minerals

MDPI

Article

Integrating Production Planning with Truck-Dispatching Decisions through Reinforcement Learning While Managing Uncertainty

Joao Pedro de Carvalho * and Roussos Dimitrakopoulos *

COSMO—Stochastic Mine Planning Laboratory, Department of Mining and Materials Engineering, McGill University, 3450 University Street, Montreal, QC H3A 0E8, Canada
* Correspondence: joao.decarvalho@mail.mcgill.ca (J.P.d.C.); roussos.dimitrakopoulos@mcgill.ca (R.D.)

Abstract: This paper presents a new truck dispatching policy approach that is adaptive given different mining complex configurations in order to deliver supply material extracted by the shovels to the processors. The method aims to improve adherence to the operational plan and fleet utilization in a mining complex context. Several sources of operational uncertainty arising from the loading, hauling and dumping activities can influence the dispatching strategy. Given a fixed sequence of extraction of the mining blocks provided by the short-term plan, a discrete event simulator model emulates the interaction arising from these mining operations. The continuous repetition of this simulator and a reward function, associating a score value to each dispatching decision, generate sample experiences to train a deep Q-learning reinforcement learning model. The model learns from past dispatching experience, such that when a new task is required, a well-informed decision can be quickly taken. The approach is tested at a copper–gold mining complex, characterized by uncertainties in equipment performance and geological attributes, and the results show improvements in terms of production targets, metal production, and fleet management.

Keywords: truck dispatching; mining equipment uncertainties; orebody uncertainty; discrete event simulation; Q-learning

check for updates

Citation: de Carvalho, J.P.; Dimitrakopoulos, R. Integrating Production Planning with Truck-Dispatching Decisions through Reinforcement Learning While Managing Uncertainty. *Minerals* **2021**, *11*, 587. https://doi.org/10.3390/min11060587

Academic Editors: Rajive Ganguli, Sean Dessureault and Pratt Rogers

Received: 16 April 2021
Accepted: 26 May 2021
Published: 31 May 2021

1. Introduction

In short-term mine production planning, the truck dispatching activities aim to deliver the supply material, in terms of quantity and quality, being extracted from the mining fronts by the shovels to a destination (e.g., processing facility, stockpile, waste dump). The dispatching decisions considerably impact the efficiency of the operation and are of extreme importance as a large portion of the mining costs are associated with truck-shovel activities [1–4]. Truck dispatching is included under fleet optimization, which also comprises equipment allocation, positioning shovels at mining facies and defining the number of trucks required [2,5,6]. Typically, the truck dispatching and allocation tasks are formulated as a mathematical programming approach whose objective function aims to minimize equipment waiting times and maximize production [7–11]. Some methods also use heuristic rules to simplify the decision-making strategy [12–14]. In general, a limiting aspect of the structure of these conventional optimization methods is related to the need to reoptimize the model if the configuration of the mining complex is modified, for example, if a piece of fleet equipment breaks. Alternatively, reinforcement learning (RL) methods [15] provide means to make informed decisions under a variety of situations without retraining, as these methods learn from interacting with an environment and adapt to maximize a specific reward function. The ability to offer fast solutions given multiple conditions of the mining complex is a step towards generating real-time truck dispatching responses. Additionally, most methods dealing with fleet optimization are applied to single mines, whereas an industrial mining complex is a set of integrated operations and facilities transforming geological resource supply into sellable products. A mining complex can include multiple mines, stockpiles, tailing dams, processing routes, transportation systems, equipment types and sources of uncertainty [16–27].

The truck-dispatching model described herein can be viewed as a particular application belonging to the field of material delivery and logistics in supply chains, commonly modelled as vehicle routing problems and variants [28–30]. Dynamic vehicle routing problems [31,32] are an interesting field which allows for the inclusion of stochastic demands [33] and situations where the customer's requests are revealed dynamically [34]. These elements can also be observed in truck-dispatching activities in mining complexes, given that different processors have uncertain performances and that production targets may change, given the characteristics of the feeding materials. One particularity of the truck-dispatching model herein is that the trips performed between shovels and destinations usually have short lengths and are repeated multiple times. Another important aspect is that uncertainty arises from the geological properties of the transported materials and the performance of different equipment. Over the last two decades, there is an effort to develop frameworks accommodating uncertainties in relevant parameters within the mining complex operations to support more informed fleet management decisions. Not accounting for the complexity and uncertainties inherent to operational aspects misrepresent queue times, cycling times and other elements, which inevitably translates to deviation from production targets [6,35]. Ta et al. [9] allocate the shovels by a goal programming approach, including uncertainties in truckload and cycle times. Few other approaches optimize fleet management and production scheduling in mining complexes under both geological and equipment uncertainty [22,36,37].

A common strategy to model the stochastic interactions between equipment and processors in an operating mining environment is through the use of discrete event simulation (DES) approaches [35,38–43]. The DES allows for replacing an extensive mathematical description or rule concerning stochastic events by introducing randomness and probabilistic parameters related to a sequence of activities. The environment is characterized numerically by a set of observable variables of interest, such that each event modifies the state of the environment [44]. This simulation strategy can be combined with ideas from optimization approaches. Jaoua et al. [45] describe a detailed truck-dispatching control simulation, emulating real-time decisions, coupled with a simulated annealing-based optimization that minimizes the difference between tonnage delivered and associated targets. Torkamani and Askari-Nasab [35] propose a mixed integer programming model to allocate shovels to mining facies and establish the number of required truck trips. The solution's performance is assessed by a DES model that includes stochastic parameters such as truck speed, loading and dumping times, and equipment failure behavior. Chaowasakoo et al. [46] study the impact of the match factor to determine the overall efficiency of truck-shovel operations, combining a DES and selected heuristics maximizing production. Afrapoli et al. [47] propose a mixed integer goal programming to reduce shovel and truck idle times and deviations from production targets. A simulator of the mining operations triggers the model to be reoptimized every time a truck requires a new allocation. Afrapoli et al. [11] combine a DES with a stochastic integer programming framework to minimize equipment waiting times.

It is challenging to formulate all the dynamic and uncertain nature of the truck-shovel operation into a mathematical formulation. The daily operations in a mining complex are highly uncertain; for example, equipment failure, lack of staff or weather conditions can cause deviations in production targets and cause modifications in the dispatching policy. These events change the performance of the mining complex; thus, the related mathematical programming model needs to be reoptimized. The use of DES of the mining complex facilitates the modelling of such events. Note that some of the above mentioned approaches simulate the mining operations to assess the dispatching performance or improve it, using heuristic approaches. This strategy can generate good solutions, but the models do not learn from previous configurations of the mining complex.

Unlike the in the mentioned heuristic approaches, RL-based methods can take advantage of a mining complex simulator to define agents (decision-makers) that interact with this environment based on actions and rewards. The repetition of such interaction provides these agents with high learning abilities, which enables fast responses when a new assignment is required. Recent approaches have achieved high-level performances over multiple environments that require complex and dynamic tasks [48–55]. They have also been applied to some short-term mine planning aspects showing interesting results [56–58].

This paper presents a truck-dispatching policy based on deep Q-learning, one of the most popular RL approaches, in order to improve daily production and overall fleet performance, based on the work in Hasselt et al. [50]. A DES is used to model daily operational aspects, such as loading, hauling and dumping activities, generating samples, to improve the proposed truck dispatching policy. A case study applies the method to a copper–gold mining complex, which considers equipment uncertainty, modelled from historical data, and orebody simulations [59–63] that assess the uncertainty and variability related to metal content within the resource model. Conclusion and future work follow.

2. Method

The proposed method adapts the deep double Q-learning neural network (DDQN) method [50] for dispatching trucks in a mining environment. The RL agents continually take actions over the environment and receive rewards associated with their performances [15]. Herein, each mining truck is considered an agent; therefore, these terms are used interchangeably throughout this paper. The DES, described in Section 2.1, receives the decisions made by the agents, simulates the related material flow and a reward value evaluating each action. Section 2.2 defines the reward function and how the agents interact with the RL environment; where the observed states and rewards compose the samples used to train the DDQN. Subsequently, Section 2.3 presents the training algorithm based on Hasselt et al. [50], updating the agent's parameters (neural network weights). Figure 1 illustrates the workflow showing the interaction between the DES and the DDQN policy.

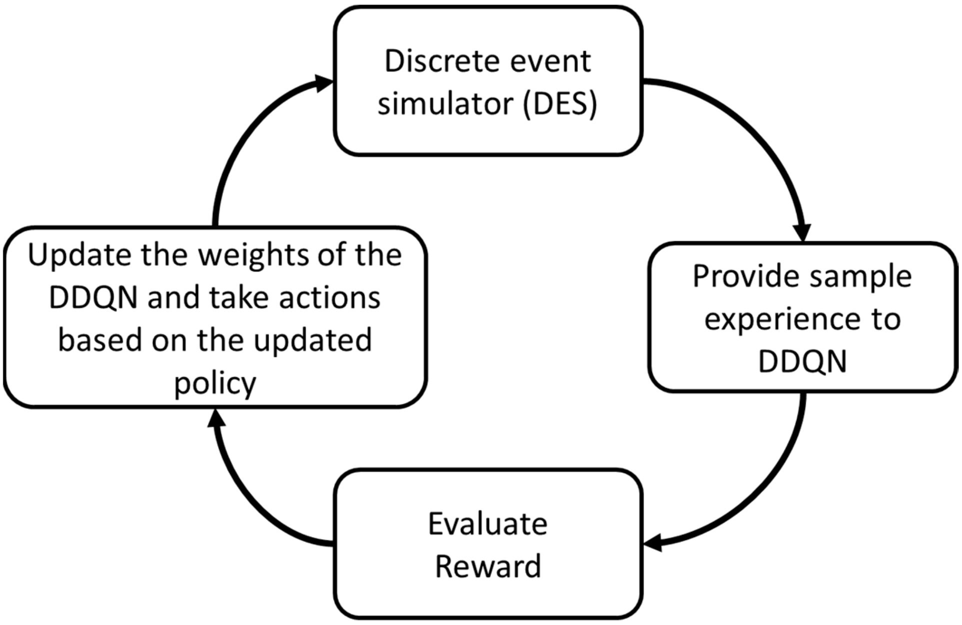

Discrete event simulator (DES)
Provide sample experience to DDQN
Evaluate Reward
Update the weights of the DDQN and take actions based on the updated policy

Figure 1. Workflow of the interaction between the DES and the DDQN method.

2.1. Discrete Event Simulator

The discrete event simulator presented in this work assumes a predefined sequence of extraction, the destination policy of each mining block and the shovel allocation. It also presumes that the shortest paths between shovels and destinations have been defined.

Figure 2 illustrates this relationship where the black arrow is the predefined destination path for the block being extracted by the shovel. After the truck delivers the material to the dumping point (waste dump, processing plant or leaching pad, for example), a dispatching policy must define the next shovel assignment. The red arrow illustrates the path options for dispatching.

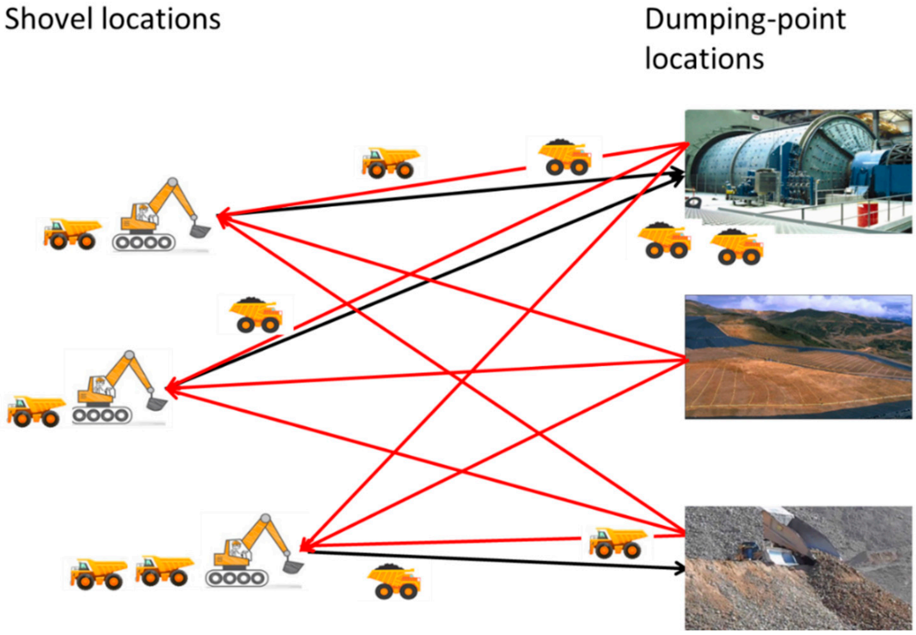


Figure 2. Representation of the possible paths a truck can follow: the pre-defined destination of each block (black arrow); the possible next dumping point (red arrows).

To simulate the operational interactions between shovels, trucks and dumping locations present in the mining complex, the DES considers the following major events:

Shovel Loading Event: The shovel loads the truck with an adequate number of loads. The total time required for this operation is stochastic, and once the truck is loaded, it leaves the shovel as the destination, triggering the "Truck Moving Event." If the shovel must move to a new extraction point, it incurs a delay, representing the time taken to reposition the equipment. After the truck leaves the loading point, this event can trigger itself if there is another truck waiting in the queue.

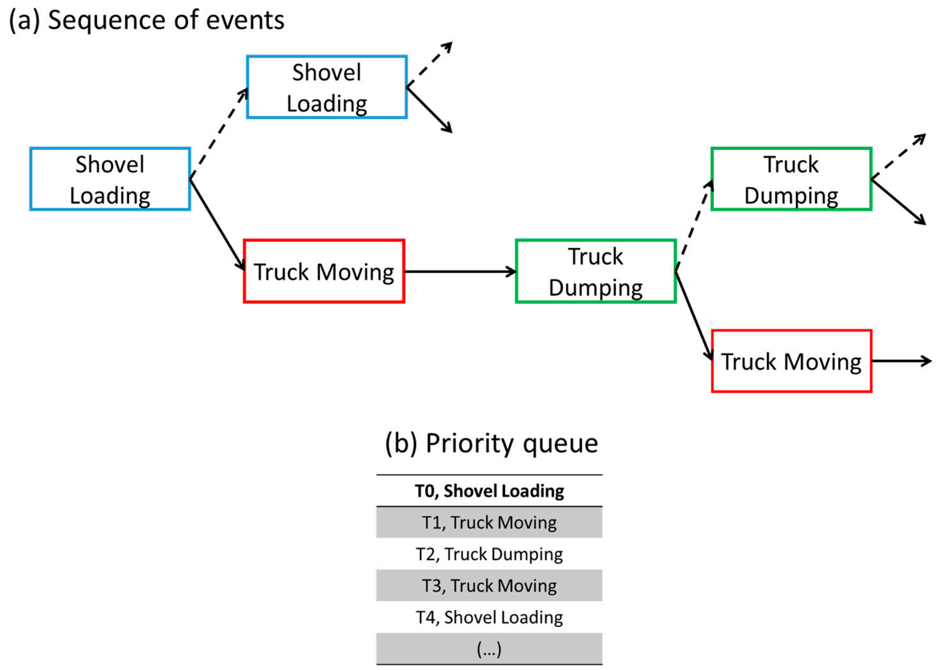
Truck Moving Event: This event represents the truck going from a shovel to a dumping location, or vice versa. Each travelling time is sampled from a distribution approximated from historical data. Travelling empty or loaded impacts on the truck speed, meaning that time values are sampled from different distributions in these situations. When the truck arrives at the loading point and the shovel is available, this event triggers a "Shovel Loading Event"; otherwise, it joins the queue of trucks. If the truck arrives at the dumping location, the event performs similarly; if the destination is empty, this event triggers a "Truck Dumping Event," otherwise, the truck joins the queue of trucks.

Truck Dumping Event: This event represents the truck delivering the material to its destination, to a waste dump or a processing plant, for example. The time to dump is stochastic, and after the event is resolved, a "Truck Moving Event" is triggered to send the truck back to be loaded. Here, a new decision can be made, sending the truck to a different shovel. Similar to the "Shovel Loading Event," once this event is finished, it can trigger itself if another truck is in the queue waiting for dumping.

Truck Breaking Event: Represents a truck stopping its activities due to maintenance or small failures. In this event, a truck is removed from the DES regardless of its current assignment. No action can be performed until it is fixed and can be returned to the operation.

Shovel Breaking Event: Represents the shovel becoming inaccessible for a certain period due to small failures or maintenance. No material is extracted during this period, and no trucks are sent to this location, being re-routed until the equipment is ready to be operational again.

Figure 3a shows a diagram illustrating a possible sequence of events that can be triggered. In the figure, the solid lines represent the events triggered immediately after the end of a particular event. The dashed lines are related to events that can be triggered if trucks are waiting in the queue. To ensure the sequence respects a chronological ordering, a priority queue is maintained, where each event is ranked by its starting time, as illustrated in Figure 3b.



T0, Shovel Loading
T1, Truck Moving
T2, Truck Dumping
T3, Truck Moving
T4, Shovel Loading
(…)

Figure 3. Discrete event simulation represented in terms of: (**a**) an initial event and possible next events that can be triggered; (**b**) a priority queue that ranks each event by its starting time.

The DES starts with all the trucks being positioned at their respective shovel. This configuration triggers a "Shovel Loading Event," and the DES simulates the subsequent events and how much material flows from the extraction point to their destinations by the trucks. Once the truck dumps, a new decision is taken according to the DDQN policy. The DES proceeds by simulating the subsequent operations triggered by this assignment. This is repeated until the predefined time horizon, which represents N_{days} of simulated activities, is reached by the DES. All events that occur between the beginning and the end of the DES constitute an episode. Subsequent episodes start by re-positioning the trucks at their initial shovel allocation.

2.2. *Agent–Environment Interaction*

2.2.1. Definitions

The framework considers N_{trucks} trucks interacting with the DES. At every time step $t \in T$, after dumping the material into the adequate location, a new assignment for truck $i \in N_{trucks}$ is requested. The truck-agent i observes the current state $S_t^i \in S$, where S_t^i represents the perception of truck i on how the mining complex is performing at step t and takes an action $A_t^i \in A$, defining the next shovel to which the truck will be linked. The state S_t^i is a vector encoding all attributes relevant to characterize the current status of the mining complex. Figure 4 illustrates these attributes describing the state space, such as current queue sizes, current GPS location of trucks and shovels, and processing plant requirements.

This state information is encoded in a vector and inputted into the DDQN neural network, which outputs action-values, one for each shovel, representing the probability that the truck be dispatched to a shovel-dumping point path. A more detailed characterization of the state S_t^i is given in Appendix A.

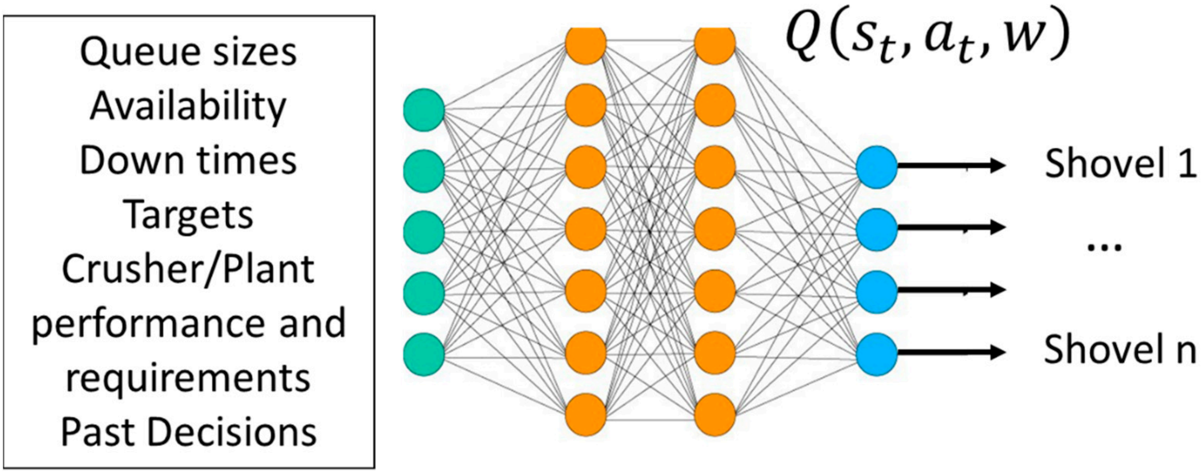


Figure 4. Illustration of the DDQN agent, which receives as input the state of the environment as input and outputs the desirability probability of choosing an action.

2.2.2. Reward Function

Once the agent outputs the action A_t^i, the DES emulates how the mining complex environment evolves by simulating, for example, new cycle times, the formation of queues, taking into consideration all other trucks in operation. The environment, then, replies to this agent's decision with a reward function, represented by Equation (1):

$$R_t^i = perc_t^i - pq_t^i \tag{1}$$

where $perc_t^i$ is the reward associated with delivering material to the mill and accomplishing a percentage of the destination's requirement (e.g., mill's daily target in tons/day). pq_t^i is the penalty associated with spending time in queues at both shovels and destinations. This term guides solutions with smaller queue formation while ensuring higher productivity.

In this multi-agent setting, each truck receives a reward R_t, which is the sum of each truck R_t^i, as shown in Equation (2), to ensure that all agents aim to maximize the same reward function.

$$R_t = \sum_{i}^{N_{trucks}} R_t^i \tag{2}$$

During each episode, the agent performs N_{steps} actions, the discounted sum of rewards is the called return presented by Equation (3):

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{N_{steps}-t-1} R_{N_{steps}} = \sum_{k=t+1}^{N_{steps}} \gamma^{k-t-1} R_k \tag{3}$$

where γ is a discounting factor parameter, which defines how much actions taken far in the future impact the objective function [15]. Equation (4) defines the objective, which is to obtain high-level control by training the agent to take improved actions so that the trucks can fulfil the production planning targets and minimize queue formation.

$$\max_{a \in A} E\left[G_t \middle| S = S_t^i, A = A_t^i\right] \tag{4}$$

The environment is characterized by uncertainties related to loading, moving, dumping times of the equipment, breakdowns of both trucks and shovels. This makes it very difficult to define all possible transition probabilities between states $\left(p\left(S_{t+1}^i \middle| S = S_t^i, A = A_t^i\right)\right)$ to obtain the expected value defined in Equation (4). Therefore, these transition probabilities are replaced by the Monte Carlo approach used in the form of the DES.

The framework allows for future actions to be rapidly taken since providing the input vector S_t to the neural network and outputting the corresponding action is a fast operation. This means that the speed at which the decisions can be made depends more on how quickly the attributes related to the state of the mining complex can be collected, which has been recently substantially improved with the new sensors installed throughout the operation.

2.3. Deep Double Q-Learning (DDQN)

The approach used in the current study is the double deep Q-learning (DDQN) approach based on the work of Hasselt et al. [50]. Q-function $\boldsymbol{Q_i}\left(S_t^i, A_t^i, w_t^i\right)$ is the action-value function, shown in Equation (5), which outputs values representing the likelihood of truck i choosing action A_t^i, given the encoded state S_t^i and the set of neural-network weights w_t^i, illustrated by Figure 4.

$$\boldsymbol{Q_i}\left(S_t^i, A_t^i, w_t^i\right) = E\left[G_t \middle| S = S_t^i, A = A_t^i\right] \tag{5}$$

Denote $Q_i^*\left(s_t^i, a_t^i, w^i\right)$ to be the theoretical optimal action-value function. Equation (6) presents the optimal policy $\pi^*\left(S_t^i\right)$ for the state S_t^i, which is obtained by using the action-function greedily:

$$\pi^*\left(S_t^i\right) = \underset{a' \in A}{\operatorname{argmax}} Q_i^*\left(S_t^i, a', w^i\right) \tag{6}$$

Note that, using Equation (6), the approach directly maximizes the reward function described in Equation (4). This is accomplished by updating the $\boldsymbol{Q_i}\left(S_t^i, A_t^i, w_t^i\right)$ function to approximate the optimal action-value function $\left(\boldsymbol{Q_i}\left(S_t^i, A_t^i, w_t^i\right) \to Q_i^*\left(S_t^i, A_t^i, w^i\right)\right)$.

By letting agent i interact with the environment, given the state S_t^i, the agent chooses A_t^i, following a current dispatching policy $\pi_i\left(S_t^i\right) = \underset{a' \in A}{\operatorname{argmax}} \boldsymbol{Q_i}\left(S_t^i, a', w_t^i\right)$, the environment then returns the reward R_t and a next state S_{t+1}^i. The sample experience $e_k^i = \left(S_t^i, A_t^i, R_t, S_{t+1}^i\right)$ is stored in a memory buffer, $D_K^i = \left\{e_1^i, e_2^i, \ldots, e_K^i\right\}$, which is increased as the agent interacts with the environment for additional episodes. A maximum size limits this buffer, and once it is reached, the new sample e_k^i replaces the oldest one. This is a known strategy called experience replay, which helps stabilize the learning process [48,50,64].

In the beginning, $\boldsymbol{Q_i}\left(S_t^i, A_t^i, w_t^i\right)$ is randomly initialized, then a memory tuple e_k^i is repeatedly uniformly sampled from the memory buffer D_K^i, and the related $e_k^i = \left(S_t^i, A_t^i, R_t, S_{t+1}^i\right)$ values are used to estimate the expected future return $\overline{G}_t$, as shown in Equation (7):

$$\overline{G}_t = \begin{cases} R_t, & if\ episode\ terminates\ at\ t+1 \\ R_t + \gamma \overline{\boldsymbol{Q}}_{\boldsymbol{i}}\left(S_t^i,\ \underset{a' \in A}{\operatorname{argmax}}\ \boldsymbol{Q_i}\left(S_{t+1}^i, a', w_t^i\right),\ \overline{w}^i\right), & otherwise \end{cases} \tag{7}$$

Additionally, gradient descent is performed on $\left(\overline{G}_t - \boldsymbol{Q_i}\left(S_t^i, A_t^i, w_t^i\right)\right)^2$ with respect to the parameter weights w_t^i. Note that a different Q-function, $\overline{\boldsymbol{Q}}_{\boldsymbol{i}}(\cdot)$, is used to predict the future reward; this is simply the $\boldsymbol{Q_i}(\cdot)$ with the old weight parameters. Such an approach is also used to stabilize the agent's learning, as noisy environments can result in a slow learning process [50]. After N_{Updt} steps, the weights w_t^i are copied to $\overline{w}^i$, as follows: $\overline{\boldsymbol{Q}}_{\boldsymbol{i}}(\cdot) = \boldsymbol{Q_i}(\cdot)$.

During training, the agent i follows the greedy policy $\pi_i(S_t^i)$ meaning that it acts greedily with respect to its current knowledge. If gradient descent is performed with samples coming solely from $\pi_i(S_t^i)$, the method inevitably would reach a local maximum very soon. Thus, to avoid being trapped in a local maximum, in ϵ% of the time, the agent takes random actions exploring the solution space, sampling it from a uniform distribution $A_t^i \sim U(A)$. In $(100 - \epsilon)$% of the time, the agent follows the current policy $A_t^i \sim \pi_i(S_t^i)$. To take advantage of long-term gains, after every N_{steps_reduce} steps this value is reduced by a factor $reduce\_factor \in [0,1]$. In summary, the algorithm is presented as follows:

Algorithm 1 Proposed learning algorithm.

```
Initialize the action-functions. $\boldsymbol{Q_i}(\cdot)$ and $\overline{\boldsymbol{Q}}_{\boldsymbol{i}}(\cdot)$ by assigning initial weights to $w_t^i$ and $\overline{w}^i$.
Set $n1_{counter} = 0$ and $n2_{counter} = 0$.
Initialize the DES, with the trucks at their initial locations (e.g., queueing them at the shovel).
Repeat for each episode:
    Given the current truck-shovel allocation, the DES simulates the supply material
being transferred from mining facies to the processors or waste dump by the trucks.
    Once the truck $i$ dumps the material, a new allocation must be provided.
    At this point, the agent collects the information about the state $S_t^i$.
    Sample $u \sim U(0,100)$
    If $u < \epsilon\%$
        The truck-agent $i$ acts randomly $A_t^i \sim U(A)$
    Else:
        The truck-agent $i$ acts greedily $A_t^i \sim \pi_i\left(S_t^i\right)$
    Taking action $A_t^i$, observe $R_t$ and a new state $S_{t+1}^i$.
    Store the tuple $e_k^i = \left(S_t^i,\ A_t^i,\ R_t,\ S_{t+1}^i\right)$ in the memory buffer $D_K^i$
    Sample a batch of experiences $e_k^i = \left(S_t^i,\ A_t^i,\ R_t,\ S_{t+1}^i\right)$, of size $batch\_size$, from $D_K^i$:
    For each transition sampled, calculate the respective $\overline{G}_t$ from Equation (7).
    Perform gradient descent on $\left(\boldsymbol{Q_1^i}\left(s_{t+1}, a', w_1^i\right) - \overline{G}_t\right)^2$ according to Equation (8):

$w_{1,next}^i \leftarrow w_{1,old}^i - \alpha\left(\boldsymbol{Q_1^i}\left(s_{t+1}, a', w_1^i\right) - G_t\right)\nabla_{w_1^i}\boldsymbol{Q_1^i}\left(s_{t+1}, a', w_1^i\right)$    (8)
    $n1_{counter} \leftarrow n1_{counter} + 1$.
    $n2_{counter} \leftarrow n2_{counter} + 1$.
    If $n1_{counter} \geq N_{Updt}$:
        $\overline{w}^i \leftarrow w_t^i$.
        $n1_{counter} \leftarrow 0$.
    If $n2_{counter} \geq N_{step\_reduce}$:
        $\epsilon \leftarrow \epsilon * reduce\_factor$.
        $n2_{counter} \leftarrow 0$.
```

3. Case Study at a Copper—Gold Mining Complex

3.1. Description and Implementation Aspects

The proposed framework is implemented at a copper–gold mining complex, summarized in Figure 5. The mining complex comprises two open-pits, whose supply material is extracted by four shovels and transported by twelve trucks to the appropriate destinations: waste dump, mill or leach pad. Table 1 presents information regarding the mining equipment and processors. The shovels are placed at the mining facies following pre-defined extraction sequences, where the destination of each block was also pre-established beforehand. The mining complex shares the truck fleet between pits A and B. The waste dump receives waste material from both mines, whereas the leach pad material only processes supply material from pit B due to mineralogical characteristics. The truck going to the leach pad dumps the material into a crusher, then transported it to the leach pad. Regarding the milling material, each pit is associated with a crusher, and the trucks haul the high-grade material extracted from a pit and deliver it to the corresponding crusher. Next, a conveyor belt transfers this material to the mill combining the material from the two sources. Both the mill and the leach pad are responsible for producing copper products and gold ounces to be sold.

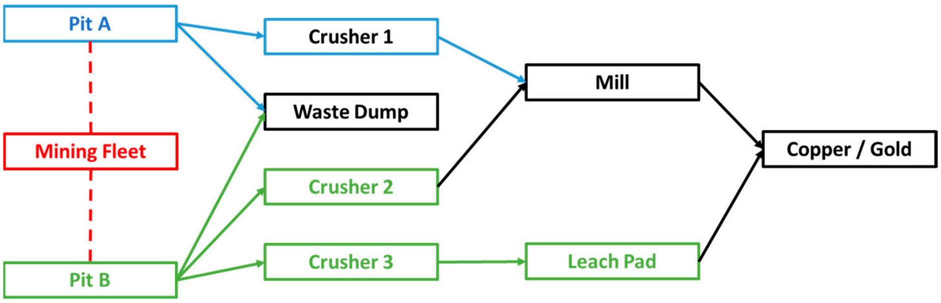


Figure 5. Diagram of the mining complex.

Table 1. Mining complex equipment and processors.

Equipment	Description
Trucks 12 in total	6 of payload capacity of 200 tons 3 of payload capacity of 150 tons 3 of payload capacity of 250 tons
Shovel 4 in total	2 of bucket payload of 80 tons 1 of bucket payload of 60 tons 1 of bucket payload of 85 tons
Mill	Capacity 80,000 ton/day, with 2 crushers.
Leach Pad	Capacity 20,000 ton/day, with one crusher.
Waste Dump	1 Waste Dump with no limitation on capacity.

The discrete event simulation, described in Section 2.1, emulates the loading, hauling and dumping operations in the mining complex. Each event is governed by uncertainties that impact the truck cycling times. Table 2 presents distributions used for the related uncertainty characterization. For simplicity, these stochastic distributions are approximated from historical data; however, a more interesting approach would have been to use the distribution directly from historical data. When the truck dumps material into a destination, a new dispatching decision must be taken by the DDQN dispatching policy. This generates samples that are used to train the DDQN dispatching policy. During the training phase, each episode lasts the equivalent of 3 days of continuous production, where the truck-agent interacts with the discrete event mine simulator environment, taking actions and collecting rewards. In total, the computational time needed for training, for the present case study, is around 4 h. For the comparison (testing) phase, the method was exposed to five consecutive days of production. This acts as a validation step, ensuring that the agents observe the mining complex's configurations which were unseen during training. The results presented show the five days of production, and the performance obtained illustrates that the method does not overfit regarding the three days of operation but maintain a consistent strategy for the additional days.

Table 2. Definition of stochastic variables considered in the mining complex.

Stochastic Variable	Probability Distribution
Loaded truck speed (km/h)	Normal (17, 4)
Empty truck speed (km/h)	Normal (35, 6)
Dumping + maneuver time (min)	Normal (1, 0.15)
Shovel bucketing load time (min)	Normal (1.1, 0.2)
Truck mean time between failures (h)	Poisson (36)
Truck mean time to repair (h)	Poisson (5)
Shovel mean time between failures (h)	Poisson (42)
Shovel mean time to repair (h)	Poisson (4)

Note that although the DDQN policy provides dispatching decisions considering a different context from the one it was trained, the new situations cannot be totally different. It is assumed that in new situations, the DDQN experiences are sampled from the same distribution observed during training. In the case where the sequence of extraction changes considerably and new mining areas as well as other destinations are prioritized, the model needs to be retrained.

Two baselines are presented to compare the performance of the proposed approach. The first one, referred to as fixed policy, is a strategy that continually dispatches the truck to the same shovel path throughout the episode. The performance comparison between the DDQN and fixed policy is denoted Case 1. The second approach, referred to as greedy policy, sends trucks to needy shovels with the shortest waiting times to decrease idle shovel time, denoted Case 2. Both cases start with the same initial placement of the trucks.

The environment is stochastic, in the sense that testing the same policy for multiple episodes generates different results. Therefore, for the results presented here, episodes of 5 days of continuous production are repeated 10 times for each dispatching policy. To assess uncertainty outcomes beyond the ones arising from operational aspects, geological uncertainty is also included in the assessment by considering 10 orebody simulations (Boucher and Dimitrakopoulos; 2009) characterizing the spatial uncertainty and variability of copper and gold grades in the mineral deposit. The graphs display results in P10, P50 and P90 percentile, corresponding to the probability of 10, 50 and 90%, respectively, of being below the value presented.

3.2. Results and Comparisons

Figure 6 presents the daily throughput obtained by running the DES over the five days of production, which is achieved by accumulating all material processed by the mill within each day. Note that here the P10, P50 and P90 are only due the equipment uncertainty. Overall, the proposed model delivers more material to the mill when compared to both cases. The DDQN method adapts the dispatching to move trucks around, relocating them to the shovels that are more in need, which constantly results in higher throughput.

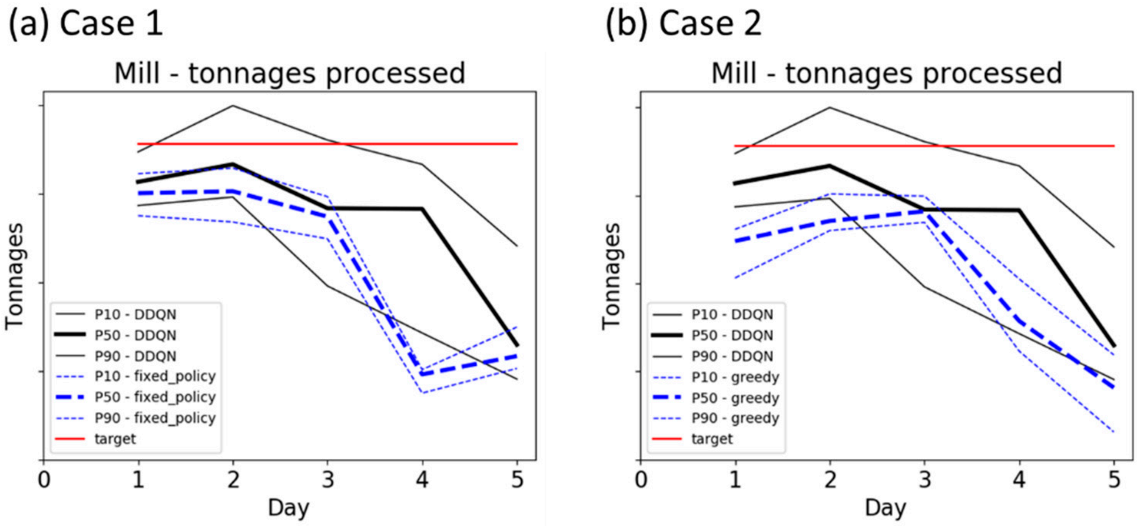


Figure 6. Daily throughput at the mill compared the DDQN policy (black line) and the respective baselines (blue line): (**a**) fixed policy and (**b**) greedy policy.

The throughput in day five drops compared to previous days, mostly due to a smaller availability of trucks as the DES considers failures in the trucks; Figure 7 presents the average number of trucks available per day. During the initial three days, the availability of trucks hovers between 10 and 12 trucks, but this rate drops in the last 2 days, which decreases the production. However, the trained policy can still provide a higher feed rate at the mill, even in this adversity. The availability of trucks on days 4 and 5 is smaller than the period for which the DDQN based method was trained, which shows an adapting capability of the dispatching approach.

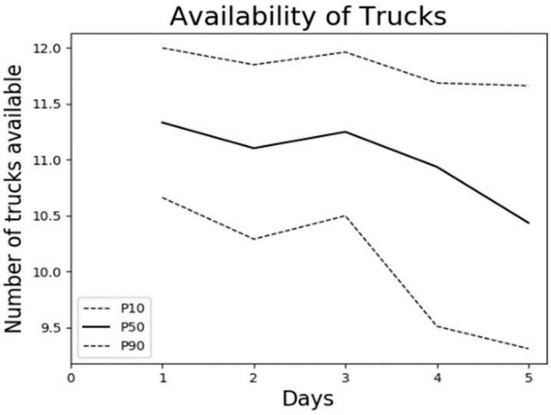


Figure 7. Availability of trucks during the five days of operation.

The framework is also efficient in avoiding queue formation. Figure 8 presents the average queue sizes at the mill and the sulphide leach. The queue at different locations is recorded hourly and averaged over each day. The plot shows that, for most of the days, the proposed approach generates smaller queues. Combined with the higher throughput obtained, this reduction in queue sizes demonstrates better fleet management. For example, during the initial three days, the DDQN approach improves the dispatching strategy by forming smaller queues at the mill. At the same time, the amount of material being delivered is continuously higher. On the 4th day, the proposed approach generates a larger queue size at the mill, which is compensated by having considerably higher throughput at this location.

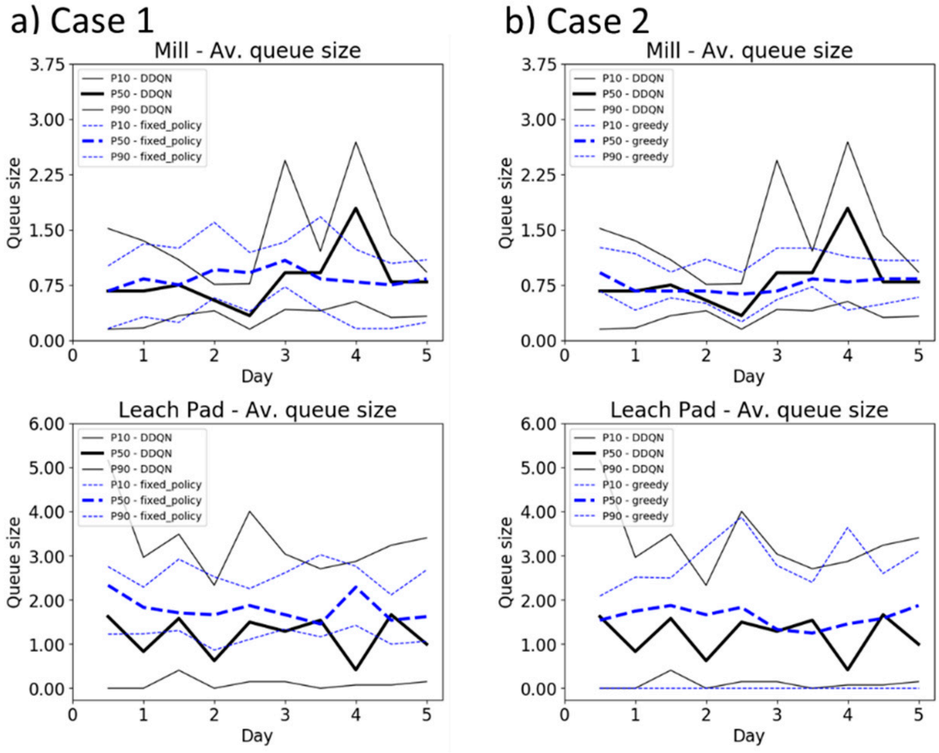


Figure 8. Queue sizes of trucks waiting at the mill (top) and Sulphide Leach (bottom) for the Deep DQN policy (black line) and the respective baseline (blue line): (**a**) fixed policy and (**b**) greedy policy.

Figure 9 displays the cumulative total copper recovered at the mining complex over the five days. Interestingly, during the first three days of DES simulation, corresponding to the training period of the DDQN approach, the total recovered copper profile between the proposed method and the baselines is similar. However, this difference is more pronounced over the last two days, which represents the situation that the trained method has not seen. This results in 16% more copper recovered than the fixed policy and 12% more than the greedy strategy. This difference in results is even larger when the total gold recovered is compared. The DDQN method generates a 20 and 23% higher gold profile in Case 1 and Case 2, respectively, Figure 10.

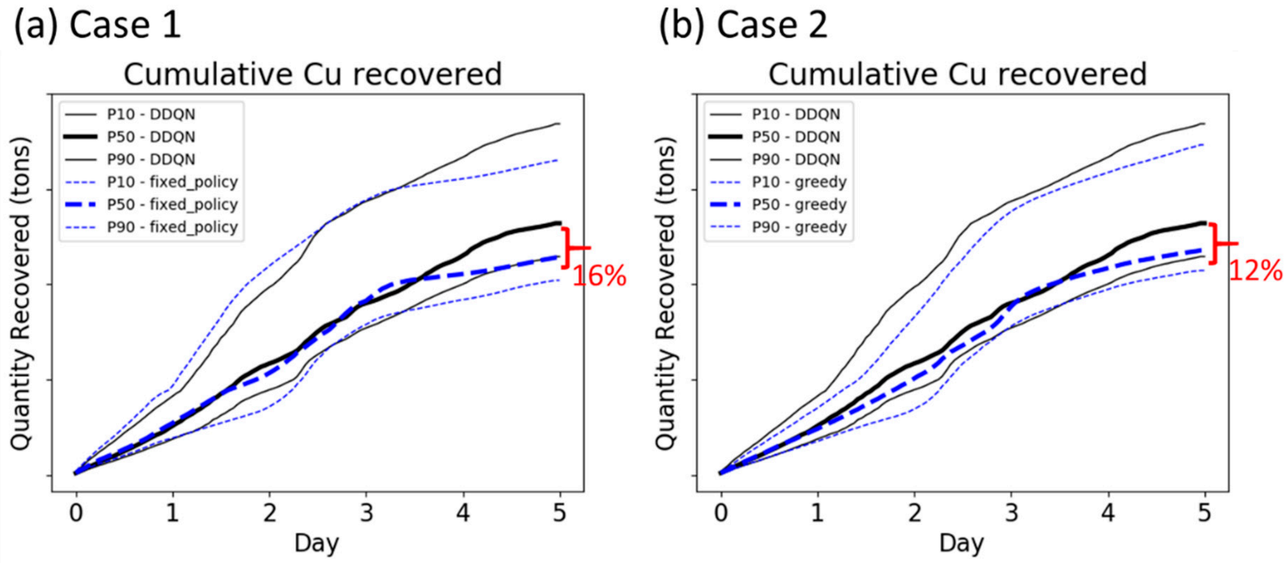


Figure 9. Cumulative copper recovered for the optimized DDQN policy (black line) and the respective baseline (blue line): (**a**) Case 1 and (**b**) Case 2.

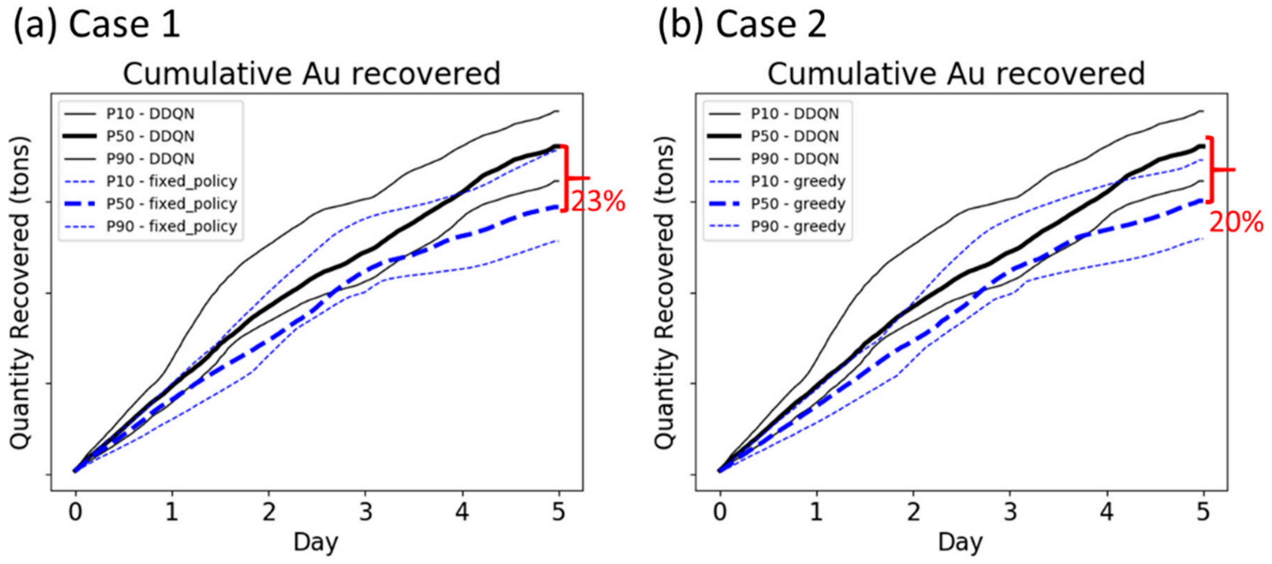


Figure 10. Cumulative gold recovered for the DDQN policy (black line) and the respective baseline (blue line): (**a**) fixed policy and (**b**) greedy policy.

4. Conclusions

This paper presents a new multi-agent truck-dispatching framework based on a reinforcement learning framework. The approach involves the interaction between a DES, simulating the operational events in a mining complex, and a truck-dispatching policy based on the DDQN method. Given a pre-defined schedule in terms of the sequence of extraction and destination policies for the mining blocks, the method improves the real-time truck-dispatching performance. The DES mimics daily operations, including loading, transportation and dumping, and equipment failures. A truck delivers the material to a processor or waste dump, and the truck-dispatcher provides it with a different shovel path. At this point, the truck receives information about the mining complex, such as other truck locations via GPS tracking, the amount of material feeding the processing plant and queue sizes at different locations. This state information is encoded into a vector, characterizing the state of the mining complex. This vector is inputted into the DDQN neural network, which outputs action values, describing the likelihood to send the truck to each shovel. Each dispatching decision yields a reward, which is received by the agent, as a performance evaluation. Initially, the truck-agent acts randomly; as the agent experiences many situations during training, the dispatching policy is improved. Thus, when new dispatching decisions are requested, an assignment is quickly obtained by the output of the DDQN agent. It differs from previous methods that solve a different optimization repeatedly during dispatching. Instead, the only requirement is to collect information regarding the state of the mining complex. With the digitalization of the mines, obtaining the required information can be done quickly.

The method is applied to a copper–gold mining complex composed of two pits, three crushers, one waste dump, one mill and one leach-pad processing stream. The fleet is composed of four shovels, and twelve trucks that can travel between the two pits. The DDQN-based method is trained for the equivalent of three days, while the results are presented for five days of production. Two dispatching baseline policies are used for comparison to assess the capabilities of the proposed method: fixed truck-shovel allocations and a greedy approach that dispatches trucks to needy shovels with the smallest queue sizes. The results show that the DDQN-based method provides the mill processing stream with higher throughput while generating shorter queues at different destinations, which shows a better fleet utilization. Over the five days of production, the proposed policy produces 12 to 16% more copper and 20 to 23% more gold than the baseline policies. Overall, the reinforcement learning approach has shown to be effective in training truck-dispatching agents, improving real-time decision-making. However, future work needs explore the development of new approaches that address the impact and adaptation of truck-dispatching decisions to changes and re-optimization of short-term extraction sequences given to the acquisition of new information in real-time and uncertainty in the properties of the materials mind.

Author Contributions: Conceptualization, J.P.d.C. and R.D.; methodology, J.P.d.C. and R.D.; software, J.P.d.C.; validation, J.P.d.C. and R.D; formal analysis, J.P.d.C. and R.D; investigation, J.P.d.C.; resources, R.D.; data curation, J.P.d.C.; writing—original draft preparation, J.P.d.C. and R.D.; writing—review and editing, R.D.; visualization, J.P.d.C.; supervision, R.D.; project administration, R.D.; funding acquisition, R.D. Both authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the National Science and Engineering Research Council of Canada (NSERC) CRD Grant CRDPJ 500414-16, NSERC Discovery Grant 239019, and the COSMO mining industry consortium (AngloGold Ashanti, AngloAmerican, BHP, De Beers, IAMGOLD, Kinross, Newmont Mining and Vale).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A.

Appendix A.1. State Definition

The definition of the state of the mining complex vector S_t^i encodes all attributes relevant to characterize the current status of the mining complex. Table A1 presents where the attributes are taken from and how it is represented in a vector format. Note that the encoding used here simply transforms the continuous attributes into values between 0 and 1, by a division of a large number. For discrete ones, a one-hot-encoding approach is used, where the number of categories defines the size of the vector, and a value of 1 is placed in the location corresponding to the actual category. This strategy attempts to avoid generating large gradients during gradient descent and facilitates the learning process. This idea can be further generalized, and other attributes judged relevant by the user can also be included.

Table A1. Attributes defining the current state of the mining complex.

Attribute in Consideration		Representation
Shovel related attributes	Destination policy of the block being currently extracted	1-hot-encoded vector (3-dimensional)
	Destination policy of next 2 blocks	1-hot-encoded (6 dimensional in total)
	Shovel capacity	1 value divided by the largest capacity
	Variable indicating if the shovel is currently in maintenance	1 value (0 or 1)
	Current distance to destination	1 value divided by a large number
	Number of trucks associated	1 value divided by a large number
	Approximated queue sizes	1 value divided by a large number
	Approximated waiting times	1 value divided by a large
Number of attributes per shovel		15
Destination related attributes	% target processed	1 value
	Amount of material received at crushers	2 values divided by a large number
	Distance to each shovel	4 values dived by a large number
	Approximated queue sizes	1 value divided by a large number
	Approximated waiting times	1 value divided by a large number
Number of attributes per destination		9
Truck related attributes	Truck capacity	1 value divided by the largest capacity
	Current number of trucks currently in operation.	1 value divided by the total number of trucks
	The last shovel visited	1-hot-encoded (4 values)
Number of attributes of each truck		
Total of attributes		102

Appendix A.2. Neural Network Parameters

Table A2. Reinforcement learning parameters.

Neural Network	Input layer = 102 nodes with ReLU activation function;
	Hidden layer 306 nodes with ReLU activation function;
	Output layer: 4 nodes without activation function.

Gradient descent	Adam optimization, with learning rate = 2×10^{-4}.
DDQN parameters	$\gamma = 0.99$ $\epsilon = 0.25$, with $reduce\_factor = 0.98$ 10,000 episodes of training.

References

1. Chaowasakoo, P.; Seppälä, H.; Koivo, H.; Zhou, Q. Digitalization of Mine Operations: Scenarios to Benefit in Real-Time Truck Dispatching. *Int. J. Min. Sci. Technol.* **2017**, *27*, 229–236. [CrossRef]
2. Alarie, S.; Gamache, M. Overview of Solution Strategies Used in Truck Dispatching Systems for Open Pit Mines. *Int. J. Surf. Min. Reclam. Environ.* **2002**, *16*, 59–76. [CrossRef]
3. Munirathinarn, M.; Yingling, J.C. A Review of Computer-Based Truck Dispatching Strategies for Surface Mining Operations. *Int. J. Surf. Min. Reclam. Environ.* **1994**, *8*, 1–15. [CrossRef]
4. Niemann-Delius, C.; Fedurek, B. Computer-Aided Simulation of Loading and Transportation in Medium and Small Scale Surface Mines. In *Mine Planning and Equipment Selection 2004*; Hardygora, M., Paszkowska, G., Sikora, M., Eds.; CRC Press: London, UK, 2004; pp. 579–584.
5. Blom, M.; Pearce, A.R.; Stuckey, P.J. Short-Term Planning for Open Pit Mines: A Review. *Int. J. Min. Reclam. Environ.* **2018**, *0930*, 1–22. [CrossRef]
6. Afrapoli, A.M.; Askari-Nasab, H. Mining Fleet Management Systems: A Review of Models and Algorithms. *Int. J. Min. Reclam. Environ.* **2019**, *33*, 42–60. [CrossRef]
7. Temeng, V.A.; Otuonye, F.O.; Frendewey, J.O. Real-Time Truck Dispatching Using a Transportation Algorithm. *Int. J. Surf. Min. Reclam. Environ.* **1997**, *11*, 203–207. [CrossRef]
8. Li, Z. A Methodology for the Optimum Control of Shovel and Truck Operations in Open-Pit Mining. *Min. Sci. Technol.* **1990**, *10*, 337–340. [CrossRef]
9. Ta, C.H.; Kresta, J.V.; Forbes, J.F.; Marquez, H.J. A Stochastic Optimization Approach to Mine Truck Allocation. *Int. J. Surf. Min. Reclam. Environ.* **2005**, *19*, 162–175. [CrossRef]
10. Upadhyay, S.P.; Askari-Nasab, H. Truck-Shovel Allocation Optimisation: A Goal Programming Approach. *Min. Technol.* **2016**, *125*, 82–92. [CrossRef]
11. Afrapoli, A.M.; Tabesh, M.; Askari-Nasab, H. A Transportation Problem-Based Stochastic Integer Programming Model to Dispatch Surface Mining Trucks under Uncertainty. In Proceedings of the 27th International Symposium on Mine Planning and Equipment Selection—MPES 2018; Springer: Berlin/Heidelberg, Germany, 2019; pp. 255–264. Available online: https://www.springerprofessional.de/en/a-transportation-problem-based-stochastic-integer-programming-mo/16498454 (accessed on 15 May 2021).
12. White, J.W.; Olson, J.P. Computer-Based Dispatching in Mines with Concurrent Operating Objectives. *Min. Eng.* **1986**, *38*, 1045–1054.
13. Soumis, F.; Ethier, J.; Elbrond, J. Truck Dispatching in an Open Pit Mine. *Int. J. Surf. Min. Reclam. Environ.* **1989**, *3*, 115–119. [CrossRef]
14. Elbrond, J.; Soumis, F. Towards Integrated Production Planning and Truck Dispatching in Open Pit Mines. *Int. J. Surf. Min. Reclam. Environ.* **1987**, *1*, 1–6. [CrossRef]
15. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, UK, 2018.
16. Del Castillo, M.F.; Dimitrakopoulos, R. Dynamically Optimizing the Strategic Plan of Mining Complexes under Supply Uncertainty. *Resour. Policy* **2019**, *60*, 83–93. [CrossRef]
17. Montiel, L.; Dimitrakopoulos, R. Simultaneous Stochastic Optimization of Production Scheduling at Twin Creeks Mining Complex, Nevada. *Min. Eng.* **2018**, *70*, 12–20. [CrossRef]
18. Goodfellow, R.; Dimitrakopoulos, R. Global Optimization of Open Pit Mining Complexes with Uncertainty. *Appl. Soft Comput. J.* **2016**, *40*, 292–304. [CrossRef]
19. Pimentel, B.S.; Mateus, G.R.; Almeida, F.A. Mathematical Models for Optimizing the Global Mining Supply Chain. In *Intelligent Systems in Operations: Methods, Models and Applications in the Supply Chain*; Nag, B., Ed.; IGI Global: Hershey, PA, USA, 2010; pp. 133–163.
20. Levinson, Z.; Dimitrakopoulos, R. Adaptive Simultaneous Stochastic Optimization of a Gold Mining Complex: A Case Study. *J. S. African Inst. Min. Metall.* **2020**, *120*, 221–232. [CrossRef] [PubMed]
21. Saliba, Z.; Dimitrakopoulos, R. Simultaneous stochastic optimization of an open pit gold mining complex with supply and market uncertainty. *Min. Technol.* **2019**, *128*, 216–229. [CrossRef]
22. Both, C.; Dimitrakopoulos, R. Joint Stochastic Short-Term Production Scheduling and Fleet Management Optimization for Mining Complexes. *Optim. Eng.* **2020**. [CrossRef]
23. Whittle, J. The Global Optimiser Works—What Next? In *Advances in Applied Strategic Mine Planning*; Dimitrakopoulos, R., Ed.; Springer International Publishing: Cham, Switzerland, 2018; pp. 31–37.
24. Whittle, G. Global asset optimization. In *Orebody Modelling and Strategic Mine Planning: Uncertainty and Risk Management Models*; Dimitrakopoulos, R., Ed.; Society of Mining: Carlton, Australia, 2007; pp. 331–336.
25. Hoerger, S.; Hoffman, L.; Seymour, F. Mine Planning at Newmont's Nevada Operations. *Min. Eng.* **1999**, *51*, 26–30.
26. Chanda, E. *Network Linear Programming Optimisation of an Integrated Mining and Metallurgical Complex. Orebody Modelling and Strategic Mine Planning*; Dimitrakopoulos, R., Ed.; AusIMM: Carlton, Australia, 2007; pp. 149–155.

27. Stone, P.; Froyland, G.; Menabde, M.; Law, B.; Pasyar, R.; Monkhouse, P. Blasor–Blended Iron Ore Mine Planning Optimization at Yandi, Western Australia. In *Orebody Modelling and Strategic Mine Planning: Uncertainty and Risk Management Models*; Dimitrakopoulos, R., Ed.; Spectrum Series 14; AusIMM: Carlton, Australia, 2007; Volume 14, pp. 133–136.
28. Sitek, P.; Wikarek, J.; Rutczyńska-Wdowiak, K. Capacitated Vehicle Routing Problem with Pick-Up, Alternative Delivery and Time Windows (CVRPPADTW): A Hybrid Approach. In *Distributed Computing and Artificial Intelligence, Proceedings of the 16th International Conference, Special Sessions, Avila, Spain, 26–28 June 2019*; Springer International Publishing: Cham, Switzerland, 2019.
29. Gola, A.; Kłosowski, G. Development of Computer-Controlled Material Handling Model by Means of Fuzzy Logic and Genetic Algorithms. *Neurocomputing* **2019**, *338*, 381–392. [CrossRef]
30. Bocewicz, G.; Nielsen, P.; Banaszak, Z. Declarative Modeling of a Milk-Run Vehicle Routing Problem for Split and Merge Supply Streams Scheduling. In *Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology—ISAT 2018*; Świątek, J., Borzemski, L., Wilimowska, Z., Eds.; Springer: Cham, Switzerland, 2019; pp. 157–172.
31. Pillac, V.; Gendreau, M.; Guéret, C.; Medaglia, A.L. A Review of Dynamic Vehicle Routing Problems. *Eur. J. Oper. Res.* **2013**, *225*, 1–11. [CrossRef]
32. Gendreau, M.; Potvin, J.-Y. Dynamic Vehicle Routing and Dispatching. In *Fleet Management and Logistics*; Crainic, T.G., Laporte, G., Eds.; Springer US: Boston, MA, USA, 1998; pp. 115–126.
33. Secomandi, N.; Margot, F. Reoptimization Approaches for the Vehicle-Routing Problem with Stochastic Demands. *Oper. Res.* **2009**, *57*, 214–230. [CrossRef]
34. Azi, N.; Gendreau, M.; Potvin, J.-Y. A Dynamic Vehicle Routing Problem with Multiple Delivery Routes. *Ann. Oper. Res.* **2012**, *199*, 103–112. [CrossRef]
35. Torkamani, E.; Askari-Nasab, H. A Linkage of Truck-and-Shovel Operations to Short-Term Mine Plans Using Discrete-Event Simulation. *Int. J. Min. Miner. Eng.* **2015**, *6*, 97–118. [CrossRef]
36. Matamoros, M.E.V.; Dimitrakopoulos, R. Stochastic Short-Term Mine Production Schedule Accounting for Fleet Allocation, Operational Considerations and Blending Restrictions. *Eur. J. Oper. Res.* **2016**, *255*, 911–921. [CrossRef]
37. Quigley, M.; Dimitrakopoulos, R. Incorporating Geological and Equipment Performance Uncertainty While Optimising Short-Term Mine Production Schedules. *Int. J. Min. Reclam. Environ.* **2019**, *34*, 362–383. [CrossRef]
38. Bodon, P.; Fricke, C.; Sandeman, T.; Stanford, C. Combining Optimisation and Simulation to Model a Supply Chain from Pit to Port. *Adv. Appl. Strateg. Mine Plan.* **2018**, 251–267. [CrossRef]
39. Hashemi, A.S.; Sattarvand, J. Application of ARENA Simulation Software for Evaluation of Open Pit Mining Transportation Systems—A Case Study. In Proceedings of the 12th International Symposium Continuous Surface Mining—Aachen 2014; Niemann-Delius, C., Ed.; Springer: Cham, Switzerland, 2015; pp. 213–224. Available online: https://link.springer.com/chapter/10.1007/978-3-319-12301-1_20 (accessed on 15 May 2021).
40. Jaoua, A.; Riopel, D.; Gamache, M. A Framework for Realistic Microscopic Modelling of Surface Mining Transportation Systems. *Int. J. Min. Reclam. Environ.* **2009**, *23*, 51–75. [CrossRef]
41. Sturgul, J. *Discrete Simulation and Animation for Mining Engineers*; CRC Press: Boca Raton, FL, USA, 2015.
42. Upadhyay, S.P.; Askari-Nasab, H. Simulation and Optimization Approach for Uncertainty-Based Short-Term Planning in Open Pit Mines. *Int. J. Min. Sci. Technol.* **2018**, *28*, 153–166. [CrossRef]
43. Yuriy, G.; Vayenas, N. Discrete-Event Simulation of Mine Equipment Systems Combined with a Reliability Assessment Model Based on Genetic Algorithms. *Int. J. Min. Reclam. Environ.* **2008**, *22*, 70–83. [CrossRef]
44. Law, A.M.; Kelton, W.D. *Simulation Modeling and Analysis*; McGraw-Hill.: New York, NY, USA, 1982.
45. Jaoua, A.; Gamache, M.; Riopel, D. Specification of an Intelligent Simulation-Based Real Time Control Architecture: Application to Truck Control System. *Comput. Ind.* **2012**, *63*, 882–894. [CrossRef]
46. Chaowasakoo, P.; Seppälä, H.; Koivo, H.; Zhou, Q. Improving Fleet Management in Mines: The Benefit of Heterogeneous Match Factor. *Eur. J. Oper. Res.* **2017**, *261*, 1052–1065. [CrossRef]
47. Afrapoli, A.M.; Tabesh, M.; Askari-Nasab, H. A Multiple Objective Transportation Problem Approach to Dynamic Truck Dispatching in Surface Mines. *Eur. J. Oper. Res.* **2019**, *276*, 331–342. [CrossRef]
48. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
49. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.P.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Balcan, M.F., Weinberger, K.Q., Eds.; Volume 48, pp. 1928–1937.
50. Van Hasselt, H.; Guez, A.; Silver, D.; Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the 30th AAAI Conference, Shenzhen, China, 21 February 2016.
51. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the Game of Go without Human Knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef]
52. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering Atari, Go, Chess and Shogi by Planning with Learned Model. *Nature* **2020**, *588*, 604–609. [CrossRef]

53. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* **2016**, *529*, 484–489. [CrossRef]
54. Vinyals, O.; Ewalds, T.; Bartunov, S.; Georgiev, P.; Vezhnevets, A.S.; Yeo, M.; Makhzani, A.; Küttler, H.; Agapiou, J.; Schrittwieser, J.; et al. StarCraft II: A New Challenge for Reinforcement Learning.cs.LG. *arXiv* **2017**, arXiv:1708.04782.
55. Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nat. Cell Biol.* **2019**, *575*, 350–354. [CrossRef]
56. Paduraru, C.; Dimitrakopoulos, R. Responding to new information in a mining complex: Fast mechanisms using machine learning. *Min. Technol.* **2019**, *128*, 129–142. [CrossRef]
57. Kumar, A.; Dimitrakopoulos, R.; Maulen, M. Adaptive self-learning mechanisms for updating short-term production decisions in an industrial mining complex. *J. Intell. Manuf.* **2020**, *31*, 1795–1811. [CrossRef]
58. Kumar, A. Artificial Intelligence Algorithms for Real-Time Production Planning with Incoming New Information in Mining Complexes. Ph.D. Thesis, McGill University, Montréal, QC, Canada, 2020.
59. Goovaerts, P. *Geostatistics for Natural Resources Evaluation*; Applied Geostatistics Series; Oxford University Press: New York, NY, USA, 1997.
60. Remy, N.; Boucher, A.; Wu, J. *Applied Geostatistics with SGeMS: A User's Guide*; Cambridge University Press: Cambridge, UK, 2009; Volume 9780521514.
61. Rossi, M.E.; Deutsch, C.V. *Mineral Resource Estimation*; Springer: Dordt, The Netherlands, 2014.
62. Gómez-Hernández, J.J.; Srivastava, R.M. One Step at a Time: The Origins of Sequential Simulation and Beyond. *Math. Geol.* **2021**, *53*, 193–209. [CrossRef]
63. Minniakhmetov, I.; Dimitrakopoulos, R. High-Order Data-Driven Spatial Simulation of Categorical Variables. *Math. Geosci.* **2021**. [CrossRef]
64. Lin, L.-J. Reinforcement Learning for Robots Using Neural Networks. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1993.