



Integration of strategic open-pit mine planning into hierarchical artificial intelligence

by A. Quelopana^{1,2} and A. Navarra²

Affiliation:

¹Department of Systems and Computer Engineering, Universidad Católica del Norte, Antofagasta, Chile.

²Department of Mining and Materials Engineering, McGill University, Montreal, Canada.

Correspondence to:

A. Navarra

Email:

alessandro.navarra@mcgill.ca

Dates:

Received: 13 Sep. 2021

Revised: 27 Oct. 2021

Accepted: 17 Nov. 2021

Published: December 2021

How to cite:

Quelopana, A. and Navarra, A. 2021 Integration of strategic open-pit mine planning into hierarchical artificial intelligence. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 121, no. 12, pp. 643-652

DOI ID:

<http://dx.doi.org/10.17159/2411-9717/1367/2021>

ORCID:

A. Quelopana
<https://orcid.org/0000-0003-3002-4014>

A. Navarra

<https://orcid.org/0000-0001-6613-6750>

Synopsis

The mine production scheduling problem (MPSP) has been studied since the 1960s, and remains an active area of computational research. In extending the concepts of the MPSP, the automated mine may now be regarded as a hierarchical intelligent agent in which the bottom layer consists of distributed robotic equipment, while strategic functionality occupies the higher layers. Here we present a disambiguation of artificial intelligence, machine learning, computational optimization, and automation within the mining context. Specifically, the Q-learning algorithm has been adapted to generate the initial solutions for a high-performing strategic mine planning algorithm, originally developed by Lamghari, Dimitrakopoulos and Ferland, based on the variable neighbourhood descent (VND) metaheuristic. The hierarchical intelligent agent is presented as an integrative conceptual platform, defining the interaction between our new Q-learning adaptation and Lamghari's VND, and potentially other hierarchically controlled components of an artificially intelligent mine, having various degrees of automation. Sample computations involving Q-learning and VND are presented.

Keywords

open-pit strategic mine planning, artificial intelligence, machine learning, metaheuristics, Q-learning, variable neighbourhood descent.

Introduction

Solving the mine production scheduling problem (MPSP) consists of identifying which blocks should be mined during each period within the life-of-mine to maximize the total net present value (NPV). This problem is divided into block-level resolution and aggregation methods (Campos, Arroyo, and Morales, 2018) and deterministic and stochastic versions (Lamghari, Dimitrakopoulos, and Ferland, 2014a; 2014b). Finding the optimal schedule is a complex task, and a significant amount of research is indeed oriented towards developing methods to solve more detailed and realistic models (Huang *et al.*, 2020; Lamghari and Dimitrakopoulos, 2016; Villalba and Kumral, 2019). Unfortunately, this complexity involves high computation times, which has led some scholars to deal with this challenge by developing solutions under the aggregation approach (Mai *et al.*, 2019; Tabesh and Askari-Nasab, 2019). However, the block-level resolution presents advantages such as considering the temporality of the problem, opportunity cost when sequencing blocks, and the possibility of obtaining a production plan in a few steps (Campos, Arroyo, and Morales, 2018). On the deterministic side, researchers have developed efficient deterministic algorithms to manage over 1 000 000 blocks and, thus, to tackle realistic-size mines (Rezakhah, Moreno, and Newman, 2020; Rivera *et al.*, 2020; Muñoz *et al.*, 2018). Nevertheless, for open-pit mining, managing stochasticity has shown significant improvements in expected NPV, increasing of the likelihood of meeting production forecast, and finding pit limits larger than those found by deterministic approaches (Lamghari, Dimitrakopoulos, and Ferland, 2014b). This is the main reason why stochastic optimization continues to be an active area of research (Navarra, Grammatikopoulos, and Waters, 2018a).

In the context of open-pit mines, researchers have devised constraints for this type of problem, thereby defining a constrained optimization, as an increasing number of features are considered. Among these formulations, it is possible to find reserve, slope, mining capacity, processing capacity, and stockpiling constraints (Lamghari *et al.*, 2014a; Lamghari and Dimitrakopoulos, 2016). The

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

features include metal uncertainty (Lamghari Dimitrakopoulos, and Ferland, 2014b), geometallurgical modelling (Navarra, Grammatikopoulos, and Waters, 2018a), supply uncertainty (Goodfellow and Dimitrakopoulos, 2016; Senecal and Dimitrakopoulos, 2020), the actions of mineral concentrators (Navarra *et al.*, 2017), prevention of excessive movement of mining equipment between benches (Gholamnejad, Lotfian, and Kasmaeeyazdi, 2020), and commodity price uncertainty (Rimélé, Dimitrakopoulos, and Gamache, 2020).

Montiel and Dimitrakopoulos (2015) have extended their focus beyond a single mine and consider the production planning of a 'mining complex'. The authors indicate that a mining complex can be interpreted as a supply chain system where the material is transformed from one activity to another. Any change in the sequence of extraction of the mining blocks modifies the activities downstream, including blending, processing, and transporting the processed material to output stockpiles or ports. This work seeks to extend the paradigm of the MPSP into a broader holistic view (Levinson and Dimitrakopoulos, 2019; Saliba and Dimitrakopoulos, 2019).

Diverse methods have been used to solve the open-pit MPSP, including the Bienstock-Zuckerberg approach, which is based on integer programming with special decomposition techniques (Bienstock and Zuckerberg, 2010; Muñoz *et al.*, 2018). However, metaheuristics provide a useful platform for global optimization because of their ability to handle large-scale nonlinear optimization models and fast resolution times. Even though they do not necessarily provide the optimum solution (Navarra *et al.*, 2018b), they are extendable to incorporate critical features, including geometallurgical modelling and concentrator actions. Simulated annealing (Mousavi *et al.*, 2016; Sari and Kumral, 2016), Tabu Search (Lamghari and Dimitrakopoulos, 2020; Senecal and Dimitrakopoulos, 2020), variable neighbourhood descent (VND) (Navarra *et al.*, 2017), and colony optimization (Sattarvand and Niemann-Delius, 2013) are only a few of these metaheuristic approaches used. The VND algorithm developed by Lamghari, Ferland and Dimitrakopoulos (2014a) is among the most effective approaches, giving good performance on realistically sized data-sets, even on a standard desktop computer. Moreover, this approach does not depend on extraneous computation parameters such as the 'temperatures' that are required for simulated annealing (Navarra, Montiel, and Dimitrakopoulos, 2018b).

Given the absence of the extraneous computing parameters, Lamghari's VND is an appropriate metaheuristic algorithm to be hybridized into artificial intelligence (AI) frameworks (Poole and Mackworth, 2017). Indeed, the following development will introduce three new computing parameters related to learning, and it will be beneficial for these parameters not to be confounded with metaheuristic parameters. More broadly, AI-based approaches are gaining interest throughout the mining industry, as evidenced by the demand for practitioners (World Economic Forum, 2018). Besides, AI has been included as part of virtual reality applications in mining (Mitra and Saydam, 2014), considered in mine modernization processes (Jacobs and Webber-Youngman, 2017), and provided advances in mining geomechanics (McGaughey, 2020). The notion of a mine (or mining complex) that can behave as an intelligent agent is appealing, although there does not seem to be a convincing treatment of this notion in the literature.

For instance, McCoy and Auret (2019) describe AI and machine learning in the mineral processing field. One of their conclusions is the identification of a domain knowledge problem,

i.e. an interdisciplinary gap between data scientists and the current generation of experts in the mineral processing field. The authors state that plant metallurgists may not be the best choice to construct and analyse complex machine-learning models, unless they have a sufficiently broad training in modern quantitative methods. Conversely, data scientists may be equipped to run the analyses but may not be prepared to deal with mineral processing data and systems. This conclusion can be extended upstream into the mining operations, as indicated by Ali and Frimpong (2020); they conducted a study to review and analyse recent automation-related work in different sectors of the mining industry. Despite the papers they found dating between 2008 and 2019 related to AI in mine planning, Ali and Frimpong conclude that there has not been a general approach for critical tasks such as re-optimizing a shift plan of mine activities, predicting activities that can become bottlenecks, and finding patterns in productivity variance.

The authors of the present work strongly believe that a systematic view based on AI is necessary to simulate the behaviour of mining systems when working under a stochastic environment. Therefore, the objective of the study is to reduce the aforementioned interdisciplinary gap by providing the theoretical background to properly extend AI approaches to the mining context. A computational framework will be presented to illustrate the application of the concepts to mine planning and optimization problems, incorporating Lamghari's VND algorithm as a high-level function within an intelligent agent. Under this same approach, an AI algorithm is introduced as a part of the computational framework, opening the discussion about how these algorithms contribute to finding the best way of solving complex problems, extending beyond the relatively nondescript MPSP.

AI and the concept of agents

AI definitions are not unique and have been categorized into four groups: those which are related to how humans think, how human acts, how to think rationally, and how to act rationally (Russell and Norvig, 2016). For this research, an acting rationally approach will be considered, as it is more amenable to scientific development than the other categories (Russell and Norvig, 2016). Some works have used the terms AI and machine learning interchangeably; however, it is essential to clarify that machine learning is a branch of AI concerned with systems that can learn from data in a manner of being trained (Bell, 2014). Actions that are informed from previous training are indeed rational.

The concept of 'agent' is fundamental in the AI field, and is defined as something that acts in an environment (Poole and Mackworth, 2017). As shown in Figure 1, an agent is represented as a combination of a controller and a body by which it perceives stimuli, generates percepts, and gives commands to be performed through actions. A controller is the intelligence (or mind) of the agent that provides the commands, yet it has limited memory and computational capabilities. In contrast, the body is implemented either physically or computationally based on the environment in which the agent may interact.

It is relevant to mention that an agent is autonomous; therefore it is not only constrained to compute online, but it can also perform activities offline (Poole and Mackworth, 2017). Online computation means that organizational tasks are done between observing and acting in the environment, similar to a standard program that gets data, processes it, and then gives the corresponding answer when interacting with a user. In contrast, offline computation is done before it is deployed, not necessarily triggered by an interaction with the environment; however,

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

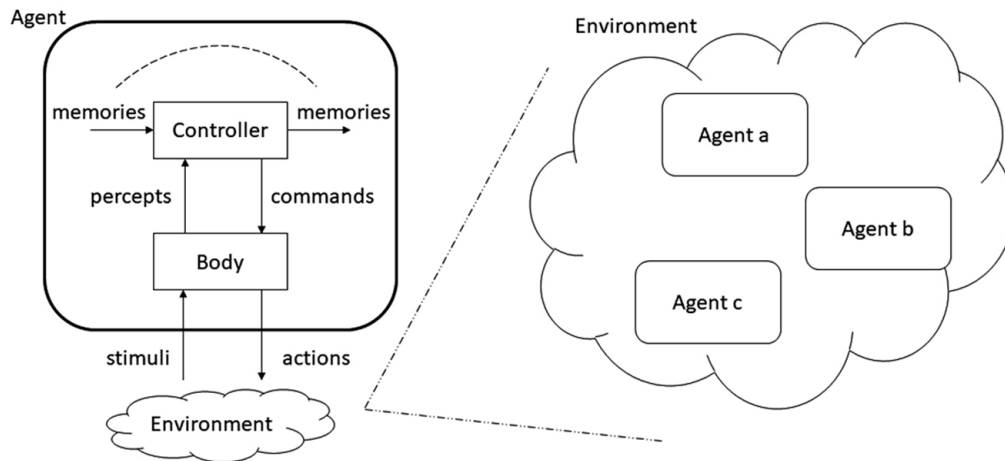


Figure 1—Parts of an agent and its relationship with the environment (adapted from Poole and Mackworth, 2017)

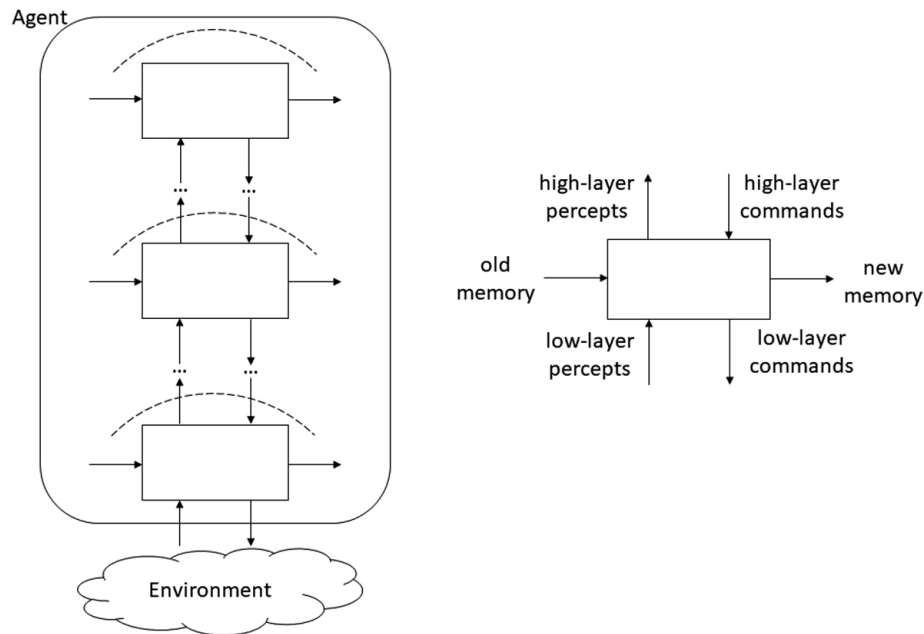


Figure 2—An agent can be seen as a hierarchy of control layers (adapted from Poole and Mackworth, 2017)

these offline computations typically alter how the agent will later respond when it is ultimately deployed in the environment. For instance, machine learning computations are typically done offline, to predetermine how the agent would eventually respond to a set of stimuli when it is online.

It is relevant to mention that an agent is autonomous; therefore it is not only constrained to compute online, but it can also perform activities offline (Poole and Mackworth, 2017). Online computation means that organizational tasks are done between observing and acting in the environment, similar to a standard program that gets data, processes it, and then gives the corresponding answer when interacting with a user. In contrast, offline computation is done before it is deployed, not necessarily triggered by an interaction with the environment; however, these offline computations typically alter how the agent will later respond when it is ultimately deployed in the environment. For instance, machine learning computations are typically done offline, to predetermine how the agent would eventually respond to a set of stimuli when it is online.

Relying on the definition given, an agent can represent any complex decision or task within a mine or mineral processing plant. The environment may be comprised of other agents, which allows interaction between the agents in a competitive or collaborative manner. Agents enable the modelling and simulation of mining systems, including what-if scenarios that clearly extend the paradigm of MPSP into a holistic view. This is critical due to the suboptimization principle of the general system theory, stating that ‘if each subsystem, regarded separately, is made to operate with maximum efficiency, the system as a whole will not operate with utmost efficiency’ (Skyttner, 2001).

The concept of ‘agent’ generally considers a hierarchy of control layers (Figure 2). Each layer has its own memory and sees the layer below as a virtual body from which it gets percepts and sends commands. Likewise, each layer sees the layer above as a virtual controller where to send percepts and from which to receive commands. This flexibility allows an agent to distribute its reasoning among its layers, reaching specializations, and thus to add more details and complexity as needed.

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

Strategic mine planning should be understood as a high-layer functionality that parameterizes decisions and actions carried out by lower layers. To represent industrial systems, including mining systems, it may be helpful to decompose the individual layers to represent distributed control systems (Eloranta *et al.*, 2014). Figure 3 illustrates a fully automated mine, in which the centralized strategic planning may consist of one or more layers; the lower layers make decisions within shorter timeframes based on tactical data, whereas the bottom layer consists of automated equipment that is distributed throughout the mine and which can even perform *in-situ* geomechanical analysis (McGaughey, 2020). The individual equipment items (e.g. robotic trucks) are equipped with sensors to adjust their immediate actions (e.g. a small course correction to avoid debris on the haul road). The sensor data for immediate action is not generally transmitted all the way up to the strategic planning layers, although unexpected incidents can be relayed part-way (e.g. sending a signal for another item of equipment, such as a road cleaner to clear the debris).

Open-pit mine plan optimization using Lamghari's VND

Stochastic mine plan optimization involves determining which blocks should be excavated during each period of the mine life, so as to maximize the expected (NPV) of the blocks to be mined. Following the development of Navarra, Grammatikopoulos, and Waters (2018), the expected NPV can be expressed as:

$$f(x) = -c^{Mining}(x) + \frac{1}{n_g} \sum_{g=1}^{n_g} v_g^{Process}(x, y_g(x)) \quad [1]$$

where

x : The strategic plan that lists which blocks are to be excavated in which time period (longterm decisions)

$yg(x)$: Given strategic plan x , describes how and when exactly the blocks are processed if scenario g is realized (short-term decisions)

$c^{Mining}(x)$: Discounted mining cost

$v_g^{Process}(x, y_g(x))$: Discounted value obtained under scenario g as incurred by long-term plan x , and short-term processing decisions $y_g(x)$

n_g : The number of geological scenarios.

The optimization of $f(x)$ is considered to be a two-stage optimization, since $y_g(x)$ is itself the result of a decision-making process,

$$y_g(x) = \operatorname{argmax}_{y \in yg(x)} (v_g^{Process}(x, y)) \quad [2]$$

in which $yg(x)$ is the set of feasible short-term decision values under geological scenario g and strategic mine plan x . Thus y_g is adjusted in function of the incoming short-term geological information that constitutes the geological scenario. Assuming that the n_g scenarios are equi-probable and generated from the same underlying geological samples, the optimization of f results in a mine plan x that is expected to perform well for the entire distribution of possible scenarios (Navarra, Grammatikopoulos, and Waters, 2018a).

By considering a distribution of geological scenarios, the resulting mine plan ensures enough flexibility to perform effective short-term decisions in the lower layers, even those which are not explicitly characterized by y_g . In practice, n_g is between 10 and 20 scenarios, and each additional scenario has a diminishing impact on the final result.

The strategic mine plan x is a list of block sets $x = [B_1, B_2, \dots, B_{n_T}]$, in which B_t is the set of blocks to be mined in period t , and n_T is the number of periods under consideration within the mine plan. Given a feasible initial solution, Lamghari's VND algorithm

Fully Automated Mine

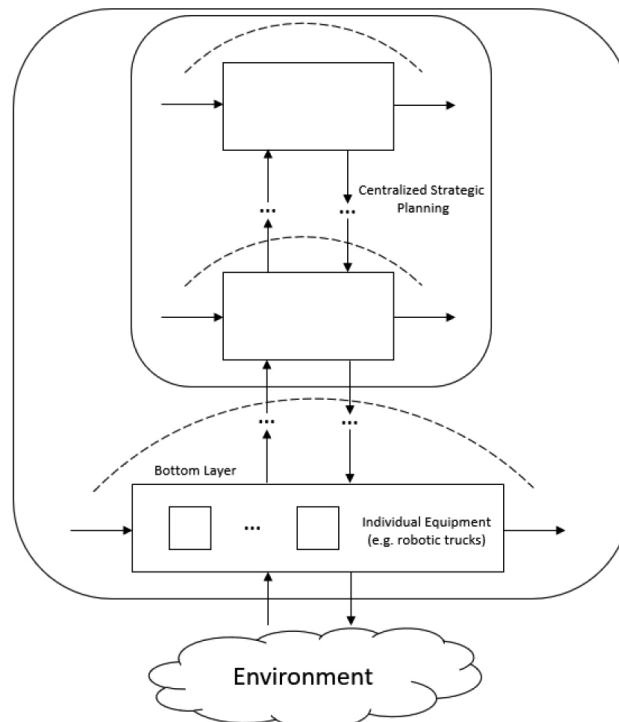


Figure 3—Fully automated mine. Equipment in the bottom layer is physically distributed throughout different sectors of the mining system

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

performs numerous modifications in which blocks are transferred between the sets B_t . The algorithm considers three types of modifications: Exchange, Shift-After, and Shift-Before. Given a particular period t ,

- *Exchange*: swaps pairs of blocks that are scheduled to be mined in periods t and $t+1$
- *Shift-After*: transfers a block and its successors extracted during the same period t , to the subsequent period $t+1$
- *Shift-Before*: transfers a block and its predecessors mined in the same period t to the preceding period $t-1$.

Each of these types of movements define a different type of ‘neighborhood’ within the solution space of feasible mine plans, hence the name ‘variable neighborhood descent’ (Hansen and Mladenovic, 2001). In traversing from one solution to the next, the algorithm only accepts movements that result in an improvement in the objective f , and which are feasible.

Feasibility implies satisfying mining capacity constraints and block precedence constraints. The mining capacity limits the number of blocks that can be excavated within a single period; in general, the capacity is expressed as a maximum tonnage of rock within a given period, considering that each block may consist of a different tonnage of rock. In the open-pit context, the block precedence constraints usually relate to the maximum slope angle required to safely access a given block; for instance, Figure 4 illustrates a 45° maximum slope, such that the grey block cannot not be excavated in period t until the white blocks (the predecessors) are excavated in period $t' \leq t$.

Lamghari’s VND algorithm is remarkably fast, capable of obtaining near-optimal solutions for industrial-scale problems on a standard laptop within a matter of hours (Lamghari, Dimitrakopoulos, and Ferland, 2014a; Navarra *et al.*, 2018b). Nonetheless, the original algorithm was not designed to support detailed descriptions of downstream operations, including mineral processing. In this implementation, any ore that is mined within a given period, and which exceeds the downstream processing capacity, is simply treated as waste rock rather than stockpiled for a future period (Lamghari, Dimitrakopoulos, and Ferland, 2014a, 2014b). However, Navarra, Grammatikopoulos, and Waters (2018) have indicated that strategic stockpiles can be represented within Lamghari’s VND through the development of customized data structures. Moreover, the work by Lamghari, Dimitrakopoulos, and Ferland (2014a; 2014b) and Navarra, Grammatikopoulos, and Waters (2018) applies two different approaches to generating an initial solution, as the former uses a linear programming approximation and the latter uses a simplified version of the VND. This aspect of initial solution generation is relevant for other open-pit MPSP metaheuristics, as well as Lamghari’s VND. Indeed, the following section presents an entirely different approach to initial solution generation and which has never been published. In principle, there may be several candidates for initial solutions, generated from different algorithms; this motivates a two-layered approach to centralized mine planning depicted in Figure 5, which fits within the larger scheme of Figure 3.

Implementation of Q-learning within the initial solution layer

The inputs into an open-pit mine plan optimization include:

- Geostatistical results from core sampling campaigns to generate scenarios
- Strategic directives that parameterize the broad layout of the

expanding open pit, to ensure the safe placement of critical equipment, tailings treatment ponds, *etc.*

- Strategic directives that balance the excavation of different rock types, to ensure favourable ore blends are fed into the process, neutralizing blends of waste rock, *etc.*
- A mechanism to generate an initial solution.

Decisions regarding exploration and sampling campaigns are beyond the scope of the MPSP as developed previously by Lamghari, Dimitrakopoulos, and Ferland (2014a, 2014b), Navarra, Grammatikopoulos, and Waters (2018), and others (Bienstock and Zuckerberg, 2010; Levinson and Dimitrakopoulos, 2019; Montiel and Dimitrakopoulos, 2015; Muñoz *et al.*, 2018; Saliba and Dimitrakopoulos, 2019); although they could be part of the future work framed by Figure 3. Moreover, Navarra, Grammatikopoulos, and Waters (2018) give only a cursory discussion regarding strategic directives for linking rock types to concentrator operational modes, identifying data structure development as an avenue for incorporating downstream realism. However, the intelligent generation of MPSP inputs does fit into the hierarchical decomposition of Figure 3. The most basic demonstration of agent-based hierarchization is in the consideration of initial solution generation (Figure 5); the mathematical programming approaches of Bienstock and Zuckerberg (Bienstock and Zuckerberg, 2010; Muñoz *et al.*, 2018), and the numerous metaheuristic approaches all rely on *ad-hoc* constructions of initial solutions which undoubtedly affect the performance of the subsequent optimization. The remainder of this section describes a new approach to initial solution generation. This new

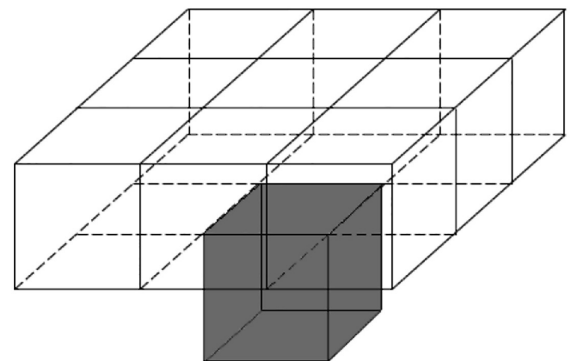


Figure 4—A block may be excavated if its nine overlying blocks have been previously mined (slope constraint)

Centralized strategic planning

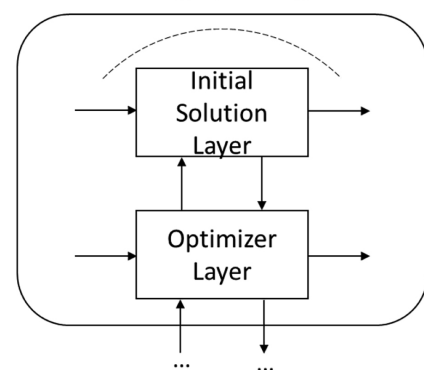


Figure 5—Centralized strategic mine planning, considering distinct layers for initial solution generation and for optimization

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

approach has led to unexpected computational results, shown in the following section, regarding what could be the conceptual division between the initial solution layer and the subsequent optimization.

As stated earlier, machine learning is a branch of AI, which postulates that agents behave rationally as a function of what they have previously learnt. There are three main ways by which an agent can do this: unsupervised learning, supervised learning, and reinforcement learning (Russell and Norvig, 2016). In particular, reinforcement learning is a variation of supervised learning, in which the agent ‘supervises itself’, i.e. the agent autosupervises the balance between exploration of new regions of the parameter space and exploitation (‘mining’) of known regions of the parameter space. In contrast, conventional supervised learning is concerned with the relation of input predictor variables \mathbf{x}_i , and output observable variables \mathbf{y}_i , and is an extension of classical statistical regression applied to data pairs $(\mathbf{x}_i, \mathbf{y}_i)$, for observations $i = 1, 2, \dots, n$. The distinction between supervised and unsupervised learning is that the former considers pairs $(\mathbf{x}_i, \mathbf{y}_i)$ in determining predictive patterns, while the latter foregoes the (human) supervised distinction between predictor and observable variables, effectively incorporating all significant variables into \mathbf{y}_i . In addition to the three main categories of machine learning (unsupervised, supervised, and reinforcement), some authors consider a fourth category ‘semisupervised learning’, which employs a combination of supervised and unsupervised techniques (Russell and Norvig, 2016).

Reinforcement learning is arguably the most appropriate approach in the initial solution layer, since the agent does not initially know the composition of the orebody, relying merely on geostatistical estimations. Reinforcement learning allows the agent to gain knowledge from a series of reward or punishment outcomes each time it acts. The agent is not told which movement to take but instead must discover which actions yield the most reward by trying them (Sutton and Barto, 2018). The concept of a policy emerges, which is the set of rules that tell the agent how to behave given a specific state (Sutton and Barto, 2018). As an agent learns a strategic planning policy, it is enabled to make better decisions and act rationally in the *ad-hoc* construction of a strategic plan.

In this regard, pioneering work was done by Askari-Nasab in developing an intelligent open-pit simulator (IOPS) that comprises algorithms based on reinforcement learning (Askari-Nasab, Frimpong, and Szymanski, 2007, 2008; Askari-Nasab and Awuah-Offei, 2008, 2009). These works are based on the aggregation method, which lacks certain advantages mentioned in the introduction, such as the consideration of the temporality of the problem, opportunity cost when sequencing blocks, and the possibility of obtaining a production plan in a few steps (Campos, Arroyo, and Morales, 2018). The current study is a further incorporation of reinforcement learning, which is better described as a block-level resolution method since the proposed learning process relies on geological scenarios, characterized by scenario-specific attribute values within each mine block, as explained in the previous section.

An orebody is represented by a set of blocks with specific coordinates XYZ. For simplicity, the orebody can be confined to a bounding cube, and each block is an equally sized right-angled prism. Moreover, each of the blocks can be ascribed a state value, i.e. a value that describes the extent to which the block data has been learnt by the agent and has hence been incorporated into the agent’s policy. In describing these state transitions, a Markov

decision process (MDP) is a classical formalization of sequential decision-making wherein actions influence not just immediate rewards, but also subsequent states and indeed the future rewards. Figure 6 presents how an agent interacts with the environment under an MDP approach. When an agent reaches a state S_t in the environment, it obtains a reward R_t , and is enabled to perform an action A_t . This action will take the agent to a state S_{t+1} , earning a reward R_{t+1} , and so on.

The actions taken to discover the patterns in an orebody are to explore north, south, east, west, up, and down as shown in Figure 7. It is important to keep in mind that these movements are not necessarily the way that blocks will be extracted but are only the way the agent is exploring the orebody for learning purposes, i.e. in the construction of its planning policy.

Q-learning is an algorithm that allows the agent to learn the optimal policy from its history of querying the environment. A Q-learning agent learns a Q-function, which gives the expected utility of taking a given action in a state, the so-called Q-value or ‘quality’. Even though after many iterations the Q-values converge, these values should not be confused with the real utility of the movement; it is only a way of creating a policy based on the expected utility. This approach seems appropriate for the construction of an initial solution for the MPSP, but it is questionable how much subsequent optimization will be required. The basic algorithm is presented in Figure 8.

The outcome of a Q-value is based on the state and the action $Q[s, a]$, which means that from a state with four possible movements it will consist of four Q-values. The update of the two-dimensional array $Q[s, a]$ is given by line 16 of the algorithm.

$$Q[s, a] \leftarrow Q[s, a] + \alpha(r + \gamma \max_{a'} Q[s', a'] - Q[s, a]) \quad [3]$$

where s is a state, a is the action carried out on the state, s' is the next state, a' is the next action from this next state s' , r is the reward obtained after being in state s and performing action a , and finally the parameters α and γ are the learning and discount factors respectively, both of which are in the range 0 to 1. The

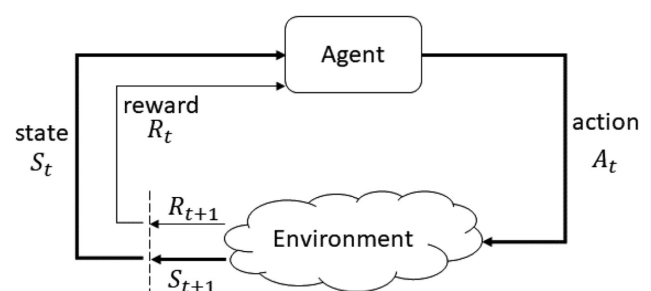


Figure 6—Interaction under an MDP approach (Sutton and Barto, 2018)

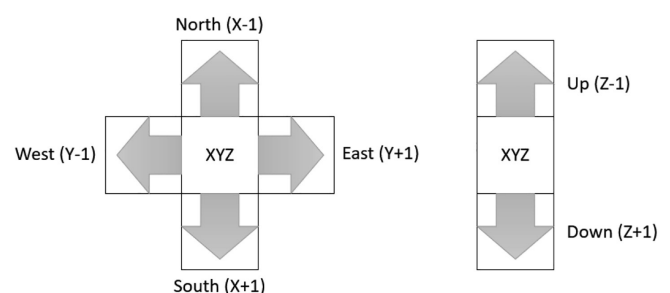


Figure 7—Actions considered in order to learn from the orebody

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

controller Q-learning(S,A,γ,α)

```

2:   Inputs
3:       S is a set of states
4:       A is a set of actions
5:       γ is the discount factor
6:       α is the learning factor
7:   Local
8:       real array Q[S,A]
9:       state s, s' ∈ S
10:      action a ∈ A
11:  initialize Q[S,A] arbitrarily
12:  observe current state s
13:  repeat
14:      select and carry out an action a
15:      observe reward r and next state s'
16:       $Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma \max_{a'} Q[s',a'] - Q[s,a])$ 
17:       $s \leftarrow s'$ 
18:  until termination
    
```

Figure 8—Q-learning algorithm (adapted from Poole and Mackworth, 2017)

learning factor α quantifies the contribution of the (modified) difference between the maximum Q-values of the next state and the current Q-value, in updating the Q-value array. The discount factor γ measures the extent to which the possibility of future rewards is prioritized at the expense of known short-term gains; if $\gamma = 1$, the equation favours patience in considering that the motion s to s' may eventually lead to future possible rewards, whereas $\gamma = 0$ causes the agent to be greedy for the immediate rewards as it ignores all possible future rewards.

Data scientists consider that an agent must balance short-term exploitation to maximize its immediate reward and exploration to maximize its long-term wellbeing. Pure exploitation risks getting stuck in a rut, whereas pure exploration to improve one's knowledge is of no ultimate use if the learnt knowledge is not implemented in an action plan. An additional parameter ϵ is introduced in line 14 of the algorithm (Figure 8) to select an action based on the following random function.

$$f(\epsilon, s) = \begin{cases} \max_a Q(s), & \text{with probability } 1 - \epsilon \\ \text{random } a, & \text{otherwise} \end{cases} \quad [4]$$

Therefore, the current implementation of Q-learning considers three parameters: α , γ , and ϵ . However, this is a manageable number since the optimizer layer is based on Lamghari's VND algorithm and thus does not introduce any additional parameters. The tuning of $(\alpha, \gamma, \epsilon)$ could in principle be a task that would be part of the initial solution layer (Figure 5), measuring the performance of the subsequent optimization as a function of $(\alpha, \gamma, \epsilon)$. However, the following sample computations consider static values of $(\alpha, \gamma, \epsilon) = (1, 0.7, 0.3)$, which were deemed acceptable from preliminary computational trials. Additionally, all of the Q-values have been initiated to zero, but different configurations may also be considered as part of future research.

In the following sample computations, the initial mine plan is constructed from the Q-learning algorithm, considering the actions defined by Figure 7. As a first action, the surface block with the highest economic block is found ($Z=1$) and is selected to be mined in the first period, i.e. it is included in B1. In the current scope, each subsequent action is to extract a selected block plus all its predecessors, while respecting the slope (Figure 4) and mining capacity constraints. The actions are north, south, west, east, and down. For instance, Figure 9 presents the blocks that would be mined if the action 'down' were selected.

The following section is the first attempt at developing an automated mine as a hierarchically controlled intelligent agent, which segregates the MPSP optimizer from one of its computational inputs. The computational input is, in this case, the initial solution. Other inputs could be the results of dynamic geological sampling and metallurgical plant tests (Navarra, Grammatikopoulos, and Waters, 2018a), which may be the subject of future work.

Sample computations

An agent described by Figure 5 has been applied to an ore deposit composed of 9953 blocks, in which the initial solution is generated according to the Q-learning algorithm described in the previous section, and the optimization is based on Lamghari's VND algorithm. However, it is relevant to mention that since the agent includes this orebody within a bounding cube, the number of blocks increases to 144 420. The bounding cube includes all potential blocks that would be predecessors of the original 9953 blocks. Conditional simulation has allowed the generation of 20 scenarios for this deposit, 10 of which have been used to construct an optimized mine plan, while the other 10 are used for an a-posteriori construction of cumulative NPV profiles that will be discussed below (Figure 10).

The sample computations are based on the same set of geological scenarios presented by Navarra, Grammatikopoulos,

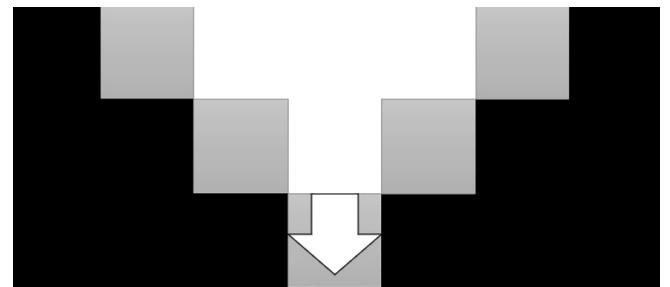


Figure 9—Blocks extracted (in grey) if action 'down' is selected

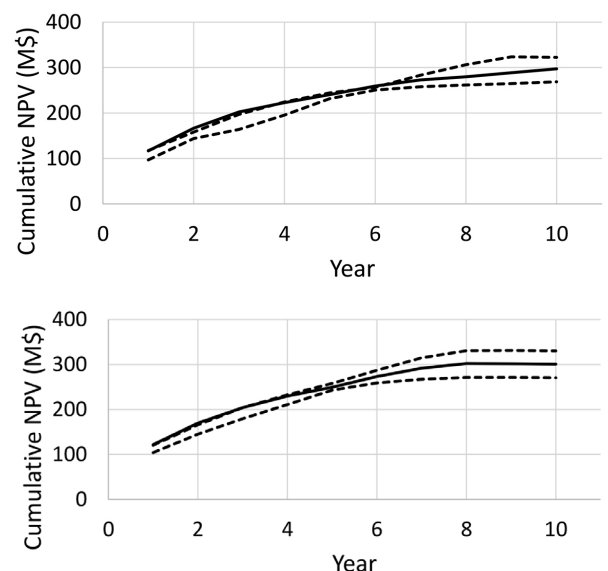


Figure 10—Cumulative NPV. 10th, 50th, and 90th percentiles, (a) before optimization, and (b) after optimization

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

and Waters (2018) to provide a comparison. Each of the blocks has identical dimensions, 20 m × 20 m × 10 m, and identical weight of 10 000 t. Based on the scenario, the blocks have a higher or lower grade of a certain metal, which varies from 0.01% to 4.14%, with an average of 0.33%. The operational data for this calculation is presented on Table I.

The Q-learning and VND algorithms were programmed in C++ and run on a standard ASUS laptop with an Intel Core™ i5 CPU, and 8 GB of RAM. The total computations were completed in roughly 60 minutes. The previous computations in 2018 apply the VND algorithm and were completed is roughly 90 minutes; this earlier work used a slightly older laptop, having an i3 CPU and 4 GB of RAM. It was surprising, however, that the newer computations devote over 95% of the computation time to initial solution generation, whereas the previous work devoted roughly 50%. This has caused us to reinterpret the potential roles of the initial construction of a plan, *versus* its final optimization (*i.e.* refinement), which will be discussed below.

The learning parameters (α, γ, ϵ) were set to (1, 0.7, 0.3) following preliminary experimentation. For simplicity, the preliminary tests assumed $\alpha = 1$, and $(\gamma + \epsilon) = 1$, and so only the balance between γ and ϵ was modified. The control of these learning parameters may be a moot point, however, since an initial solution layer can include parallel computing that would simultaneously attempt a spectrum of parameter values (and possibly even several completely different algorithms), transmitting only the most promising ones as candidates for subsequent optimization. Moreover, this balance between γ and ϵ did not affect the outcome in any observable manner.

It was surprising that the initial solution generator produced high-value solutions comparable to those obtained in previous work (Navarra, Grammatikopoulos, and Waters, 2018a), with the life-of-mine NPV of approximately \$300 million over ten years (Figure 10a). Furthermore, the subsequent VND optimization was completed in under three minutes, and did not result in a notable increase in the life-of-mine NPV, as the spread between the 10th and 90th percentiles is \$269 million to \$323 million for the preoptimized (Figure 10a) and \$271 million to \$330 million for the optimized (Figure 10b); thus the 10th percentile was improved by 1% and the 90th percentile by 2%, which is surprisingly small.

The VND algorithm clearly does have an impact. However,

Table I

Operational data

Effective metal price	\$5000 per ton
Annual discount rate	8%
Maximum pit angle	45°
Mining capacity	15 Mt/a
Mining cost	\$1.5 per ton
Processing capacity	6 Mt/a
Processing cost	\$6 per ton
Recovery	85%

since the optimized plan effectively completes the production in eight years, rather than ten, considering that the curves in Figures 11 and 10b are approximately flat after year 8; for comparison, the pre-optimized solution yields between \$261 million and \$306 million in the first eight years, whereas the optimized solution yields the complete life-of-mine NPV, between \$271 million and \$330 million, which is between 4% and 8% higher. Moreover, the optimized solution is observed to be less erratic in the first five years of the mine life, as all three curves follow a gradual attenuation. The optimization also has a noteworthy impact on the shape of the pits, as illustrated in Figure 12; the optimized plan is organized such that only the very bottom of the deposit is excavated after the end of year 8 (Figure 12b), whereas the pre-optimized plan continues a broad advancement into years 9 and 10 (Figure 12a). The same figures also indicate the high number of active benches in the periods, which typically happens in the absence of a dedicated constraint, as developed by Gholamnejad, Lotfian, and Kasmaeeyazdi (2020). In practice this might be fixed in a subsequent step by a postprocessing algorithm that refines the mine plan generated, although approaches similar to Gholamnejad, Lotfian, and Kasmaeeyazdi (2020) may be superior.

The authors were genuinely surprised by the effectiveness of the Q-learning, but refrain from drawing overly broad conclusions. For the given sample calculations, much of the computational effort was consumed by the initial solution generation, which resulted in a high expected NPV. The subsequent application of the VND optimization was comparatively short and did not significantly increase the life-of-mine NPV, but did however enhance the solution by shortening the life of mine and smoothing

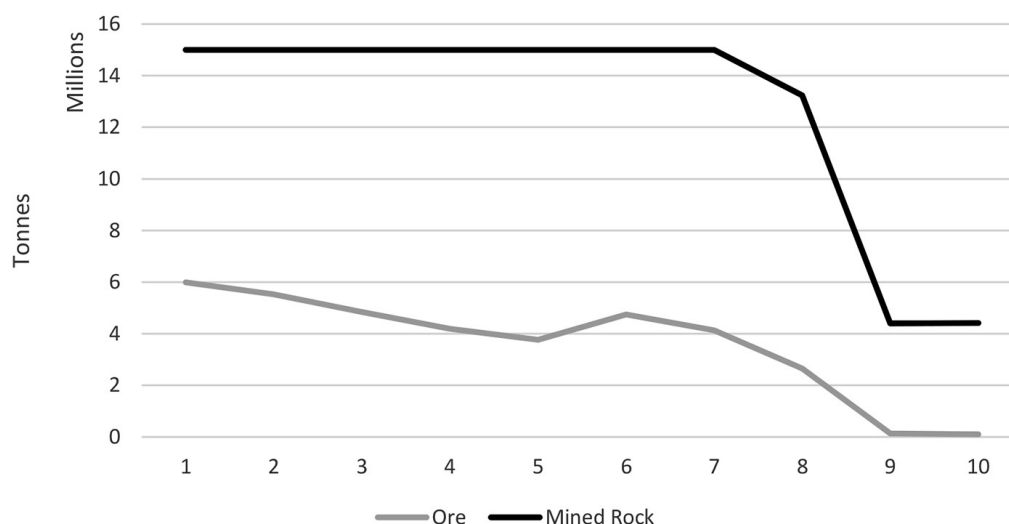


Figure 11—Production chart of the schedule after optimization

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

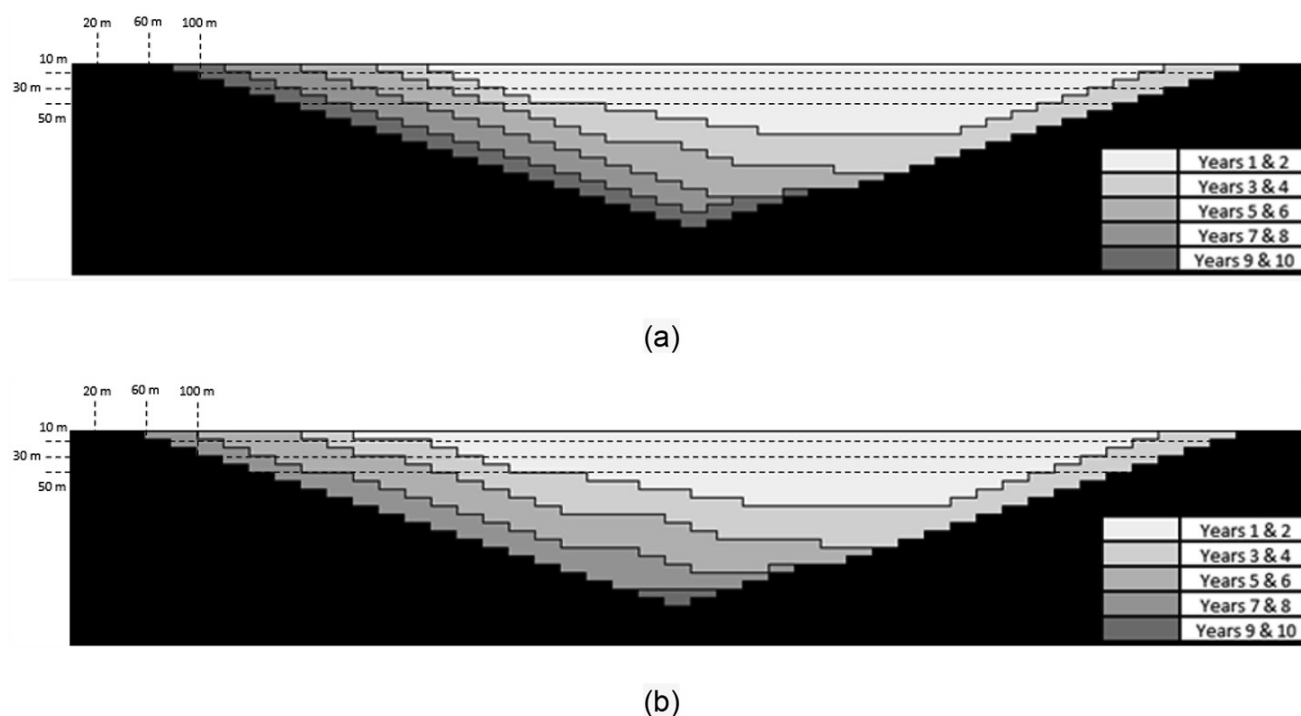


Figure 12—XZ-planar cut along Y = 6 of mine plan (a) prior to optimization and (b) after optimization

the cumulative NPV profiles. Given that the VND algorithm has been previously published and validated, its impact within the current implementation is to refine the plan constructed by Q-learning, as well as creating a final plan that is validated. Indeed, the VND algorithm ensures that the final solution cannot be further improved within the Exchange, Shift-Before, and Shift-After neighbourhoods that were initially conceived by Lamghari, Dimitrakopoulos, and Ferland (2014a). The definition of additional neighbourhoods would allow further avenues of refinement and validation for the results of Q-learning, or potentially any other approach to initial plan generation.

Conclusions and future work

This paper introduced essential notions of AI and machine learning to embed strategic mine planning within the higher layers of an intelligent agent. The intent of this work is to favour future interdisciplinary work. The concepts were demonstrated in the generation of sample computations, showing their suitability in contributing to MPSP research (Navarra, Grammatikopoulos, and Waters, 2018a). Moreover, an agent can be divided into several layers to reach specialization, allowing the hierarchical modelling of complex mine operations problems, thereby extending the scope of MPSP research to consider the tactical coordination of robotic equipment (Figure 3).

Within the sample computations, the policy developed by the Q-learning algorithm explores the orebody successfully, as it identifies those blocks that if excavated sooner will ultimately result in a higher NPV. A similar approach could be adapted for underground mines. Other opportunities for future work include:

- Further comparisons and hybridization of initial solution generation, which could be refined through VND optimization
- Inclusion of more features in the agent such as operational modes, stockpiling, efficient number of active benches, coordination of robotic equipment, etc.

- Improvements to deal with more extensive mines (a greater number of blocks)
- Exploration of the collaborative and competitive relation between agents within mining systems
- Finding new ways of applying other AI structures (such as neural networks) to MPSPs.

Acknowledgments

Funding for this work was provided by the Chilean government through its Becas Chile programme allowing students to pursue graduate degrees abroad.

References

- ALI, D. and FRIMPONG, S. 2020. Artificial Intelligence, machine learning and process automation: Existing frontier and way forward for mining sector. *Artificial Intelligence Review*, vol. 53, no. 8. pp. 1049–1067.
- ASKARI-NASAB, H., FRIMPONG, S., and SZYMANSKI, J. 2007. Modelling open pit dynamics using discrete simulation. *International Journal of Mining, Reclamation and Environment*, vol. 21, no. 1. pp. 35–49.
- ASKARI-NASAB, H., FRIMPONG, S., and SZYMANSKI, J. 2008. Investigating continuous time open pit dynamics. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 108, no. 2. pp. 61–71.
- ASKARI-NASAB, H. and AWUAH-OFFEI, K. 2008. An agent based framework for open pit mine planning. *CIM Journal*, vol. 110. pp. 145–154.
- ASKARI-NASAB, H. and AWUAH-OFFEI, K. 2009. Open pit optimization using discounted economic block values. *Mining Technology*, vol. 118, no. 1. pp. 1–12.
- BELL, J. 2014. *Machine Learning*. Wiley.
- BIENSTOCK, D. and ZUCKERBERG, M. 2010. Solving LP relaxations of large-scale precedence constrained problems. *Proceedings of the 14th International Conference of Integer Programming and Combinatorial Optimization*. Einsenbrand F. and Shepherd, F.B. (eds). Lecture Notes in Computer Science, vol. 6080. Springer, Berlin. pp. 1–14.

Integration of strategic open-pit mine planning into hierarchical artificial intelligence

- CAMPOS, P, ARROYO, C., and MORALES, N. 2018. Application of optimized models through direct block scheduling in traditional mine planning. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 118, no. 4. pp. 381–386.
- ELORANTA, V., KOSKINEN, J., LEPPANEN, M., and REIJONEN, V. 2014. Designing Distributed Control Systems: A Pattern Language. Wiley.
- GHOLAMNEJAD, J., LOTFIAN, R., and KASMAEYAZDI, S. 2020. A practical, long-term production scheduling model in open pit mines using integer linear programming. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 120, no. 12. pp. 665–670.
- GOODFELLOW, R. and DIMITRAKOPOULOS, R. 2016. Global optimization of open pit mining complexes with uncertainty. *Applied Soft Computing*, vol. 40, no. 1. pp. 292–304.
- HANSEN, P. and MLADENOVIC, N. 2001. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, vol. 130, no. 3. pp. 449–467.
- HUANG, S., LI, G., BEN-AWUAH, E., AFUM, B.O., and HU, N. 2020. A stochastic mixed integer programming framework for underground mining production scheduling optimization considering grade uncertainty. *IEEE Access*, vol. 8. pp. 24495–24505.
- JACOBS, J. and WEBBER-YOUNGMAN, R.C.W. 2017. A technology map to facilitate the process of mine modernization throughout the mining cycle. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 117, no. 7. pp. 636–647.
- LAMGHARI, A. and DIMITRAKOPOULOS, R. 2016. Network-flow based algorithms for scheduling production in multi-processor open-pit mines accounting for metal uncertainty. *European Journal of Operational Research*, vol. 250, no. 1. pp. 273–290.
- LAMGHARI, A. and DIMITRAKOPOULOS, R. 2020. Hyper-heuristics approaches for strategic mine planning under uncertainty. *Computers & Operations Research*, vol. 115. <https://doi.org/10.1016/j.cor.2018.11.010>
- LAMGHARI, A., DIMITRAKOPOULOS, R., and FERLAND, J. 2014a. A hybrid method based on linear programming and variable neighborhood descent for scheduling production in open-pit mines. *Journal of Global Optimization*, vol. 63, no. 3. pp. 555–582.
- LAMGHARI, A., DIMITRAKOPOULOS, R., and FERLAND, J. 2014b. A variable neighbourhood descent algorithm for the open-pit mine production scheduling problem with metal uncertainty. *Journal of the Operational Research Society*, vol. 65, no. 9. pp. 1305–1314.
- LEVINSON, Z. and DIMITRAKOPOULOS, R. 2019. Simultaneous stochastic optimisation of an open-pit gold mining complex with waste management. *International Journal of Mining Reclamation and Environment*, vol. 34, no. 6. pp. 1–15.
- MAI, N., TOPAL, E., ERTEN, O., and SOMMERVILLE, B. 2019. A new risk-based optimisation method for the iron ore production scheduling using stochastic integer programming. *Resources Policy*, vol. 62. pp. 571–579.
- MCCOY, J. and AURET, L. 2019. Machine learning applications in mineral processing: A review. *Minerals Engineering*, vol. 132, no. 1. pp. 95–109.
- MCGAUGHEY, J. 2020. Artificial intelligence and big data analytics in mining geomechanics. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 120, no. 1. pp. 15–21.
- MITRA, R. and SAYDAM, S. 2014. Can artificial intelligence and fuzzy logic be integrated into virtual reality applications in mining? *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 114, no. 12. pp. 1009–1016.
- MONTIEL, L. and DIMITRAKOPOULOS, R. 2015. Optimizing mining complexes with multiple processing and transportation alternatives: An uncertainty-based approach. *European Journal of Operational Research*, vol. 247, no. 1. pp. 166–178.
- MOUSAVI, A., KOZAN, E., and LIU, S. 2016. Open-pit block sequencing optimization: A mathematical model and solution technique. *Engineering Optimization*, vol. 48, no. 11. pp. 1932–1950.
- MUÑOZ, G., ESPINOZA, M., GOYCOOLEA, M., MORENO, E., QUEYRANNE, M., and RIVERA, O. 2018. A study of the Bienstock-Zuckerberg algorithm: Applications in mining and resource constrained project scheduling. *Computational Optimization and Applications*, vol. 69. pp. 501–534.
- NAVARRA, A., GRAMMATIKOPOULOS, T., and WATERS, K. 2018a. Incorporation of geometallurgical modelling into long-term production planning. *Minerals Engineering*, vol. 120, no. 1. pp. 118–126.
- NAVARRA, A., MENZIES, A., JORDENS, A., and WATERS, K. 2017. Strategic evaluation of concentrator operational modes under geological uncertainty. *International Journal of Mineral Processing*, vol. 164, no. 1. pp. 45–55.
- NAVARRA, A., MONTIEL, L., and DIMITRAKOPOULOS, R. 2018b. Stochastic strategic planning of open-pit mines with ore selectivity recourse. *International Journal of Mining, Reclamation and Environment*, vol. 32, no. 1. pp. 1–17.
- POOLE, D. and MACKWORTH, R. 2003. Artificial Intelligence: Foundations of Computational Agents. 2nd edn. Cambridge University Press, Cambridge.
- REZAKHAH, M., MORENO, E., and NEWMAN, A. 2020. Practical performance of an open pit mine scheduling model considering blending and stockpiling. *Computers and Operations Research*, vol. 115. doi: 10.1016/j.cor.2019.02.001
- RIMÉLÉ, A., DIMITRAKOPOULOS, R., and GAMACHE, M. 2020. A dynamic stochastic programming approach for open-pit mine planning with geological and commodity price uncertainty. *Resources Policy*, vol. 65. <https://doi.org/10.1016/j.resourpol.2019.101570>
- RIVERA, O., ESPINOZA, D., GOYCOOLEA, M., MORENO, E., and MUÑOZ, G. 2020. Production scheduling for strategic open pit mine planning: A mixed-integer programming approach. *Operations Research*, vol. 68, no. 5. pp. 1425–1444.
- RUSSELL, S. and NORVIG, P. 2016. Artificial Intelligence: A Modern Approach. 3rd edn. Pearson Education, Harlow.
- SALIBA, Z. and DIMITRAKOPOULOS, R. 2019. Simultaneous stochastic optimization of an open-pit mining complex with tailings management. *Mining Technology*, vol. 128, no. 4. pp. 216–229.
- SARI, Y. and KUMRAL, M. 2016. An improved meta-heuristics approach to extraction sequencing and block routing. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 116, no. 7. pp. 673–680.
- SATTARVA ND, J. and NIEMANN-DELIUS. 2013. A new metaheuristic algorithm for longterm open-pit production planning. *Archives of Mining Sciences*, vol. 58, no. 1. pp. 107–118.
- SENEGAL, R. and DIMITRAKOPOULOS, R. 2020. Long-term mine production scheduling with multiple processing destinations under mineral supply uncertainty, based on multi-neighbourhood Tabu search. *International Journal of Mining, Reclamation and Environment*, vol. 34, no. 7. pp.459–475.
- SKYTTER, L. 2001. General Systems Theory: Ideas and Applications. World Scientific Publishing, Singapore.
- SUTTON, R. and BARTO, A. 2018. Reinforcement Learning. MIT Press, London.
- TABESH, M. and ASKARI-NASAB, H. 2019. Clustering mining blocks in presence of geological uncertainty. *Mining Technology*, vol. 128, no. 3. pp. 162–176.
- VILLALBA, M. and KUMRAL, M. 2019. Underground mine planning: stope layout optimization under grade uncertainty using genetic algorithms. *International Journal of Mining, Reclamation and Environment*, vol. 33, no. 5. pp. 353–370.
- WORLD ECONOMIC FORUM. 2018. The Future of Jobs Report. Geneva