those in Table 4, we observe that the performance of the neural network remains approximately the same. The neural network is capable of estimating the objective function value with high accuracy even when less data is used for training, indicating that the objective function value is

Table 6 Performance of algorithms **H-DA** and **H-NN**

Instance	Number of blocks	*IterMax*	**H-DA**		**H-NN**	
			Imp.%	Time (in minutes)	*Imp.*%	Time (in minutes)
Mine 1	68,450	5000	9.79	10,420.32	6.33	550.17
Mine 1 and 2	171,840	100	11.73	2943.24	3.52	65.03
		1000	–	–	7.45	355.96

highly correlated to the inputs. If we compare the five methods to define the input features, **TGPS** seems again to be the best. Thus, **TGPS** is used in all further experiments.

4.4 Effectiveness of using the neural network-based surrogate model instead of the decomposition algorithm

The purpose of the last experiments is to evaluate the effectiveness of the neural network-based surrogate model (NN) when embedded within the proposed matheuristic for the SSOMC. To this end, we compare the algorithm introduced in Sect. 3.1, henceforth referred to as **H-DA**, to a variant, **H-NN,** which is equivalent to **H-DA** except that the neighbor solutions generated by the hyper-heuristic (H) are first evaluated using NN instead of DA to identify the most promising ones. Only those most promising neighbor solutions are subsequently evaluated using DA. Tests are performed on two instances derived from the mining complex described in Sect. 4.1. In the first instance (*Mine 1*), only the smaller mine with 68,450 blocks is considered. In the second instance (*Mine 1 and 2*), the two mines, with a total of 171,840 blocks, are considered. The number of periods and scenarios is similar in the two instances (8 and 15, respectively).

The two algorithms, **H-DA** and **H-NN**, terminate after a specified number of iterations denoted *IterMax*. Because computational times tend to grow significantly when DA is used, for the small instance, *IterMax* was set to 5000 iterations for both algorithms, while for the large instance, it was set to 100 for **H-DA**, and two different values were considered for **H-NN**: 100 and 1000.

To assess the performance of **H-DA** and **H-NN**, the following measures are used:

- The percentage of improvement measured as $Imp.\% = \frac{Z^* - Z_0}{Z_0} \times 100$, where Z_0 and Z^* are the value of the initial solution and the value of the final solution, respectively. Note that to ensure a fair comparison, both **H-DA** and **H-NN** start from the same random initial solution.
- The CPU time, in minutes, denoted *Time*.

Table 6 summarizes the results obtained. As one would expect, **H-DA** outperforms **H-NN** in terms of solution quality; however, **H-NN** is significantly faster than **H-DA**. On the small instance *Mine 1*, **H-DA** provides a better solution than **H-NN** (the gap between the two solutions is 3.15%), but it is 20 times slower than **H-NN**

(7.24 days as opposed to 9.17 h). If we consider the largest instance with two mines, when the number of iterations is fixed to 100, the gap between the solution found by **H-DA** and that found by **H-NN** is 7.35%. The difference in CPU time is much more pronounced: one hour versus two days (CPU time is reduced by a factor of approximately 46 when **H-NN** is used). **H-NN** can further improve the solution quality when more iterations are allowed. After 1000 iterations, the gap between the two solutions is reduced to 3.83%. Performing 1000 iterations of **H-NN** requires less than 6 h, which is still much shorter than the 2 days required by **H-DA**.

Overall, the results clearly show the benefits of using the neural network-based surrogate model. While using the decomposition algorithm is better in terms of solution quality because the downstream sub-problem is solved to optimality, using this algorithm to evaluate each and every neighbor solution results in prohibitive computational times, especially when solving large-scale instances of the SSOMC. Significant performance improvement is achieved when the neural network-based surrogate model is embedded within the matheuristic to sample a subset of the neighborhood for exact evaluation with the decomposition algorithm.

5 Conclusions

This paper deals with the development of new models and solution approaches to address the large and complex problems faced by the mining industry when simultaneously optimizing mineral value chains (mining complexes) under uncertainty. A general mathematical model that integrates geological uncertainty and multiple mines sharing the same downstream infrastructure has been proposed to simultaneously optimize mining decisions and the flow of the extracted material. A new methodology that integrates components from exact algorithms (relaxation and decomposition), machine learning techniques (reinforcement learning and artificial neural networks), and heuristics (local improvement and randomized search) has been developed to efficiently solve the proposed formulation.

The proposed approach opens doors for new methodologies that capitalize on the synergies between machine learning and optimization techniques. Future work will follow this direction.

Acknowledgements The work in this paper was funded by NSERC CRDPJ 411270-10, NSERC Discovery Grant 239019-06, and the industry members of the COSMO Stochastic Mine Planning Laboratory: AngloGold Ashanti, Barrick, BHP, De Beers, Kinross, Newmont, and Vale. This support is gratefully acknowledged. The authors would like to thank two anonymous referees for their valuable comments and suggestions that helped improve the paper.

References

Androulakis I, Maranas C, Floudas C (1995) α bb: a global optimization method for general constrained nonconvex problems. J Glob Optim 7:337–363

Benders JF (1962) Partitioning procedures for solving mixed variables programming problems. Numer Math 4(1):238–252
Birge J, Louveaux F (2011) Introduction to stochastic programming, 2nd edn. Springer, New York
Del Castillo MF, Dimitrakopoulos R (2019) Dynamically optimizing the strategic plan of mining complexes under supply uncertainty. Resour Policy 60:83–93
Goodfellow R, Dimitrakopoulos R (2016) Global optimization of open pit mining complexes with uncertainty. Appl Soft Comput 40:292–304
Gupte A, Ahmed S, Dey S, Cheon M (2017) Relaxations and discretizations for the pooling problem. J Glob Optim 7:337–363
Haverly C (1978) Studies of the behavior of recursion for the pooling problem. ACM SIGMAP Bull 25:19–28
Kingma DP, Ba JL (2014) ADAM: a method for stochastic optimization. arXiv:1412.6980
Laporte G, Louveaux FV (1993) The integer L-shaped method for stochastic integer programs with complete recourse. Oper Res Lett 13:133–142
Lamghari A, Dimitrakopoulos R (2012) A diversified tabu search approach for the open-pit mine production scheduling problem with metal uncertainty. Eur J Oper Res 222:642–652
Lamghari A, Dimitrakopoulos R (2020) Hyper-heuristic approaches for strategic mine planning under uncertainty. Comput Oper Res 115:1–18. https://doi.org/10.1016/j.cor.2018.11.010
McCormick G (1976) Computability of global solutions to factorable nonconvex programs: part I—convex underestimating problems. Math Program 10:147–175
Montiel L, Dimitrakopoulos R (2015) Optimizing mining complexes with multiple processing and transportation alternatives: an uncertainty-based approach. Eur J Oper Res 247(1):166–178
Montiel L, Dimitrakopoulos R (2018) Simultaneous stochastic optimization of production scheduling at Twin Creeks mining complex. Nevada Min Eng 70(12):48–56
Montiel L, Dimitrakopoulos R, Kawahata K (2016) Globally optimising open-pit and underground mining operations under geological uncertainty. Min Technol 125(1):2–14
Zhang J, Nault BR, Dimitrakopoulos R (2019) Optimizing a mineral value chain with market uncertainty using benders decomposition. Eur J Oper Res 274(1):227–239