Applied Soft Computing 110 (2021) 107644

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Production scheduling in industrial mining complexes with incoming new information using tree search and deep reinforcement learning

Ashish Kumar *, Roussos Dimitrakopoulos

COSMO – Stochastic Mine Planning Laboratory, Department of Mining and Materials Engineering, McGill University, FDA Building, 3450 University Street, Montreal, Quebec, H3A 0E8, Canada

ARTICLE INFO

Article history:
Received 7 May 2020
Received in revised form 9 November 2020
Accepted 19 June 2021
Available online 30 June 2021

Keywords:
Artificial intelligence
Production scheduling
Mining complexes
Reinforcement learning
New information

ABSTRACT

Industrial mining complexes have implemented digital technologies and advanced sensors to monitor and gather real-time data about their different operational aspects, starting from the supply of materials from the mineral deposits involved to the products provided to customers. However, technologies are not available to respond in real-time to the incoming new information to adapt the short-term production schedule of a mining complex. A short-term production schedule determines the daily/weekly/monthly sequence of extraction, the destination of materials and utilization of processing streams. This paper presents a novel self-learning artificial intelligence algorithm for mining complexes that learns, from its own experience, to adapt the short-term production scheduling decisions by responding to incoming new information. The algorithm plays the game of short-term production scheduling on its own using a Monte Carlo tree search to train a deep neural network agent that adapts the short-term production schedule with incoming new information. The deep neural network agent evaluates the short-term production scheduling decisions and, in parallel, performs searches using the Monte Carlo tree search to generate experiences. The experiences are then used to train the agent. The agent improves the strength of the tree search, which results in an even stronger self-play to generate better experiences. An application of the proposed algorithm at a real-world copper mining complex shows its exceptional performance to adapt the 13-week short-term production schedule almost in real-time. The adapted production schedule successfully meets the different production requirements and makes better use of the processing capabilities, while also increasing copper concentrate production by 7% and cash flows by 12% compared to the initial production schedule. A video of the proposed algorithm can be found at https://youtu.be/_gSbzxMc_W8.

1. Introduction

Artificial intelligence (AI) algorithms have already been developed for applications in different engineering fields, but have not been developed and extended for short-term production scheduling in industrial mining complexes [1,2]. A mining complex is an integrated value chain with multiple interrelated components, such as raw material suppliers (mineral deposits), heavy machinery (shovels and trucks), destinations (crushers, stockpiles, and waste dumps), processing streams (processing mills and leach pads), tailings, and customers. Heavy machinery extracts and transports raw materials to the destinations. The materials from the destinations are then transported to processing streams, which process the materials to generate products that are delivered to different customers. A long-term production schedule is developed for a mining complex to provide the annual strategic decisions, targets, and forecasts while maximizing the cumulative discounted cash flows and accounting for supply and market uncertainties [3–7]. A short-term production schedule, i.e. an operational production schedule at a monthly/weekly/daily timescale, is then generated within the predefined long-term schedule and aims to ensure compliance with the long-term targets while maximizing cash flows. The major short-term production scheduling decisions in a mining complex are extraction sequencing, destination policies, and processing stream utilization. Extraction sequencing refers to determining the location and time of raw material extraction from the mineral deposit while adhering to mine slope stability and equipment movement restrictions. Mine slope stability constraints state that a block cannot be extracted safely until all overlying blocks within a predefined inclination are extracted first. Equipment movement restrictions state that a block cannot be extracted until one of the surrounding blocks is extracted first to provide access. Destination policies refer to finding the destination of the extracted raw

* Correspondence to: Vale, Digital Transformation Department, 2060 Flavelle Boulevard, Mississauga, Ontario, L5K 1Z9, Canada.
E-mail addresses: ashish.kumar@mail.mcgill.ca (A. Kumar), roussos.dimitrakopoulos@mcgill.ca (R. Dimitrakopoulos).

https://doi.org/10.1016/j.asoc.2021.107644

materials while satisfying material eligibility conditions. Material eligibility conditions state that specific types of materials cannot go to some destinations due to limitations with their downstream recovery process. Processing stream utilization refers to finding what proportions of materials to send from a destination to different processing locations (downstream processes) while respecting the mass flow conservation constraints. In addition, various production limit constraints related to the quality and quantity of materials mined, handled, processed, and sold also need to be respected in the short-term production schedule. Like any complex network, uncertainty is inherent in a mining complex. This uncertainty stems from the quality, quantity, and spatial location of raw material within the mineral deposits – referred to as supply uncertainty – and from the production capabilities of different machinery, destinations, and processing streams—referred to as equipment performance uncertainty. A set of equally probable scenarios/simulations are generated using stochastic simulation methods to quantify supply uncertainty and variability [8–11], and equipment performance uncertainty [2].

A mining complex collects new information during its day-to-day operations with conventional and new digital technologies, specifically advanced sensors and monitoring devices. The new information acquired can pertain to the flow of materials [12], equipment location [13], equipment production capabilities [14,15], and the quality and quantity of the material extracted, handled, and processed [16–22]. The radio-frequency identification tags [12] are dropped in the blastholes and are then read by a reader on the conveyor belt during the flow of materials from the mine to customers, to identify the origin of materials. The global position system pinpoints the location and status of the shovels and trucks in a mining complex [13]. Control unit sensors [14,15] provide information about health, fuel consumption, gas emission and more about the equipment. Infrared sensors pass a beam of infrared radiations through the surface of materials and measures the frequencies and amount of energy of the beam absorbed to determine the characteristics of the mineralization [16,20–22]. Laser-induced breakdown spectroscopy sensors ablate the surface of materials by a laser beam that breaks down the surface into a plasma and then measures the radiation emitted during the cooling of plasma, which is then used to characterize the mineralization [17]. Dual-energy X-ray transmission sensors bombard broad-band radiation to penetrate the surface of materials. The amount of absorption is then used to characterize the material composition [18,19].

This new information provides an opportunity to better understand the state of a mining complex and to respond accordingly by adapting the short-term production schedule quickly (in real-time) to better meet the long-term targets. However, the core short-term scheduling decisions, such as the extraction sequence and destination policies, comprise a difficult combinatorial optimization problem and are computationally expensive to reoptimize with existing techniques [23]. In addition, the incoming new information is partial and noisy, and is, therefore, uncertain. Thus, the new information cannot be used directly in an optimization model to make short-term production scheduling decisions. The uncertain incoming new information needs to be assimilated to update the material supply and equipment performance uncertainties. Updating the supply uncertainty is more challenging compared to the equipment performance uncertainty due to the presence of multivariate spatial correlation. Ensemble Kalman filter is a well-known, two-step assimilation process that has been used to assimilate uncertain new information in petroleum reservoirs for decades [24–26] and has, recently, been extended to mineral deposits [27,28].

AI agents, such as deep neural networks and convolution neural networks, are function approximators that are trained to make decisions by responding to the incoming new information/observations/states [29] via reinforcement learning algorithms that use their own experiences generated by interacting with an environment (a model that mimics the intricacies of the industrial process under consideration). The updated supply and equipment performance uncertainties allow such an AI agent to better perceive the updated state of a mining complex to adapt the short-term production schedule in real-time. The adapted production schedule is then used to perform day-to-day operations. In parallel, a new AI agent is trained with the updated supply and equipment performance uncertainties to further learn from the incoming new information. Therefore, a continuous updating framework is necessary, which first updates the uncertainties with the incoming uncertain new information, and then learns and adapts the short-term production schedule of a mining complex with AI agents. Benndorf and Buxton [30] and Hou et al. [31] proposed a framework that updates the supply uncertainty with incoming new information but relies on existing optimization techniques for adapting the production scheduling decisions. Paduraru and Dimitrakopoulos [32] proposed a policy gradient reinforcement learning algorithm for deciding the short-term destination of materials in a single product mining complex. Kumar et al. [33] further extended this algorithm for multiple product mining complexes and proposed a continuous updating framework that updates the supply uncertainty with an extended ensemble Kalman filter (EnKF) and adapts the short-term destination of materials with an extended policy gradient reinforcement learning algorithm. However, the method does not adapt the extraction sequence, destination policies and processing stream utilization decisions simultaneously and does not update the equipment performance uncertainty.

The work presented herein proposes a novel self-play reinforcement learning algorithm that adapts all the short-term production scheduling decisions simultaneously in a mining complex. The proposed algorithm is inspired by the AlphaGo and AlphaGoZero algorithms [34,35]. The proposed algorithm plays the game of short-term production scheduling by itself using a Monte Carlo tree search to train a deep neural network agent to learn how to adapt the short-term production schedule with incoming new information in an operating mining environment. Additionally, the work also proposes a Monte Carlo simulation algorithm to update the equipment performance uncertainty. In the following sections, the proposed method that learns and adapts short-term decisions in a mining complex is detailed first. Next, an application at a copper mining complex is presented to show the efficiency of the proposed algorithm in terms of learning and adapting the short-term production schedule to generate more metal and achieve improved compliance with production targets. Conclusions and directions for future research follow.

2. Method

This section describes the algorithm for learning short-term production scheduling and then adapting the short-term production schedules with incoming new information in an operating mining environment. The complete workflow of the proposed algorithm is shown in Fig. 1. The initial supply and equipment performance uncertainties are used to generate a stochastic short-term production schedule and forecast within the long-term production schedule. A mining complex operates with this short-term production schedule and generates new information with sensors installed on its various components during operations (Fig. 1(b)).

The new information is first used to update supply and equipment performance uncertainty with ensemble Kalman filter and Monte Carlo simulation method respectively (Fig. 1(c)). A deep

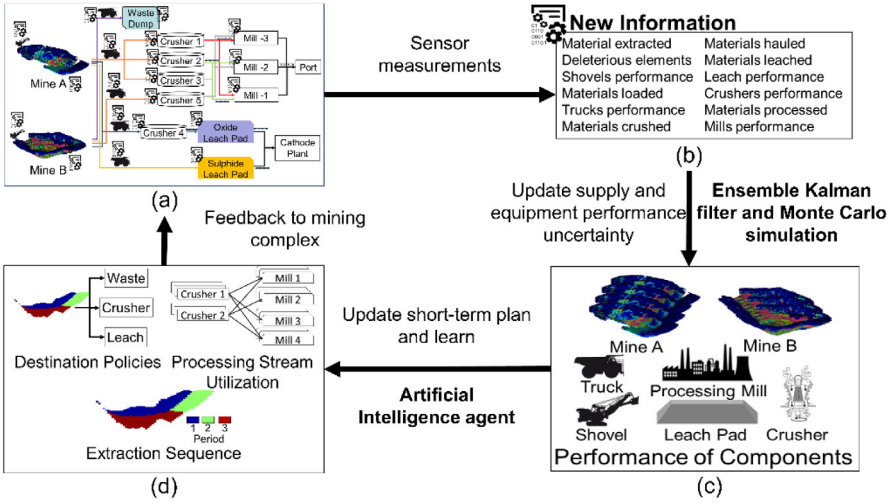

Fig. 1. The complete workflow of the proposed algorithm.
Source: Modified from Kumar et al. [33].

neural network agent (already trained via self-play reinforcement learning algorithm with the initial supply and equipment uncertainty) is then used to adapt the short-term production plan with the updated uncertainties (Fig. 1(d)). The adapted schedule and its respective forecasts are then fed back to the operation. The deep neural network agent is trained in parallel using the updated uncertainties. The workflow then continues where more new information is collected and further updating, adapting, and learning is performed. Section 2.1 details the process of transforming raw material into sellable products in a mining complex. The algorithm for learning and adapting the short-term production schedule is detailed next. A list of the notations used in this section is available in Appendix A.

2.1. Modelling a mining complex

Raw materials in a mining complex can be supplied from various sources, such as multiple mines, denoted by a set M. A mine is developed within the mineral deposit to extract materials. The materials in the mines consist of revenue-generating properties, denoted by a set $\mathbb{P}_{\mathbb{R}}$, deleterious properties, denoted by a set $\mathbb{P}_{\mathbb{D}}$, and rock mass, denoted by $\mathbb{P}_{\mathbb{M}}$. The material in the mines is discretized into a set of three-dimensional volumes called mining blocks, denoted by a set $\mathbb{Z}_m(x)$, where x denotes the spatial location of the block within the mine $m \in M$. The quality of material in the mines is uncertain. Therefore a set of initial, I, stochastic simulations, denoted by $\mathbb{S}_{l,a,m}$, of mining blocks, $\mathbb{Z}^{l,s}_{a,m}(x)$, about multiple spatial correlated properties $a \in \mathbb{P}_{\mathbb{R}} \cup \mathbb{P}_{\mathbb{D}}$ is generated based on existing drill hole information, denoted by $dH^{l}_{a,m}$, to quantify the supply uncertainty for mine $m \in M$. Each mine has a set of associated shovels, $\mathcal{S}_m$.The material extracted with the shovels is loaded into trucks, denoted by a set $\mathcal{T}_m$, available at each mine $m \in M$. The trucks haul the materials to their destinations, denoted by a set $\mathcal{D}$. The materials from the destinations are then transported via conveyor belts to processing streams, denoted by a set $\mathcal{P}$, to generate products, which are then transported and sold to customers/markets. The processing streams have restrictions on the quantity of deleterious properties in the final product. In addition, the trucks, shovels, destinations, and processing streams, $\mathbb{E} = \{\mathcal{T}, \mathcal{S}, \mathcal{D}, \mathcal{P}\}$, have restrictions on their production capacity, denoted by $\mathbb{P}_{\mathbb{P},e}(T), \forall e \in \mathbb{E}, T \in [1, N_{week}]$, where N_{week} denotes the total number of weeks. A set of initial stochastic simulations of production capabilities, denoted by $\mathbb{S}'_{l,e}(T), \forall e \in \mathbb{E}, T \in [1, N_{week}]$, is generated based on historical production information, denoted by eP^{I}_{e}, to quantify the performance uncertainty of components $e \in \mathbb{E}$. The initial supply and equipment performance uncertainties are used in a self-play reinforcement learning algorithm to train a deep neural network (DNN). The trained DNN agent is then used to make the short-term production scheduling decisions in real-time with the incoming new information.

The first short-term production scheduling decision in a mining complex is to decide when to extract a mining block. However, the multiple shovels located within the mine operate simultaneously; therefore, this decision variable is modified in such a way to take this into account. Let the blocks that can be extracted by a shovel be denoted by a set $\mathbb{Z}_m(s_i), \forall s_i \in \mathcal{S}_m, m \in M$. Let B represent a set whose elements are computed by finding all possible combinations that have exactly one element from each set $\mathbb{Z}_m(s_i), \forall s_i \in \mathcal{S}_m, m \in M$. The second short-term production scheduling decision in a mining complex is to decide the destination of the extracted block. Let $\mathcal{D}(z), \forall z \in \mathbb{Z}_m(x), m \in M$ denote the set of permissible destinations where each mining block can be sent. Therefore, the decision variable/action is defined by $x_{b,d,t}, \forall b \in \mathcal{B}, d \in \{\mathcal{D}(i), \forall i \in b\}$ and represents whether (1) or not (0) a set of mining blocks b is extracted, and each block within the set b is sent to a set of destinations d at time step t.

$$x_{b,d,t} \leq x_{k,d',t-1}, \forall k \in b_V, d' \in \{D(k), \forall k \in b_V\}, \quad b \in B, d \in \{D(i), \forall i \in b\} \tag{1}$$

In order to extract a block safely, it is necessary to extract all its overlying blocks to ensure the stability of the mine wall, and at least one surrounding block to provide access to the equipment. The vertical predecessor $V(i)$ defines all the mining blocks that overlie a block i, and is calculated by finding blocks that are within a predefined vertical inclination (known as the slope angle), as shown by the dashed lines in Fig. 2. Let $b_V = \{V(i), \forall i \in b\}$ be a set consisting of all the blocks that lie above the blocks in a set b. Therefore, an action $x_{b,d,t}$ is only eligible to be made if all the overlying blocks (represented by decisions $x_{k,d',t-1}, \forall k \in b_V$) are extracted first, as shown in Eq. (1).

Horizontal and vertical successor mining blocks define mine equipment access constraints. Horizontal successors $H(i,r)$ and vertical successors $V(i,r)$ for a block $i \in b$, define all the surrounding blocks within a given radius r of a block i in the horizontal (solid line in Fig. 2) and vertical directions (dotted line in Fig. 2), respectively.

$$x_{b,d,t} \leq \sum_{k \in b_H} x_{k,d',t-1}, \forall k \in b_H, d' \in \{D(k), \forall k \in b_H\},$$

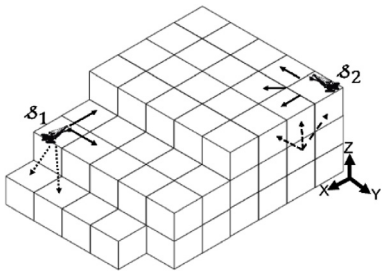

Fig. 2. Permissible block extraction representation in a mining operation.

$$b \in B, d \in \{D(i), \forall i \in b\} \tag{2}$$

Let b_H be a set consisting of k sets, where each set k consists of one of the blocks that surround a block i in the set b within a given radius r in either the horizontal or vertical directions. Therefore, an action $x_{b,d,t}$ is permissible only if at least one of the horizontal or vertical successors blocks (represented by decisions $x_{k,d',t-1}, \forall k \in b_H$) is extracted first, as shown in Eq. (2). The blocks that satisfy Eqs. (1) and (2) also need to satisfy material classification conditions related to their destination because the processing streams that are fed by destinations are designed to process a specific type of material. For example, a sulphide ore processor, by the construction of its metal recovery process, cannot accept oxide type materials. Therefore, a material classification condition is defined that finds the permissible destinations for a mining block. For example, if a mining block i has more soluble copper, then the permissible destination for this block can either be a destination that feeds the oxide processing stream or a waste dump. The permissible destinations of a mining block under supply uncertainty are determined as the most probable destinations (ties are broken randomly). The materials at the destinations incur a cost denoted by $C_{a,d}, \forall a \in \mathbb{P}_{\mathbb{M}}, d \in \mathcal{D}$ and are then sent to the processing streams. The third short-term production scheduling decision is to determine how to utilize the processing streams in a mining complex, represented by $y_{a,d,p,t,s_l^{\mathbb{J}}}$. The decision variable $y_{a,d,p,t,s_l^{\mathbb{J}}}$ denotes the amount of attribute $a \in \mathbb{P}_{\mathbb{R}} \cup \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}$ sent from a destination $d \in \mathcal{D}$ to a processing location $p \in \mathcal{P}$ at time step t under joint uncertainty scenario $s_l^{\mathbb{J}} \in \mathbb{S}_l^{\mathbb{J}}$. Here, joint uncertainty scenarios refer to uncertainty in both the supply of materials, $\mathbb{S}_{l,a,m}$, and performance of equipment, $\mathbb{S}'_{l,e}$. The processing streams recover the metal with a factor of $r_{a,p}, \forall a \in \mathbb{P}_{\mathbb{R}}, p \in \mathcal{P}$ and incur a processing cost of $C_{a,p}, \forall a \in \mathbb{P}_{\mathbb{M}}, j \in \mathcal{P}$. The products are then transported and sold to the customers with a price of $P_{a,p}, \forall a \in \mathbb{P}_{\mathbb{R}}, p \in \mathcal{P}$.

2.2. A self-play reinforcement learning algorithm

The algorithm proposed to learn to adapt the short-term production schedule (extraction sequence, destination policies, and processing stream utilization decisions, simultaneously) of a mining complex consists of a deep neural network (DNN) agent f_θ with parameters θ and a Monte Carlo tree search (MCTS) within a self-play reinforcement learning architecture, as shown in Fig. 3. The proposed algorithm starts at time $t = 1$ with the input state s_1 of a mining complex. The state s_t of a mining complex at any time step t is described by the performance of its different components and is herein defined by (i) the position of the different shovels (the mining blocks that the shovels are extracting) located in the multiple mines, (ii) the supply uncertainty $\mathbb{S}_{l,a,m}$ of blocks located within a neighbourhood of the different shovels, (iii) the equipment performance uncertainty of different components $\mathbb{S}'_{l,e}$, and (iv) history about the quality and quantity of material at destinations and processing streams. At each state, a set of permissible actions $x_{b,d,t}$ is identified using the mine wall slope stability, equipment access, and material classification criteria defined in Section 2.1. The state s_t is then fed to a DNN agent which outputs both a vector of selection probabilities and a vector of scalar evaluations $(p_t, v_t) = f_\theta(s_t)$ for all the permissible actions $x_{b,d,t}$ with the given input state s_t, as shown in Fig. 3(a).

For each time step $t < T$ an MCTS search α_θ is executed using tree policy P_π (see Section 2.2.1), guided by the DNN agent f_θ until the end time step T. The MCTS outputs probabilities π_t of selecting each action and scalar evaluations z_t for each action from the set of permissible actions at time step t as shown in Fig. 3(b). An action $x_{b,d,t}$ at time step t is selected proportional to $\pi_t(x_{b,d,t}|s_t)$ (selection probabilities). These tree search probabilities π_t select much stronger actions $x_{b,d,t}$ than the raw probabilities p_t of the DNN agent $f_\theta(s_t)$. At this point, the DNN agent's parameters θ are updated to make its selection probabilities and scalar evaluations $(p_t, v_t) = f_\theta(s_t)$ match the improved search probabilities π_t and evaluations z_t more closely (see Fig. 3(a)). These new parameters make the MCTS search even stronger in the next round of self-play. The selected action $x_{b,d,t}$ is then used to update the input state as s_{t+1} and the process is repeated until $t = T, \forall T \in [1, N_{week}]$.

2.2.1. Monte Carlo tree search

The MCTS uses the DNN agent f_θ to guide its search. The search involves generating a rollout simulation of a feasible (that satisfies Eqs. (1) and (2)) short-term production schedules (extraction sequence, destination policies, and processing stream utilization).

For simplicity and without loss of generality, consider next a case where two shovels are located at the extreme ends of the mine (represented by a 2-dimensional grid), as shown in Fig. 4(a). The colour of the grid cell indicates whether the block is extracted (coloured) or not (white) and which destination it is sent to after it is extracted (red—crusher, green—leach, yellow—waste, and white—not extracted). Each node in the search tree represents a state s_t, and contains edges which represent a state–action pair $s_t, x_{b,d,t}$ where $x_{b,d,t}$ is a permissible child/action of the node s_t. Each edge in the search tree stores a set of statistics as prior probability $P(s_t, x_{b,d,t})$, visit count $N(s_t, x_{b,d,t})$, total action value $W(s_t, x_{b,d,t})$ and mean action-value $Q(s_t, x_{b,d,t})$. Multiple roll-out simulations are executed by iterating over four phases (a–d in Fig. 4) and then selecting an action proportional to the search probabilities, i.e., $\pi_t(x_{b,d,t}|s_t) \propto N(s_t, x_{b,d,t}) / \sum_{b',d'} N(s_t, x_{b',d',t})$ (Fig. 4(e)). Here, $\sum_{b' \in \mathcal{B}, d' \in \{\mathcal{D}(i), \forall i \in b'\}} N(s_t, x_{b',d',t})$ represents the sum of visit counts of all the permissible actions of the node s_t. The details of the MCTS in terms of selection, expansion, evaluation, simulation, and play phase are described next.

2.2.1.1. Selection. The selection phase (Fig. 4(a)) is the first step in the tree search, which begins at the root node, i.e. in the input state s_1 of the MCTS, and finishes when the leaf node s_L at time step L is encountered. At each of these time steps $t < L$, an action $(x_{b,d,t})$ is selected that maximizes the upper confidence bound, $Q(s_t, x_{b,d,t}) + U(s_t, x_{b,d,t})$, calculated using the statistics stored in the tree search, as shown in Eq. (3), a variant of the PUCT algorithm [35].

$$x_{b,d,t} = \underset{x}{\operatorname{argmax}} \left(Q(s_t, x_{b,d,t}) + U(s_t, x_{b,d,t}) \right) \tag{3}$$

$$U(s_t, x_{b,d,t}) = c_{puct} P(s_t, x_{b,d,t}) \frac{\sqrt{\sum_{b' \in \mathcal{B}, d' \in \{\mathcal{D}(i), \forall i \in b'\}} N(s_t, x_{b',d',t})}}{1 + N(s_t, x_{b,d,t})} \tag{4}$$

where, $U(s_t, x_{b,d,t})$ is calculated using Eq. (4) that uses the prior probability $P(s_t, x_{b,d,t})$ generated by the DNN agent and the visit

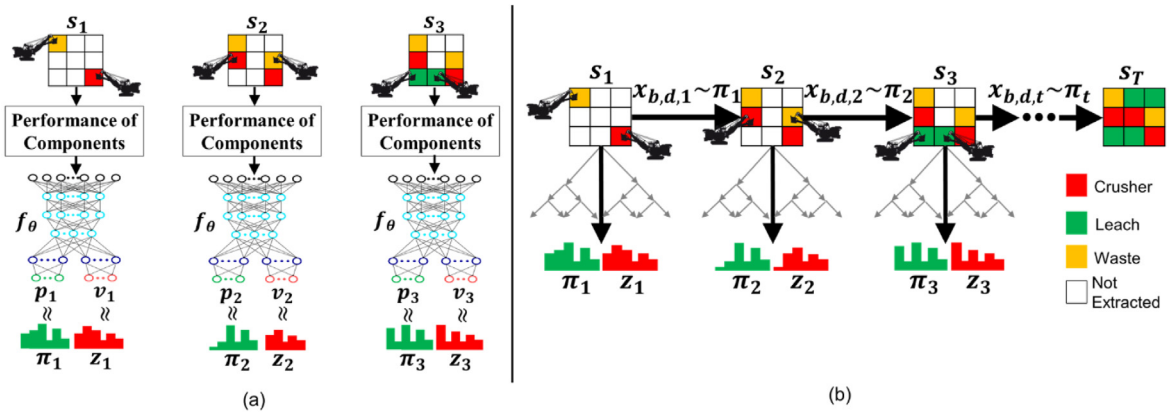

Fig. 3. Self-play reinforcement learning architecture for short-term production scheduling in mining complexes. It includes (a) DNN agent training by using the output of the Monte-Carlo tree searches and (b) Monte-Carlo tree searches guided by the DNN agent are executed to generate selection probabilities and scalar evaluation for all permissible actions at each time step.

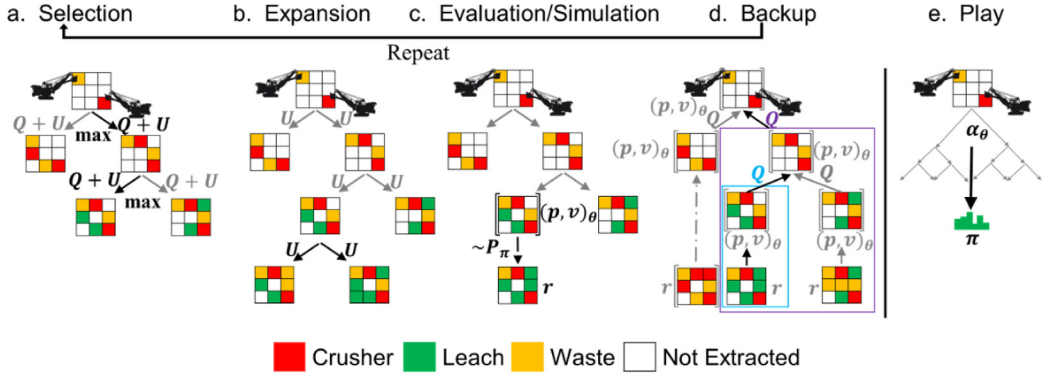

Fig. 4. Monte Carlo tree search phases for short-term production scheduling in a mining complex are: (a) selection of blocks to extract and send to a destination from multiple faces; (b) expansion to find the next permissible block extraction and destination decisions; (c) evaluation of the next permission decisions via the DNN and simulation until the end of time horizon using MCTS guided by the DNN; (d) backup of values obtained after the simulation to inform the edges in the tree; and (e) repeating the steps a–d multiple times to generate search probabilities and then taking an action by sampling the probabilities.

count statistics $N\left(s_t, x_{b,d,t}\right)$ stored in the MCTS for state–action pairs $s_t, x_{b,d,t}$. c_{puct}, is a constant that determines the level of exploration in the search, such that the search prefers actions with a high prior probability and a low visit count, but asymptotically prefers actions with high mean action-value.

2.2.1.2. Expansion. The leaf node s_L encountered during the selection phase of MCTS is then expanded (Fig. 4(b)) to find the next permissible actions that satisfy Eqs. (1) and (2), and are then added to the search tree as children of node L. The different statistics of the edges $s_L, x_{b,d,L}$ connecting node s_L to its children nodes $x_{b,d,L}$ are initialized to zero, i.e. $N\left(s_L, x_{b,d,L}\right) = 0, Q\left(s_L, x_{b,d,L}\right) = 0, P\left(s_L, x_{b,d,L}\right) = 0, P\left(s_L, x_{b,d,L}\right) = 0$.

2.2.1.3. Evaluation/simulation. The leaf node s_L state–action pair $s_L, x_{b,d,L}$ is then evaluated in two ways (Fig. 4(c)), first by the DNN agent, and second by performing a rollout simulation until the end of time T using a tree policy P_π in MCTS. In comparison to playing games, the evaluation of short-term production scheduling in a mining complex (see Section 2.1) is more intricate. For instance, in the game of Go, the outcome is either a win or a loss, but, in a mining complex, the outcome is an expected monetary gain from selling the products minus any losses and costs incurred to produce the products. In addition, the evaluation from the DNN agent requires some inputs about the history, i.e. the quantity and quality of materials at the destination and the processing streams (see Section 2.2). The history of the quantity of attribute $a \in \mathbb{P}_{\mathbb{R}} \cup \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}$ at time step L at different destinations d in a state s_L is calculated by observing all the actions taken to reach node $x_{b,d,L}$. The quantity of revenue generating and deleterious elements at destination $d \in \mathcal{D}$ at time step L under supply uncertainty $\mathbb{s}_{I,a,m} \in \mathbb{S}_{I,a,m}$ is calculated as:

$$v_{a,d,L,\mathbb{s}_I} = \sum_{t<L} \sum_{b' \in b} \mathbb{Z}^{I,\mathbb{s}}_{a,m}\left(b'\right) \cdot \mathbb{Z}^{I,\mathbb{s}}_{\mathbb{P}_{\mathbb{M}},m}\left(b'\right) \cdot x_{b,d,t},$$
$$\forall d \in \mathcal{D}, \mathbb{s}_I \in \mathbb{S}_{I,a,m}, a \in \mathbb{P}_{\mathbb{R}} \cup \mathbb{P}_{\mathbb{D}} \tag{5}$$

where, $\mathbb{Z}^{I,\mathbb{s}}_{a,m}(b')$ represent the quantity of attribute a of a mining block b' at mine m in the initial I supply uncertainty scenarios $\mathbb{s}_I$. The mass of materials at destination $d \in \mathcal{D}$ at time step L under supply uncertainty $\mathbb{s}_{I,a,m} \in \mathbb{S}_{I,a,m}$ is calculated as:

$$v_{\mathbb{P}_{\mathbb{M}},d,L,\mathbb{s}_I} = \sum_{t<L} \sum_{b' \in b} \mathbb{Z}^{I,\mathbb{s}}_{\mathbb{P}_{\mathbb{M}},m}(b') \cdot x_{b,d,t}, \forall d \in \mathcal{D}, \mathbb{s}_I \in \mathbb{S}_{I,\mathbb{P}_{\mathbb{M}},m} \tag{6}$$

The history of the quantity of attribute $a \in \mathbb{P}_{\mathbb{R}} \cup \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}$ at processing stream $p \in \mathcal{P}$ at time step L is calculated by optimizing the processing stream utilization decisions, $y_{a,d,p,t,\mathbb{s}^{\mathbb{J}}_I}$. For this, a stochastic mathematical programming model [3] is solved using the simplex method that maximizes the objective function $f_L\left(\mathbb{s}^{\mathbb{J}}_I\right)$, as shown in Eq. (7).

$$f_L\left(\mathbb{s}^{\mathbb{J}}_I\right) = \sum_{p \in \mathcal{P}} \sum_{a \in \mathbb{P}_{\mathbb{R}}} P_{a,p} \cdot v_{a,p,L,\mathbb{s}^{\mathbb{J}}_I} \cdot r_{a,i} - \sum_{p \in \mathcal{P}} \sum_{a \in \mathbb{P}_{\mathbb{M}}} C_{a,p} \cdot v_{a,p,L,\mathbb{s}^{\mathbb{J}}_I}$$
$$- \sum_{p \in \mathcal{P}} \sum_{a \in \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}} \left(c^{+}_{a,p} \cdot d^{+}_{a,p,L,\mathbb{s}^{\mathbb{J}}_I} + c^{-}_{a,p} \cdot d^{-}_{a,p,L,\mathbb{s}^{\mathbb{J}}_I} \right), \forall \mathbb{s}^{\mathbb{J}}_I \in \mathbb{S}^{\mathbb{J}}_I \tag{7}$$

subjected to:

$$v_{a,p,L,s_l^{\mathbb{J}}} - d^{+}_{a,p,L,s_l^{\mathbb{J}}} \leq U_{a,p,T}, \forall a \in \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}, p \in \mathcal{P}, s_l^{\mathbb{J}} \in \mathbb{S}_l^{\mathbb{J}} \quad (8)$$

$$v_{a,p,L,s_l^{\mathbb{J}}} + d^{-}_{a,p,L,s_l^{\mathbb{J}}} \geq L_{a,p,T}, \forall a \in \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}, p \in \mathcal{P}, s_l^{\mathbb{J}} \in \mathbb{S}_l^{\mathbb{J}} \quad (9)$$

$$\sum_{p \in \mathcal{P}} y_{a,d,p,L,s_l^{\mathbb{J}}} = 1, \forall d \in \mathcal{D}, s_l^{\mathbb{J}} \in \mathbb{S}_l^{\mathbb{J}} \quad (10)$$

where,

$$v_{a,p,L,s_l^{\mathbb{J}}} = \sum_{d \in \mathcal{D}} y_{a,d,p,L,s_l^{\mathbb{J}}} \cdot v_{a,d,L,s_l}, \forall a \in \mathbb{P}_{\mathbb{R}} \cup \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}, p \in \mathcal{P}, s_l^{\mathbb{J}} \in \mathbb{S}_l^{\mathbb{J}} \quad (11)$$

The objective function (Eq. (7)) computes the profit generated by processing the materials, minus the cost of processing the material and penalties due to deviations from the upper $U_{a,p,T}$ and lower $L_{a,p,T}$ production capacities. Here, $c^{+}_{a,p}$ and $c^{-}_{a,p}$ is the penalty cost for deviating from the upper and lower production capacities, respectively. $v_{a,p,L,s_l^{\mathbb{J}}}$ is the quantity of attribute a at processing location p under joint uncertainty scenario $s_l^{\mathbb{J}}$ at time step L. Eqs. (8) and (9) constrain the quantity of attribute $a \in \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}$ at processing location $p \in \mathcal{P}$ under joint uncertainty scenario $s_l^{\mathbb{J}} \in \mathbb{S}_l^{\mathbb{J}}$ within the upper and lower production capacities while allowing for deviations $d^{+}_{a,p,L,s_l^{\mathbb{J}}}$ and $d^{-}_{a,p,L,s_l^{\mathbb{J}}}$ from such capacities, respectively. Eq. (10) ensures that mass flow balancing is conserved while solving the stochastic optimization model. The history of materials at different destinations and processing streams, along with other information mentioned in Section 2.2, is then fed to the DNN agent to generate both prior probabilities p_L and scalar evaluations v_L for nodes $x_{b,d,L}$. The second evaluation is the rollout simulation until the time step T using a tree policy P_π. The rollout policy P_π consists of, at each time step $t \in [L, T]$, (i) finding permissible blocks to extract using Eqs. (1) and (2), (ii) finding permissible block destinations using cut-off grade policies [36,37], (iii) combining the two to form permissible actions $x_{b,d,t}$ for state s_t, (iv) randomly selecting an action for state s_t among the set of permissible actions, and then (v) optimizing the processing stream utilization decisions with the stochastic mathematical programming model (Eqs. (7)–(11)). Eqs. (5) and (6) are then used to update the history of materials at destinations at time step T with the actions $x_{b,d,t}, \forall t \in [L, T]$ selected during the Monte Carlo tree search.

$$v_{a,d,T,s_l} - d^{+}_{a,d,T,s_l^{\mathbb{J}}} \leq U_{a,d,T}, \forall a \in \mathbb{P}_{\mathbb{M}}, d \in \mathcal{D}, s_l^{\mathbb{J}} \in \mathbb{S}_l^{\mathbb{J}} \quad (12)$$

$$v_{a,d,T,s_l} + d^{-}_{a,d,T,s_l^{\mathbb{J}}} \geq L_{a,d,T}, \forall a \in \mathbb{P}_{\mathbb{M}}, d \in \mathcal{D}, s_l^{\mathbb{J}} \in \mathbb{S}_l^{\mathbb{J}} \quad (13)$$

Eqs. (12) and (13) are used to calculate the deviations $d^{+}_{a,d,T,s_l^{\mathbb{J}}}$, $d^{-}_{a,d,T,s_l^{\mathbb{J}}}$ from upper and lower production capacities, respectively, at the destinations at time step T. Here, $U_{a,d,T}$ is $\max\left(\mathbb{S}'_{l,d}(T)\right)$ and $L_{a,d,T}$ is $\min\left(\mathbb{S}'_{l,d}(T)\right), \forall a \in \mathbb{P}_{\mathbb{M}}, d \in \mathcal{D}$. The stochastic mathematical programming model (Eqs. (7)–(11)) is used to compute $y_{a,d,p,T,s_l^{\mathbb{J}}}$ and deviations $d^{+}_{a,p,L,s_l^{\mathbb{J}}}$, $d^{-}_{a,p,L,s_l^{\mathbb{J}}}$ at time step T. Finally, the expected future reward r_T under joint uncertainty $\mathbb{S}_l^{\mathbb{J}}(T)$, for the rollout simulation is calculated as:

$$r_T = \underbrace{\frac{1}{|\mathbb{S}_l^{\mathbb{J}}(T)|} \sum_{s \in \mathbb{S}_l^{\mathbb{J}}(T)} \sum_{p \in \mathcal{P}} \sum_{a \in \mathbb{P}_{\mathbb{R}}} P_{a,p} \cdot v_{a,p,T,s} \cdot r_{a,p}}_{\text{Part I}}$$

$$- \underbrace{\frac{1}{|\mathbb{S}_l^{\mathbb{J}}(T)|} \sum_{s \in \mathbb{S}_l^{\mathbb{J}}(T)} \sum_{i \in \mathcal{P} \cup \mathcal{D} \cup M} \sum_{a \in \mathbb{P}_{\mathbb{M}}} C_{a,i} \cdot v_{a,i,T,s}}_{\text{Part II}}$$

$$- \underbrace{\frac{1}{|\mathbb{S}_l^{\mathbb{J}}(T)|} \sum_{s \in \mathbb{S}_l^{\mathbb{J}}(T)} \sum_{i \in \mathcal{P} \cup \mathcal{D}} \sum_{a \in \mathbb{P}_{\mathbb{D}} \cup \mathbb{P}_{\mathbb{M}}} \left(c^{+}_{a,i} \cdot d^{+}_{a,i,T,s} + c^{-}_{a,i} \cdot d^{-}_{a,i,T,s}\right)}_{\text{Part III}} \quad (14)$$

Part I of Eq. (14) represents the profit from selling all the products, Part II includes all the costs incurred to generate the products, such as mining, crushing, stockpiling, and processing costs, and Part III represents the penalties for deviating from the different production limits.

2.2.1.4. Backup. The last phase of the tree search is the backup phase (Fig. 4(d)), where first the visit count of all nodes s_t, visited at each time step $t \leq L$ until the leaf node was reached in the selection phase, is increased by 1 using Eq. (15) shown below:

$$N\left(s_t, x_{b,d,t}\right) = N\left(s_t, x_{b,d,t}\right) + 1, \quad \forall t \leq L \quad (15)$$

The total action value of each node s_t visited at each time step $t \leq L$ is calculated as:

$$W\left(s_t, x_{b,d,t}\right) = W\left(s_t, x_{b,d,t}\right) + r_t, \quad \forall t \leq L \quad (16)$$

The mean action value for each node s_t visited at each time step $t \leq L$ is updated by mixing the scalar evaluations from DNN agent with a factor γ, and the mean action value stored in the search tree with a factor $1 - \gamma$, as shown below:

$$Q\left(s_t, x_{b,d,t}\right) = \frac{W\left(s_t, x_{b,d,t}\right)}{N\left(s_t, x_{b,d,t}\right)}(1 - \gamma) + \gamma \cdot v_t, \quad \forall t \leq L \quad (17)$$

2.2.1.5. Play. The MCTS search (Sections 2.2.1.1–2.2.1.4) is repeated for N^{Train}_{MCTS} times and finally an action $x_{b,d,t}$ at time step t is selected proportion to the visit count of all actions, i.e. $\pi_t(x_{b,d,t}\,|s_t) \propto N(s_t, x_{b,d,t}) / \sum_{b' \in \mathcal{B}, d' \in \{\mathcal{D}(i), \forall i \in b'\}} N\left(s_t, x_{b',d',t}\right)$. The selected action is used in Eqs. (5)–(11) to update the state to s_{t+1}. The selected action becomes the new root for the next round of self-play, and the process continues until the terminal time step T is reached.

2.2.2. DNN agent training

The DNN agent f_{θ_t} is initialized at time $t = 1$ with random weights θ_1. At each subsequent time step $t \leq T$, an MCTS search $\alpha_{\theta_{t-1}}(s_t)$ is executed using the DNN agent from the previous step $f_{\theta_{t-1}}$ and tree policy P_π to return both the search probabilities π_t and search scalar evaluations z_t. The data for each time-step t is stored as (s_t, π_t, z_t) and used to locally train the DNN agent to generate new parameters f_{θ_t}, for N_L iterations. More specifically, the DNN parameter f_θ is adjusted by stochastic gradient descent on a loss function (l) that minimizes the cross-entropy loss between action probabilities p and search probabilities π, and the mean-squared error between scalar predicted v and search evaluations z as shown below:

$$l = \|z - v\|^2 - \pi^{\mathrm{T}} \log p + c\,\|\theta\|^2 \quad (18)$$

where, $c\,\|\theta\|^2$ is the L2 regularization with a penalty cost c and is added in the loss function to avoid overfitting. At time step $t = T$, all the stored training data prior to time step t is then used to train the DNN agent globally for N_G iterations to avoid overfitting to any specific data instances generated from any specific time step. The new DNN agent is then used for further rounds of self-play and training.

2.3. Responding to incoming new information

The self-play reinforcement learning algorithm (Section 2.2) generates a DNN agent that can adapt the short-term production schedule of a mining complex with incoming new information.

The new information collected during mining operations is first used to generate updated supply $\mathbb{S}_{U,a,m}$ and equipment performance uncertainties $\mathbb{S}'_{U,e}$. The updated uncertainties are then fed to the trained DNN agent to adapt the short-term production schedule, which is then used to generate the updated production forecasts. In parallel, a new DNN agent is trained with the updated uncertainties to further adapt the agent parameters. The algorithm is general and can be applied to different mining complexes.

3. Application at a copper mining complex

The proposed self-play reinforcement learning algorithm is applied at a copper mining complex. In the case study, the incoming new information about the supply of materials is the blasthole data measured during drilling at the mine, and the productivity data about the shovels, trucks, and crushers collected during the mine's operation. However, the algorithm is flexible enough to include different types of incoming new information related to the supply of materials and the performance of the different components of a mining complex. The copper mining complex is currently using the blasthole data to identify ore/waste boundaries in the blasted areas of the mines [38] and production data for dispatching and assignment related equipment decisions [15].

3.1. Overview of the mining complex

The copper mining complex consists of two mines, mine A and mine B. The materials extracted with the multiple shovels at the two mines are transported via trucks to five different crushers, a waste dump, and a sulphide leach pad, represented by $\mathcal{D}$ as shown in Fig. 5. The materials from the crushers are then transported to three different processing mills and an oxide leach pad via conveyor belts. The processing mills produce copper (Cu) concentrate as a primary product, and gold (Au), silver (Ag), and molybdenum (Mo) concentrate as secondary/by-products. The leach pads supply materials to a copper cathode plant that generates copper plate products. The processing mills and the cathode plant are represented by $\mathcal{P}$. The products (concentrates and plates) are then transported and sold to different customers.

The proposed algorithm is used to adapt the weekly ($N_{week} = 13$) short-term production schedule of the copper mining complex within a given quarterly production schedule which, for this case study, consists of 3600 and 1200 mining blocks from each of the two mines, respectively. The mining blocks have properties such as copper soluble (CuS), copper total (CuT), gold (Au), silver (Ag), and Molybdenum (Mo), which generate revenue, arsenic (As) which is deleterious, and the mass of the block. All the components of the mining complex have limits on their production capacity. Additionally, the processing mill has a limit on the amount of arsenic in the product. Table 1 shows the material classification criteria, permissible destinations, and cut-off grade policies used at the copper mining complex. The economic and operational parameters used at the copper mining complex are listed in Table 2. The economic parameters are scaled for confidentiality purposes. The production limits with the different components of the copper mining complex are listed in Table 3. The production limits are scaled for confidentiality purposes.

3.2. Parameters

The proposed algorithm was run on an Intel® i7-8700 machine with an 8-core processor and an NVIDIA GeForce GTX 1050 GPU. The algorithm uses the Tensorflow Adam optimizer with default settings [39] to train the DNN agent for approximately 2 days. The neighbourhood of blocks used as an input was set to 81 blocks for each shovel. The inputs to the DNN agent are normalized between 0 and 1. The scalar evaluations of the MCTS are also normalized between 0 and 1 for them to be of the same magnitude as the search probabilities. The number of next permissible actions is fixed to be 256. The missing actions are filled in by duplicating the existing actions until 256 is reached. If there are more than 256 permissible actions, then only the first 256 actions are considered. The training process in this case study was executed with 2 local and 20 global iterations. Over the course of training, 1.9 million different weekly production schedules were generated to train the DNN agent. 500 (N^{Train}_{MCTS}) rollout simulations were performed for selecting an action in the MCTS, which corresponds to approximately 30 s of think time. The mixing parameter, L2 regularization cost, and MCTS selection probability parameter are set to 0.25, 0.01, and 2, respectively. 10 stochastic simulations for each of the two mines and all their equipment were used to train the DNN agent.

3.3. Results

The results of the proposed self-play reinforcement learning algorithm for adapting the short-term production schedule with incoming new information are presented in this section. Section 3.3.1 shows the result of updating the supply and equipment performance uncertainties with incoming new information. The new information considered for updating supply uncertainty is generated from blasthole drilling during the mine's operation. The equipment performance uncertainty is updated with its productivity data collected during the mine's operations. Section 3.3.2 details the result of adapting the 13-week short-term production schedule with the trained DNN agent. Three different sets of results are reported and compared in this section. The first set of results presents the performance of the existing short-term production schedule under 100 joint uncertainty scenarios (10 supply and 10 equipment performance uncertainties). The second set of results includes realizing the existing short-term production schedule under the updated uncertainties. In this case, the block extraction sequence and destination decisions are fixed form the existing short-term production schedule, and only the processing stream utilization decisions are reoptimized using Eqs. (7)–(11). The third and final set of results includes allowing the DNN agent to adapt the short-term production schedule based on the updated uncertainties. The three sets of results are reported using the 10th, 50th, and 90th percentile risk profiles (P10, P50, and P90, respectively) of the different performance indicators over a set of 100 joint uncertainty scenarios (separate from the ones used to train the DNN agent). The results also present a comparison between the existing and adapted short-term production schedule to highlight the capabilities of the DNN agent to generate operationally feasible short-term production schedules.

3.3.1. Updated supply and equipment performance uncertainties

The incoming new information about the properties of materials $l^{NI}_a, \forall a \in \mathbb{P}_{\mathbb{R}} \cup \mathbb{P}_{\mathbb{D}}$ is used to generate the updated supply uncertainty $\mathbb{S}_{U,a,m}$ with an extended ensemble Kalman filter (EnKF) method that accounts for the multivariate spatial correlation of different properties. The details of the method can be found in Kumar et al. [33].

Supply uncertainty about six correlated properties, namely CuS, CuT, As, Au, Ag, and Mo, is updated with the new blasthole information (3437 blasthole data) collected during the mine's operation. Fig. 6(a), (b), and (c) shows one of the initial stochastic simulations, new blasthole data and one of the updated stochastic simulations of CuT and As property, respectively, at bench 1 of mine A. The extended EnKF method updates local characteristics

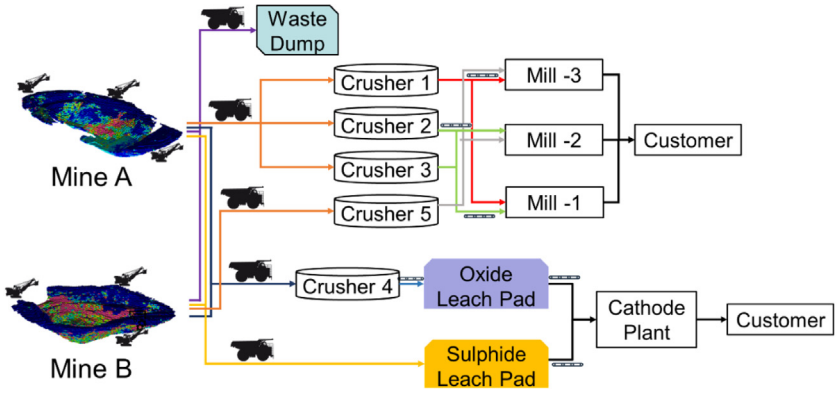

Fig. 5. Copper mining complex.

Table 1
Material classification criteria and cut-off grade policies for copper mining complex.

Materials classification	Materials classification criteria	Possible destinations	Cut-off grade destination policies	Cut-off grade destination
High-grade Sulphide	CuS/CuT $\leq$ 0.2	Processing mill, sulphide leach pad, and waste dump	CuT $\geq$ 0.6 0.3 $\leq$ CuT $<$ 0.6 CuT $<$ 0.3	Processing mill Sulphide leach pad Waste dump
Low-grade Sulphide	0.2 $<$ CuS/CuT $\leq$ 0.5	Processing mill, sulphide leach pad, and waste dump	CuT $>$ 0.3 CuT $\leq$ 0.3	Sulphide leach pad Waste dump
Oxide	CuS/CuT $\geq$ 0.5	Oxide leach pad and waste dump	CuS $\geq$ 0.2 CuS $<$ 0.2	Oxide leach pad Waste dump

Table 2
Operational and economic parameters used at the copper mining complex.

Attribute	Value
Number of mining blocks	Mine A: 3600; Mine B: 1200
Production scheduling horizon	13 weeks
Slope angle (Mine A and B)	45, 45
Radius (Mine A and B)	10 mining blocks
Recovery of copper	Oxide leach pad: 65%; Sulphide leach pad: 27%; Processing mills: 80.4%, 80% and 82.6%
Recovery of gold, silver, and molybdenum	0.25
Selling cost—processing mills, oxide leach pad, and sulphide leach pad	571, 551, and 551 $/tonne
Selling price—copper, gold, silver, and molybdenum concentrate, copper plate	5511, 35.2 $\times$ 10^6 ,4.9 $\times$ 10^5, and 1.3 $\times$ 10^4 $/ tonne
Processing cost—-processing mills, oxide leach pad, and, sulphide leach pad	5.79, 5.81, and 1.84 $/tonne
Crushing cost (Crusher 1, 2, 3, 4, and, 5)	0.58 $/tonne
Mining cost (Depending on depth)	Mine A: 0.4 - 1.27; Mine B: 0.52 - 1.09 ($/tonne)
Penalty cost: arsenic grade limit, capacity limit at crusher, oxide leach pad, sulphide leach pad, and, processing mills	5 $/ PPM, 1$/tonne, 1$/tonne, 1$/tonne, and 2$/tonne

Table 3
Production limits with different components of the copper mining complex.

Attribute	Value (Weekly)
Crusher 1, 2, 3, 4, and 5 production capacity limit (Tonnes)	Stochastic
Mill 1, 2, and, 3 capacity limit (Tonnes)	28, 33.5, and 38.5%
Oxide and sulphide leach pad capacity limit (Tonnes)	18.2 and 81.8%
Arsenic grade limit for processing mills (PPM)	1%

of materials at mine A with the new blasthole data. The concentration and spatial distribution of As have changed significantly in the updated simulations, as seen in Fig. 6(c).

The incoming new information about the productivity of different components $l_e^{NI}, \forall e \in \mathbb{E}$ is used to update the initial equipment performance uncertainty $s'_{l,e} \in \mathbb{S}'_{l,e}(T), \forall e \in \mathbb{E}$ by first computing the empirical cumulative distribution function (ECDF) with both the historical data eP_e^I and incoming new information l_e^{NI}. The ECDF is then sampled to generate the updated, U, equipment performance uncertainty $s'_{U,e} \in \mathbb{S}'_{Ue}(T)$ of the different components $e \in \mathbb{E}$. Fig. 7(a) and (b) shows the initial and updated simulations, respectively, of the production capabilities of crusher 5. The production capabilities of crusher 5 in the initial and updated simulations are different but still respect the data. Updating the supply and equipment performance uncertainties with incoming new information in this case study takes about five minutes. The updated supply $\mathbb{S}_{U,a,m}$, and equipment performance uncertainties $\mathbb{S}'_{U,e}(T)$ are fed to the DNN agent that responds to

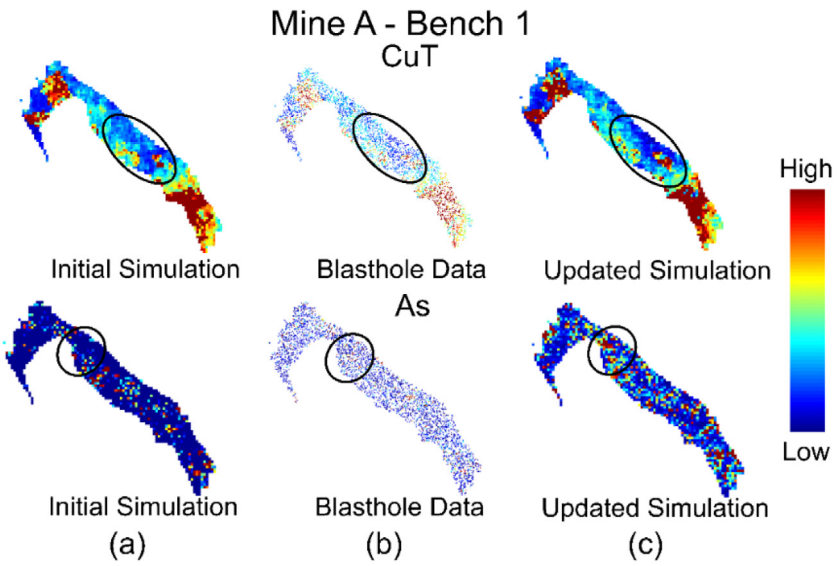

Fig. 6. (a) One of the initial stochastic simulations for CuT and As properties, (b) incoming blasthole data collected during operations about CuT and As properties, and (c) one of the updated stochastic simulations for CuT and As properties for bench 1 at mine A.

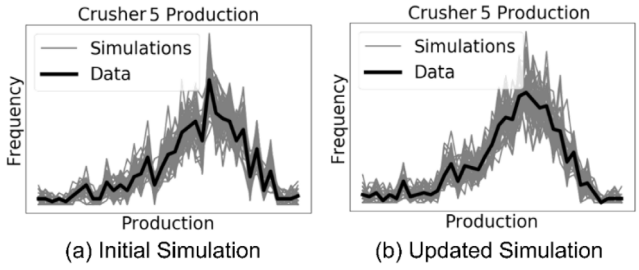

Fig. 7. (a) Initial and (b) updated simulations about crusher 5 production capacity.

the updated uncertainties by adapting the short-term production schedule (see Section 2.3). The results of the adapted short-term production schedule are discussed next.

3.3.2. Adapted short-term production schedule

The results in this section are scaled for confidentiality reasons, with the forecast of an initial production schedule under the initial uncertainties being 100%. The results indicated by (b) and (c) in Figs. 10–14 show the value of updating the uncertainties and the added value of adapting the short-term production schedule (the ability of the DNN agent to respond to incoming new information), respectively. Although a mining operation will use this framework to adapt and learn continuously, for simplicity and fair comparison, no Monte Carlo tree searches or training are performed over the updated uncertainties. This algorithm takes only about two minutes to adapt the 13-week short-term production schedule in this case study. The additional results of the case study are shown in Appendix B.

3.3.2.1. Extraction sequence and materials destination. Fig. 8 and Fig. 9 show the block extraction sequence and destination decisions, respectively, for bench 1 at mines A and B for the (a) initial and (b) adapted short-term production schedules. The initial and adapted short-term production schedules respect the permissible block extraction and destination constraints. The adapted block extraction and destination decisions are very different from the initial ones. The DNN agent is efficient at responding to the updated uncertainties by adapting the extraction sequence to better blend the material and respect the production limits of the different components. Additionally, the destination of the blocks is adapted by the DNN agent to better utilize the processing capabilities and produce a higher quantity of primary products. The reasons for the major differences in the initial and adapted short-term production schedules are due to:

a. The ability of the proposed algorithm to capitalize on the synergies between the different components of the mining complex to simultaneously adapt all relevant short-term production scheduling decisions.
b. The ability of the proposed algorithm to account for multiple sources of uncertainty related to the supply of material and production capabilities of different components of the mining complex.
c. The ability of the continuous updating framework to allow the DNN agent to better observe the updated state of the mining complex, which includes uncertainty about revenue-generating and deleterious properties of materials and the production capabilities of its components.

3.3.2.2. Ore production forecasts. The forecasts for the initial and adapted short-term production schedules for the different production limits are shown in this section.

Fig. 10(a) shows the performance of the initial short-term production schedule for the production capacity limit of mill 1. The production schedule respects the capacity limits of mill 1 with some weeks of lower utilization. Fig. 10(b) shows the risk profile of the initial production schedule over the updated uncertainties and presents a lower utilization of the capacity of mill 1. Fig. 10(c) shows the performance of the adapted short-term production schedule generated by the DNN agent. The capacity of mill 1 is better utilized in the adapted production schedule. Fig. 11(a), (b), and (c) show the forecasts for the initial short-term production schedule, its risk profile over the updated uncertainties, and the adapted schedule for the arsenic quality limit for mill 1, respectively. The arsenic quality limit is well respected until week 8 in the initial schedule (Fig. 11(a)); however, the risk of the initial schedule over the updated uncertainties (Fig. 11(b)) shows that this limit will be violated in the early weeks (because of a high arsenic concentration in the materials in mine A as seen from Fig. 6(c)). The DNN agent adapts the short-term schedule to blend materials from multiple mines, and the violations are minimal in the adapted schedule (Fig. 11(c)).

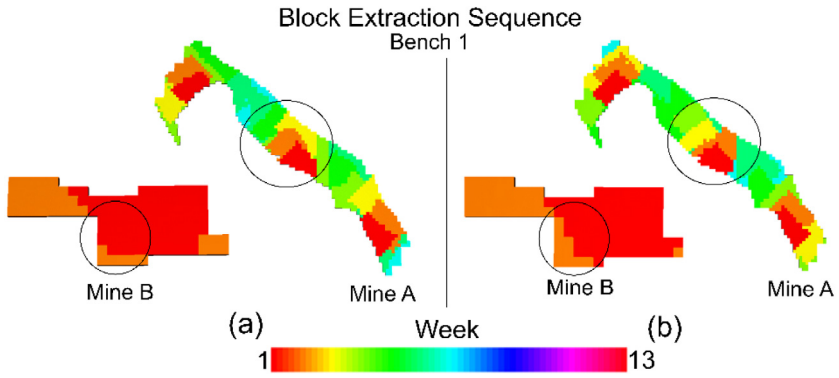

Fig. 8. Block extraction sequence in (a) the initial short-term production schedule compared to (b) the adapted short-term production schedule for bench 1.

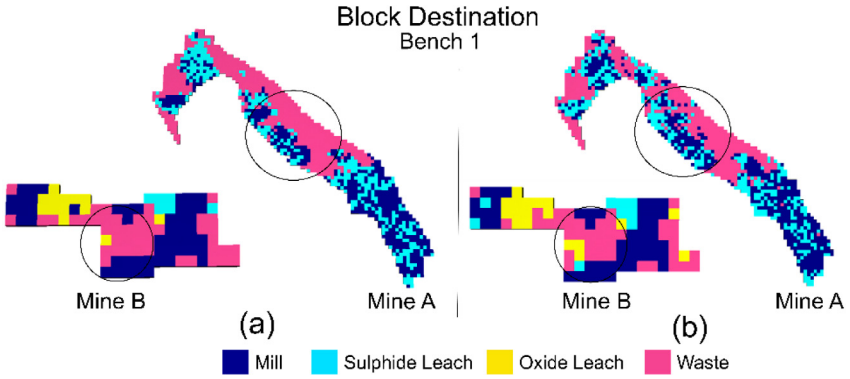

Fig. 9. Block destination decisions in (a) the initial short-term production schedule compared to (b) the adapted short-term production schedule for bench 1.

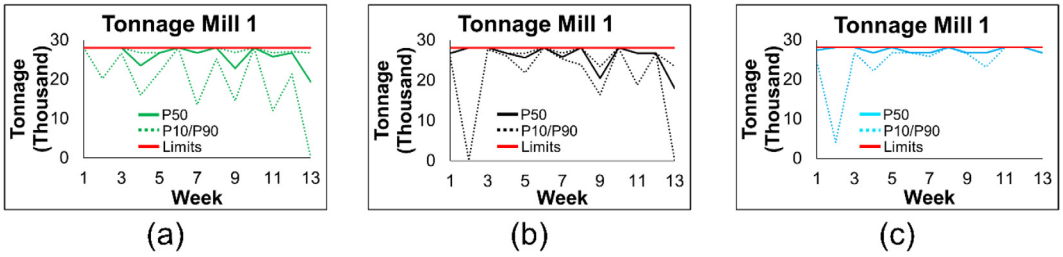

Fig. 10. Mill 1 production limit forecasts for (a) initial short-term production schedule, (b) risk profile of initial schedule over the updated uncertainties, and (c) adapted schedule.

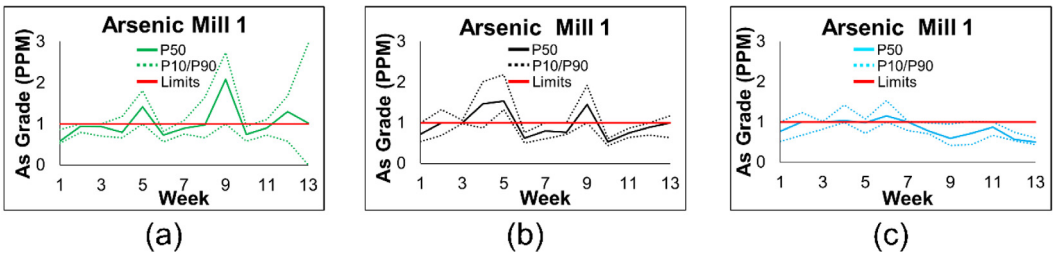

Fig. 11. Mill 1 arsenic limit forecasts for (a) initial short-term production schedule, (b) risk profile of initial schedule over the updated uncertainties, and (c) adapted schedule.

3.3.2.3. Metal production forecasts. Fig. 12(a), (b), and (c) show the forecasts for copper concentrate production for the initial short-term production schedule, the risk of the initial schedule over the updated uncertainties, and the adapted schedule, respectively. The initial schedule realized over the updated uncertainties (Fig. 12(b)) shows an increase of 2% in recovered copper concentrate production. However, the DNN agent adapts the short-term schedule to increase the copper concentrate by 14%. The adapted schedule better blends the materials to better utilize the processing mill capacities and results in a higher copper concentrate production.

Fig. 13 shows the forecasts for gold concentrate production. The initial short-term production schedule realized over the updated uncertainties shows an 11% increase in recovered gold

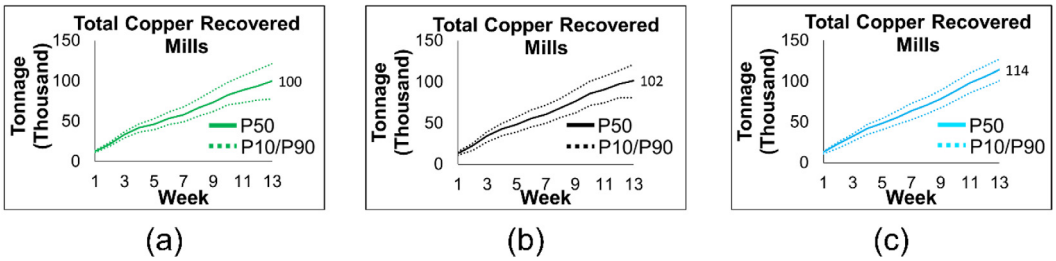

Fig. 12. Copper concentrate forecasts for (a) initial short-term production schedule, (b) risk profile of initial schedule over the updated uncertainties, and (c) adapted schedule.

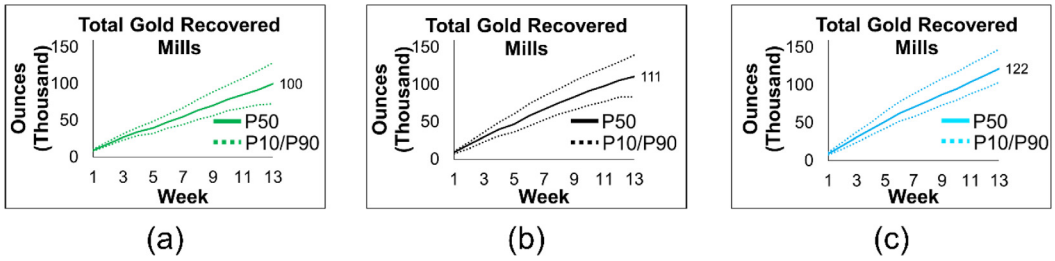

Fig. 13. Gold concentrate forecasts for (a) initial short-term production schedule, (b) risk profile of initial schedule over the updated uncertainties, and (c) adapted schedule.

production (Fig. 13(b)). The adapted schedule by the DNN agent increases gold production by 22% (Fig. 13(c)). The adapted schedule processes less material with the leach pads to better blend the materials to meet the arsenic quality limits of the processing mills (Fig. 11(c)), to better utilize the processing mill capacities (Fig. 10(c)), and to generate a higher quantity of primary copper concentrate and secondary gold, silver and molybdenum products (Figs. 12(c) and 13(c)).

3.3.2.4. Cash flows forecasts. Fig. 14(a), (b), and (c) show the cumulative cash flow forecasts of the initial short-term production schedule, the initial schedule realized over the updated uncertainties, and the adapted schedule, respectively.

The initial short-term production schedule shows an increase of 5% in cash flow when realized over the updated uncertainties, which shows the value of updating the uncertainties with incoming new information. However, the DNN agent adapts the short-term production schedule to generate a 12% increase in cash flow, i.e. an additional 7% of added value in adapting the short-term schedule. The added cash flow value in the adapted schedule is generated by a better extraction sequence and destination decisions, and improved blending strategies to maximize the utilization of the processing stream capacities.

4. Conclusions

This paper proposes a new self-play reinforcement learning algorithm that combines a Monte Carlo tree search with a deep neural network agent to adapt the short-term production schedule of a mining complex with incoming new information. The deep neural network agent evaluates the short-term production scheduling decisions and, in parallel, uses the evaluations in a Monte Carlo tree search to gather self-play experiences by performing random rollouts. The gathered experiences are then used to train the deep neural network agent, which improves the strength of the tree search and results in stronger self-plays to generate better experiences. First, the incoming new information is used in the extended EnKF algorithm proposed in Kumar et al. [33] and a Monte Carlo simulation algorithm proposed in this work to update the supply and equipment performance uncertainties. The updated uncertainties are then fed to the self-play reinforcement learning algorithm proposed in this work, to adapt all the relevant short-term production scheduling decisions (sequence of extraction, the destination of materials, and utilization of processing stream) in a mining complex simultaneously. An application of the proposed algorithm at a copper mining complex shows its exceptional performance in adapting the 13-week short-term production schedule with incoming new information. The risk profiles of realizing the initial 13-week short-term production schedule of the copper mining complex over the updated supply and equipment performance uncertainties showed an increase of 5% in cumulative cash flow, and an increase of 2%, 11%, 23%, and 32% in copper concentrate, gold, silver, and molybdenum production, respectively. It also showed a large violation of the arsenic content limit and a lowered utilization of processing capacities. The proposed self-play reinforcement learning algorithm adapted the short-term production schedule to increase the cumulative cash flow by 12%, and the copper concentrate, gold, silver, and molybdenum production by 14%, 22%, 43%, and 61%, respectively. The results for silver and molybdenum production are presented in Appendix B. In addition, the adapted short-term production schedule makes better use of the processing mill capacities and shows a minimal violation of the arsenic quality limit of the processing mills by finding better blending strategies, which result in a 2% reduced copper cathode production, as shown in Appendix B. The process of adapting the 13-week short-term production schedule of the copper mining complex takes two minutes in the case study presented. The algorithm presented in this work can be applied to any industrial environment that has the necessary infrastructure to capture data with sensors, transmit data via a local network, store data in database platforms, perform necessary computations using GPU cloud computing, and has access to a platform to visualize the results. The proposed algorithm does not adapt the fleet assignment and allocation decisions. In addition, it uses a tree policy to perform the MCTS search simulations and, most importantly, the adaptation is performed within the long-term production schedule. Future research can focus on integrating fleet assignment and allocation decisions, adapting short-and-long-term production schedules simultaneously, integrating more sources of incoming new information, and modifying the algorithm to use convolution neural network agents without a tree policy, like the AlphaGoZero algorithm.

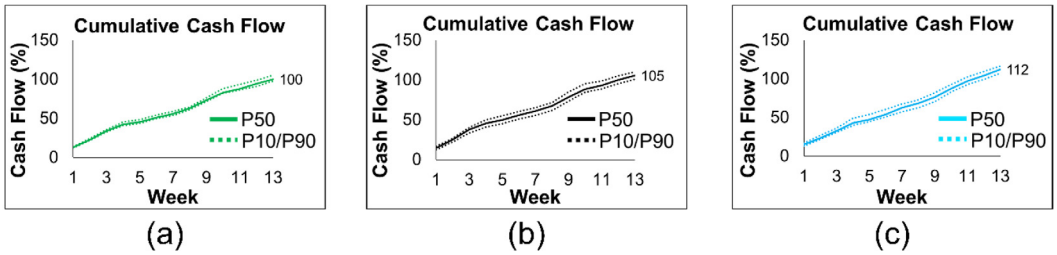

Fig. 14. Cumulative cash flow forecasts for (a) initial short-term production schedule, (b) risk profile of initial schedule over the updated uncertainties, and (c) adapted schedule.

Table A.1
Sets and indices used in the proposed algorithm.

Parameters	Definition
M	Set of mines, $m \in M$
I	Initial
U	Updated
N_{week}	Number of weeks in the quarterly short-term production schedule
$\mathbb{P}_{\mathbb{R}}$	Set of revenue-generating elements attribute
$\mathbb{P}_{\mathbb{D}}$	Set of deleterious elements attribute
$\mathbb{P}_{\mathbb{M}}$	Rock mass attribute
$\mathbb{Z}_m(x)$	Set of mining blocks in mine m located at x, $z \in \mathbb{Z}_m(x)$
$dH^I_{a,m}$	Initial drill hole information for attribute a in mine m
$\mathbb{S}_{I,a,m}$	Set of initial stochastic simulations for attribute a in mine m, $s_{I,a,m} \in \mathbb{S}_{I,a,m}$
$\mathbb{Z}^{I,s}_{a,m}(x)$	Property of a set of blocks located at x in mine m for attribute a in initial stochastic simulation $s_{I,a,m}$
$\mathbb{S}^{\mathbb{J}}_I$	Set of initial joint uncertainty scenarios, $s^{\mathbb{J}}_I \in \mathbb{S}^{\mathbb{J}}_I$, $\mathbb{S}^{\mathbb{J}}_I = \mathbb{S}_{I,a,m} \cup \mathbb{S}'_{I,e}(T), \forall T \in [1, N_{week}]$
$\mathcal{S}_m$	Set of shovels located in mine m, $s_i \in \mathcal{S}_m$
$\mathcal{T}_m$	Set of trucks used in mine m
$\mathcal{D}$	Set of destinations in the mining complex
$\mathcal{P}$	Set of processing streams in the mining complex
$\mathbb{S}'_{I,e}(T)$	Set of initial stochastic simulations for component e in period T, $s'_{I,e} \in \mathbb{S}'_{I,e}(T), \forall T \in [1, N_{week}]$
eP^I_e	Initial production information (historical information) for component e
$\mathbb{P}_{\mathbb{P},e}(T)$	Production capacity for component e in period T, $\forall e \in \mathbb{E}$, $\mathbb{E} = \{\mathcal{T}, \mathcal{S}, \mathcal{D}, \mathcal{P}\}, \forall T \in [1, N_{week}]$
$\mathbb{Z}_m(s_i)$	Set of permissible blocks that can be extracted from mine m with the shovel s_i
B	Set consisting of sets where each set has one element from each set in $\mathbb{Z}_m(s_i)$
$D(z)$	Set of possible destinations for each mining block z
$\mathbb{S}'_{Ue}(T)$	Set of updated stochastic simulations for e in period T, $s'_{U,e} \in \mathbb{S}'_{Ue}(T), \forall T \in [1, N_{week}]$
I^{NI}_e	Real-time new information about the performance of component e
I^{NI}_a	Real-time new information about the supply of material attribute a
$\mathbb{S}_{U,a,m}$	Set of updated stochastic simulations for attribute a in mine m, $s_{U,a,m} \in \mathbb{S}_{U,a,m}$
$V(i)$	Set of blocks that are the vertical predecessor of block i
b'_V	Set of all the blocks that overlie all the blocks in set $b, \forall b \in B$
$H(i, r)$	Set of blocks that are horizontal successor (surrounding blocks) of block i within a radius r
b'_H	Set consisting of k sets, with each set k consist of one block that surrounds a block i in set b within a radius r, $\forall b \in B$
v_{a,d,t,s_I}	Quantity of the attribute a at destination d at time step t under stochastic simulation $s_{I,a,m}$, $\forall t \in [1, T]$
$v_{a,p,t,s^{\mathbb{J}}_I}$	Quantity of the attribute a at processing stream p at time step t under joint uncertainty scenario $s^{\mathbb{J}}_I$, $\forall t \in [1, T]$
$L_{a,j,T}$	Lower limit for attribute a at location j in period $T, \forall T \in [1, N_{week}]$
$U_{a,j,T}$	Upper limit for attribute a at location j in period $T, \forall T \in [1, N_{week}]$

Table A.2
Variables used in the proposed algorithm.

Variable	Definition
$x_{b,d,t} \in \{0, 1\}$	Defines if a set of blocks b is extracted from a set of shovels and sent to a set of destinations d at time step $t, \forall t \in [1, T]$
$y_{a,d,p,t,s^{\mathbb{J}}_I} \in [0, 1]$	Amount of attribute a sent from destination d to processing stream p at time step t under joint uncertainty scenario $s^{\mathbb{J}}_I$, , $\forall t \in [1, T]$
f_θ	Deep neural network (DNN) agent with parameters θ
s_t	State of the mining complex at time $t, \forall t \in [1, T]$
p_t, v_t	Vector of selection probabilities and scalar evaluation for all the permissible actions $x_{b,d,t}$ in the state s_t, $(p_t, v_t) = f_\theta(s_t), \forall t \in [1, T]$
α_θ	Monte Carlo tree search (MCTS)
π_t	Selection probabilities at time step t output from Monte Carlo tree search $\alpha_\theta, \forall t \in [1, T]$
z_t	Scalar evaluation at time step t from Monte Carlo tree search $\alpha_\theta, \forall t \in [1, T]$
$P(s_t, x_{b,d,t})$	The prior probability for state–action pair $(s_t, x_{b,d,t})$
$W(s_t, x_{b,d,t})$	Total action value for state–action pair $(s_t, x_{b,d,t})$
$Q(s_t, x_{b,d,t})$	Mean action value for state–action pair $(s_t, x_{b,d,t})$
$N(s_t, x_{b,d,t})$	Visit count for state–action pair $(s_t, x_{b,d,t})$
$d^+_{a,j,t,s^{\mathbb{J}}_I}$	Continuous variable for deviation from the upper limit $U_{a,j,T}$ at time step t for attribute a at location j under joint uncertainty scenario $s^{\mathbb{J}}_I, \forall t \in [1, T]$
$d^-_{a,j,t,s^{\mathbb{J}}_I}$	Continuous variable for deviation from the lower limit $L_{a,j,T}$ at time step t for attribute a at location j under joint uncertainty scenario $s^{\mathbb{J}}_I, \forall t \in [1, T]$
r_t	Total cumulative future expected reward from time step $t, \forall t \in [1, T]$

Table A.3
Constants used in the proposed algorithm.

Constants	Definition
$C_{a,d}$	Cost incurred for material attribute a at destination d
$r_{a,p}$	Recovery factor for attribute a at processing location p
$C_{a,p}$	Cost of processing material attribute a at processing location p
$P_{a,p}$	Price of selling material attribute a at processing location p
P_{π}	Sub-optimal tree policy
c_{puct}	A factor that determines the level of exploration in the selection phase of MCTS
γ	Mixing parameter for neural network policy reward and MCTS reward
c	L2 regularization factor with neural network
N_L	Number of local epochs for training DNN agent
N_G	Number of global epochs for training DNN agent
N_{MCTS}^{Train}	Number of MCTS simulations in the training phase
N_{MCTS}^{Update}	Number of MCTS simulations in adapting phase
A	Vertical Slope angle
r	Horizontal radius

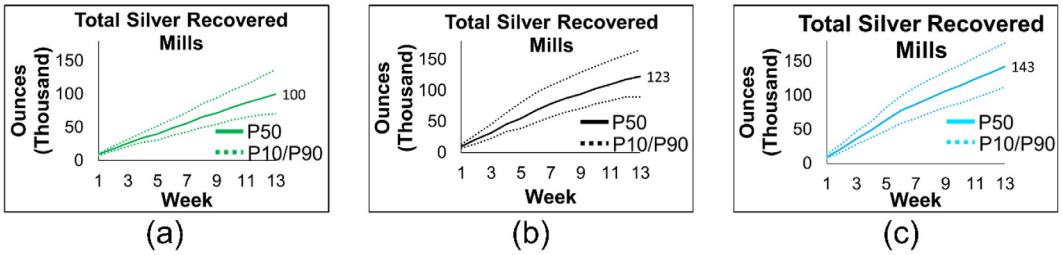

Fig. B.1. Silver production forecasts for (a) initial short-term production schedule, (b) risk profile of initial schedule over the updated uncertainties, and (c) adapted schedule.

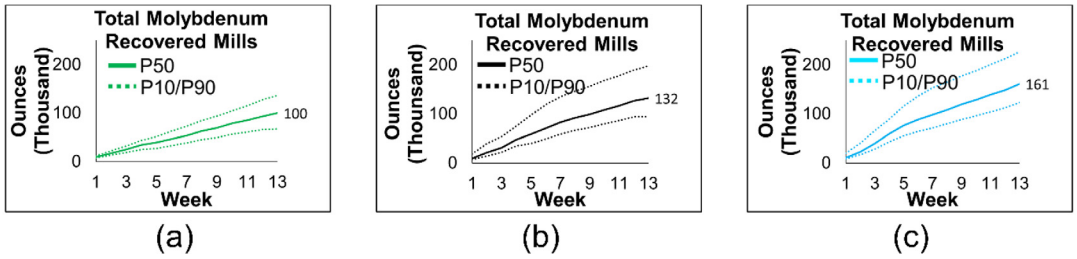

Fig. B.2. Molybdenum production forecasts for (a) initial short-term production schedule, (b) risk profile of initial schedule over the updated uncertainties, and (c) adapted schedule.

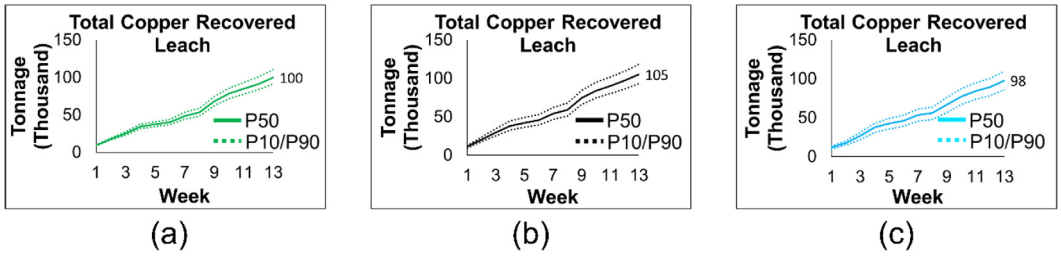

Fig. B.3. Copper plate production forecasts for (a) initial short-term production schedule, (b) risk profile of initial schedule over the updated uncertainties, and (c) adapted schedule.

CRediT authorship contribution statement

Ashish Kumar: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Writing - review and editing. **Roussos Dimitrakopoulos:** Supervision, Writing - review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work in this paper was funded by the National Sciences and Engineering Research Council (NSERC) of Canada CRD Grant

500414-16 and NSERC Discovery Grant 239019, the industry consortium members of McGill University's COSMO Stochastic Mine Planning Laboratory (AngloGold Ashanti, Barrick Gold, BHP, De Beers, IAMGOLD, Kinross Gold, Newmont Corporation, and Vale); and the Canada Research Chairs Program.

Appendix A

See Tables A.1–A.3.

Appendix B

See Figs. B.1–B.3.

Appendix C. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2021.107644.

References

[1] M.E.V. Matamoros, R. Dimitrakopoulos, Stochastic short-term mine production schedule accounting for fleet allocation, operational considerations and blending restrictions, European J. Oper. Res. 255 (2016) 911–921, https://doi.org/10.1016/j.ejor.2016.05.050.

[2] M. Quigley, R. Dimitrakopoulos, Incorporating geological and equipment performance uncertainty while optimizing short-term mine production schedules, Int. J. Min. Reclam. Environ. (2019) 1–22, https://doi.org/10.1080/17480930.2019.1658923.

[3] R. Goodfellow, R. Dimitrakopoulos, Global optimization of open pit mining complexes with uncertainty, Appl. Soft Comput. 40 (2016) 292–304, https://doi.org/10.1016/j.asoc.2015.11.038.

[4] L. Montiel, R. Dimitrakopoulos, Optimizing mining complexes with multiple processing and transportation alternatives: An uncertainty-based approach, European J. Oper. Res. 247 (2015) 166–178, https://doi.org/10.1016/j.ejor.2015.05.002.

[5] N.L. Mai, E. Topal, O. Erten, B. Sommerville, A new risk-based optimisation method for the iron ore production scheduling using stochastic integer programming, Resour. Policy. 62 (2019) 571–579, https://doi.org/10.1016/j.resourpol.2018.11.004.

[6] A. Paithankar, S. Chatterjee, R. Goodfellow, M.W.A. Asad, Simultaneous stochastic optimization of production sequence and dynamic cut-off grades in an open pit mining operation, Resour. Policy. 66 (2020) 101634, https://doi.org/10.1016/j.resourpol.2020.101634.

[7] A. Paithankar, S. Chatterjee, Open pit mine production schedule optimization using a hybrid of maximum-flow and genetic algorithms, Appl. Soft Comput. J. 81 (2019) 105507, https://doi.org/10.1016/j.asoc.2019.105507.

[8] P. Goovaerts, Geostatistics for Natural Resources Evaluation, Oxford University Press Inc., New York, 1997.

[9] A. Boucher, R. Dimitrakopoulos, Block simulation of multiple correlated variables, Math. Geosci. 41 (2009) 215–237, https://doi.org/10.1007/s11004-008-9178-0.

[10] M. Godoy, The Effective Management of Geological Risk in Long-Term Production Scheduling of Open Pit Mines, The University of Queensland, 2002.

[11] N. Remy, A. Boucher, J. Wu, Applied Geostatistics with SGeMS: A User's Guide, Cambridge University Press, 2009, https://doi.org/10.1017/CBO9781139150019.

[12] L. Rosa, David, W. Valery, M. Wortley, T. Ozkocak, M. Pike, The use of radio frequency ID tags to track ore in mining operations, in: Proc. 33rd Appl. Comput. Oper. Res. Miner. Ind., 2007, pp. 601–606.

[13] P. Chaowasakoo, C. Leelasukseree, W. Wongsurawat, Introducing GPS in fleet management of a mine: Impact on hauling cycle time and hauling capacity, Int. J. Technol. Intell. Plan. 10 (2014) 49–66, https://doi.org/10.1504/IJTIP.2014.066711.

[14] W.G. Koellner, G.M. Brown, J. Rodríguez, J. Pontt, P. Cortés, H. Miranda, Recent advances in mining haul trucks, IEEE Trans. Ind. Electron. 51 (2004) 321–329, https://doi.org/10.1109/TIE.2004.825263.

[15] H. Kargupta, K. Srakar, M. Gilligan, MineFleet®: An overview of a widely adopted distributed vehicle performance data mining system, in: Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2010, pp. 37–46, https://doi.org/10.1145/1835804.1835812.

[16] M. Dalm, M.W.N. Buxton, F.J.A. van Ruitenbeek, Ore–waste discrimination in epithermal deposits using near-infrared to short-wavelength infrared (NIR-SWIR) hyperspectral imagery, Math. Geosci. 51 (2018) 1–27, https://doi.org/10.1007/s11004-018-9758-6.

[17] D.L. Death, A.P. Cunningham, L.J. Pollard, Multi-element and mineralogical analysis of mineral ores using laser induced breakdown spectroscopy and chemometric analysis, Spectrochim. Acta B. 64 (2009) 1048–1058, https://doi.org/10.1016/j.sab.2009.07.017.

[18] T.P.R. De Jong, Automatic sorting of minerals, in: IFAC Proc., 2004, pp. 441–446, https://doi.org/10.1016/s1474-6670(17)31064-9.

[19] M. Kern, L. Tusa, T. Leißner, K.G. van den Boogaart, J. Gutzmer, Optimal sensor selection for sensor-based sorting based on automated mineralogy data, J. Clean. Prod. 234 (2019) 1144–1152, https://doi.org/10.1016/j.jclepro.2019.06.259.

[20] A.F.H. Goetz, B. Curtiss, D.A. Shiley, Rapid gangue mineral concentration measurement over conveyors by NIR reflectance spectroscopy, Miner. Eng. 22 (2009) 490–499, https://doi.org/10.1016/j.mineng.2008.12.013.

[21] M. Dalm, M.W.N. Buxton, F.J.A. van Ruitenbeek, Discriminating ore and waste in a porphyry copper deposit using short-wavelength infrared (SWIR) hyperspectral imagery, Miner. Eng. 105 (2017) 10–18, https://doi.org/10.1016/j.mineng.2016.12.013.

[22] S. Iyakwari, H.J. Glass, G.K. Rollinson, P.B. Kowalczuk, Application of near infrared sensors to preconcentration of hydrothermally-formed copper ore, Miner. Eng. 85 (2016) 148–167, https://doi.org/10.1016/j.mineng.2015.10.020.

[23] A. Lamghari, Mine planning and oil field development: A survey and research potentials, Math. Geosci. 49 (2017) 395–437, https://doi.org/10.1007/s11004-017-9676-z.

[24] S. Aanonsen, G. Nævdal, D. Oliver, A. Reynolds, B. Vallès, The ensemble Kalman filter in reservoir engineering: A review, SPE J. 14 (2009) 393–412.

[25] D.S. Oliver, Y. Chen, Recent progress on reservoir history matching: A review, Comput. Geosci. 15 (2011) 185–221, https://doi.org/10.1007/s10596-010-9194-2.

[26] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res. Ocean. 99 (1994) 10143–10162.

[27] J. Benndorf, Making use of online production data: Sequential updating of mineral resource models, Math. Geosci. 47 (2015) 547–563, https://doi.org/10.1007/s11004-014-9561-y.

[28] T. Wambeke, D. Elder, A. Miller, J. Benndorf, R. Peattie, Real-time reconciliation of a geometallurgical model based on ball mill performance measurements – A pilot study at the Tropicana gold mine, Min. Technol. 127 (2018) 115–130, https://doi.org/10.1080/25726668.2018.1436957.

[29] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, second ed., MIT press, 2017.

[30] J. Benndorf, M.W.N. Buxton, Sensor-based real-time resource model reconciliation for improved mine production control: A conceptual framework, Min. Technol. 125 (2016) 54–64, https://doi.org/10.1080/14749009.2015.1107342.

[31] J. Hou, K. Zhou, X.S. Zhang, X.D. Kang, H. Xie, A review of closed-loop reservoir management, Pet. Sci. 12 (2015) 114–128, https://doi.org/10.1007/s12182-014-0005-6.

[32] C. Paduraru, R. Dimitrakopoulos, Responding to new information in a mining complex: Fast mechanisms using machine learning, Min. Technol. 128 (2019) 129–142, https://doi.org/10.1080/25726668.2019.1577596.

[33] A. Kumar, R. Dimitrakopoulos, M. Maulen, Adaptive self-learning mechanisms for updating short-term production decisions in an industrial mining complex, J. Intell. Manuf. (2020) 1–17, https://doi.org/10.1007/s10845-020-01562-5.

[34] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of Go with deep neural networks and tree search, Nature 529 (2016) 484–489, https://doi.org/10.1038/nature16961.

[35] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, D. Hassabis, Mastering the game of Go without human knowledge, Nature 550 (2017) 354–359, https://doi.org/10.1038/nature24270.

[36] K.F. Lane, The Economic Definition of Ore: Cut-Off Grades in Theory and Practice, Mining Journal Books Limited, London, 1988.

[37] J.-M. Rendu, An Introduction To Cut-Off Grade Estimation, Society for Mining, Metallurgy & Exploration, Englewood, Colarado, 2014.

[38] G. Verly, Grade gontrol classification of ore and waste: A critical review of estimation and simulation based procedures, Math. Geol. 37 (2005) 451–475, https://doi.org/10.1007/s11004-005-6660-9.

[39] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, 2015, ArXiv Prepr, pp. 1–15, https://arxiv.org/abs/1412.6980.

Ashish Kumar is a Senior Artificial Intelligence Specialist at Vale. He has a Ph.D. specializing in artificial intelligence in mining complexes from McGill University's COSMO Laboratory. He received his Bachelor of Mining Engineering in 2014 from NIT Rourkela, India.

Roussos Dimitrakopoulos is a Professor at McGill University, Montreal, and holds the Canada Research Chair (Tier I) in Sustainable Mineral Resource Development and Optimization under Uncertainty. He leads the COSMO Research Consortium of major global mining companies, namely, AngloGold Ashanti, Barrick Gold, BHP, DeBeers, IAMGOLD, Kinross Gold, Newmont Mining, and Vale.