# Data-Driven and Artificial Intelligence Approach to Dynamic Truck Fleet Dispatching and Shovel Allocation Planning in Open-Pit Mines

by

Roberto Noriega

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mining Engineering

Department of Civil and Environmental Engineering
University of Alberta

# ABSTRACT

*An open-pit mine is a highly dynamic environment where different equipment resources are allocated to mining areas to extract metal-bearing rock and waste, for pit development, following a set flow of activities. The material mined is then transported through the mine road network to different destinations like processing plants, waste dumps, or stockpiles. Open-pit short-term planning defines the mining sequence, destinations, and equipment allocation decisions over short periods of time to meet the production and pit development targets established by the long-term strategic plan. It is a process that relies on estimating equipment productivity and mining areas' rock properties for the decision-making process. Therefore, it incurs high uncertainties from the operating cycles of mining equipment and their breakdowns, the interaction of loading and haulage equipment, and imperfect geological information from the mining areas.*

*This research proposes a practical framework for robust open-pit short-term planning based on Deep Reinforcement Learning (DRL). DRL is the Machine Learning (ML) branch that deals with computational approaches to learning an optimal sequential decision-making policy in an uncertain environment. Recent advances in ML have allowed the development of DRL frameworks for sophisticated production control systems in different industrial settings that outperform current practices. However, applications of DRL within the mining industry are still scarce. This research aims to fill that gap by developing a novel DRL framework for open-pit short-term planning, including shovel allocation and truck dispatching decisions. The proposed framework starts with developing a discrete event simulation (DES) that serves as the open-pit mine production environment based on the equipment dispatch and mine planning database. The DES then serves as a platform that provides feedback on decisions made by the DRL algorithm. Afterwards, the DRL algorithm is* *designed to achieve targets aligned with the two problems considered in this research, shovel allocation planning and truck dispatching control. The details of the DRL design process include the choice of algorithm, the state and action representations, the design of the function approximator, i.e., the architecture of the deep neural network learning the decisions, and the design of the reward structure to achieve the desired goals.*

*The DRL framework proposed for the truck dispatch control in open-pit mines provides real-time truck assignments to their next loading and dumping locations to achieve mining, processing, and blending targets. A case study is presented in an iron ore deposit where the trained agent learns a robust dispatching policy to achieve the ore and waste mining targets and maintain the metal concentration of the ore feed to the processing plants within a desired range. The DRL framework proposed for the shovel allocation planning goal is to learn a robust shovel allocation strategy for the next production quarter, 3-month, to meet the tonnes per hour (TPH) production target to be delivered to the crusher feeds, by interacting with the production simulator. Also, the framework is tested in a case study of an iron ore open-pit mine where the shovel allocation agent successfully learns a strategy that consistently delivers the desired production target. This research is expected to contribute to transferring and adopting Artificial Intelligence and ML technologies within the mining industry.*

# PREFACE

This thesis is an original work by Roberto Noriega. All or parts of Chapters 2 to 4 have been published or have been accepted into the peer-review process in different journals. I have been responsible for developing the different models presented here, programming and structuring the codebase, and wrting and editing these papers. Dr. Yashar Pourrahimian is the supervisory author and was involved with guiding the project's concept formation and manuscript composition.

Chater 2 of this work has been published as Noriega R., Pourrahimian Y. (2022). A systematic review of artificial intelligence and data-driven approaches in strategic open-pit mine planning. Resources Policy, Vol 77.

Chapter 3 of this work has been submitted and is under review as R. Noriega, Y. Pourrahimian, and H. Askari-Nasab, (2023), Deep reinforcement learning basedreal-time open-pit truck dispatching system, Computers and Operations Research.

Chapter 4 of this work has been submitted and is under review as R. Noriega, Y. Pourrahimian (2023), Shovel allocation and scheduling for open pit mining using Deep Reinforcement Learning, International Journal of Mining, Reclamation and Environment.

*This Thesis is Proudly Dedicated To:*

*My parents and sisters: Roberto, Patricia, Andrea and Valeria, who have always been there for me*

*Amanda, for being there during this long journey offering her support and love*

# ACKNOWLEDGMENT

I would like to thank my supervisor, Dr. Yashar Pourrahimian, for providing support, insightful comments and revisions on my work throughout my time as a Ph.D. student. I have enjoyed working with him and the professional relationship developed over the years.

I would like to express my appreciation to my friends at the MOL and the MRDIL at the University of Alberta; Magreth, Nasib, Alireza, Dismas, Hongshuo, Sharokh, Firouz, Soroush and many more during the years; for making my time at the University of Alberta enjoyable.

This research or any other goal I have achieved in my life would not have been possible without the support of my parents and sisters.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

**Parameters**

| | |
|---|---|
| AI | Artificial Intelligence |
| DES | Discrete Event Simulation |
| DNN | Deep Neural Network |
| DQL | Deep Q-Learning |
| DDQL | Double Deep Q-Learning |
| DRL | Deep Reinforcement Learning |
| GHG | Greenhouse Gas |
| GP | Goal programming |
| IOPS | Intelligent Agent-Based Open-Pit Mine Planning |
| IoT | Internet of Thing |
| IP | Integer programming |
| KPI | Key Performance Indicator |
| LP | Linear Programming |
| MIP | Mixed Integer Programming |

| | |
|---|---|
| ML | Machine learning |
| MSE | Mean Square Error |
| NN | Neural Network |
| NPV | Net Present Value |
| OPSTP | Open-Pit Mine Short-Term Planning |
| RL | Reinforcement Learning |

# CHAPTER 1

# INTRODUCTION

*This chapter acts as an introductory section for the research thesis, offering an overview of the background, problem statement, a summary of pertinent literature, the objectives of the study, and its existing limitations. Additionally, it presents a detailed explanation of the research methodology employed and highlights the innovative contributions made by this work. Lastly, the organization of the thesis is outlined.*

## 1.1 Background

Open-pit mining is the most common method to extract raw materials and metals from the earth. In an open-pit mine, heavy trucks move ore, broken rock with enough metal concentration to be profitable, and waste rocks between mining faces within the pit, commonly serviced by loader equipment, and different destination streams such as processing plants, stockpiles, and waste dumps (Whittle, 2011). The planning and design of an open-pit mining project starts with assessing and quantifying geological resources to develop a resource block model, a discretization of the mineral resources into blocks assigned a set of estimated characteristics such as metal grade or density. Afterwards, with the establishment of economic, market and technical scenarios, the resource model is valued to establish profitable mineable reserves, for which the final open limit, a set of pit pushbacks and yearly production schedules are designed to achieve the long-term goals for the mining project. Short-term and operational plans are then devised to provide detailed pit development decisions, material destination policies and equipment allocation decisions to meet the established long-term vision at shorter time horizons, weekly or monthly plans, that consider a detailed representation of the mining production system (Hustrulid et al., 2013).

Materials handling and transportation activities can make up to 50% or more of operating costs in open-pit mining operations (Curry et al., 2014) efficient management and allocation of loading and hauling resources is key to maintaining a profitable mining operation. Equipment performance, however, is subject to great uncertainty due to equipment operating cycles and breakdowns, operator efficiency, and external factors such as weather or road conditions. This affects assumed productivity rates during the short-term planning and scheduling stage, often leading to hard-to-achieve plans requiring frequent updating to react to different issues as they appear to avoid production shortfalls (Soofastaei et al., 2016). A primary research focus in the short-term planning of open-pit mines has been the development of mathematical models for optimal equipment allocation and pit sequencing decision-making; however, due to the complexity of the pit production systems, and the stochasticity of its components, these models struggle to capture all the operational details and are limited to deterministic evaluations (Blom et al., 2019).

This research proposes a Deep Reinforcement Learning (DRL) approach for robust and adaptive open-pit short-term planning, specifically for haul truck fleet dispatching and shovel allocation planning, which are key productivity drivers in open-pit operations. Reinforcement Learning (RL) is a branch of Machine Learning (ML) that involves a computational approach to learning from interactions with an environment to maximize a goal (R. Sutton & Barto, 2018). DRL has seen an increased application to optimize different engineering systems, such as transportation, manufacturing, and heavy industries, providing a highly flexible data-driven production control framework (Panzer & Bender, 2021).

## 1.1 Statement of the problem

The open-pit mining production system is a highly dynamic environment that comprises the operation and coordination of multiple pieces of equipment of different types and capacities to achieve a production goal, usually delivering a certain amount of ore within a certain quality range to processing facilities to comply with the long-term plan (Newman et al., 2010). A major challenge in open-pit short-term planning is the high uncertainty arising from the dynamic interaction of different machines, operating cycle times, and failures, as well as geological uncertainties in the quality of the material mined. This often leads to hard-to-achieve plans at the operational stage due to mismatches in productivity and geological forecasts, requiring frequent efforts to update plans and resolve issues as they appear.

Commercial tools and academic research in open-pit short-term planning focus on developing mathematical programming frameworks, usually linear optimization models or similar heuristics, which require the formulation of large and complex models to capture the highly dynamic open-pit production environment (Blom et al., 2019). A major drawback of these approaches is the complexity in including operational uncertainties, which make an already intractable mathematical problem substantially more complex, with limited capabilities to consider a significant number of production scenarios.

Simulation models have been used extensively in mining to estimate the productivity of mining systems by using historical data to reproduce equipment behavior and interactions to forecast future performance (Raj et al., 2009). Simulation models provide an efficient approach to quantifying the different operational uncertainties and particularities of the day-to-day operations in open-pit mines. Therefore, researchers have proposed using simulation models as a platform for an optimization engine that provides robust and optimal short-term mine planning decisions (Shishvan & Benndorf, 2019; Upadhyay & Askari-Nasab, 2018a). However, current simulation-optimization efforts use linear optimization techniques to find the planning decisions, which struggle to efficiently model the dynamic environment of day-to-day mining operations as linear constraints leading to suboptimal decisions, inability to account for a wide range of production scenarios, and sources of uncertainty, and significant computation times which could render real-world use unfeasible.

DRL has appeared in recent years as a collection of methods that allows for effective and robust data-driven decisions in real-time for optimal and adaptive production systems. The principle behind DRL is that by interacting multiple times with an uncertain environment, and receiving feedback in the form of a scalar reward for each action taken, an optimal decision-making strategy can be learned to maximize the expected cumulative reward. These approaches have been adopted for industrial systems where simulations are already used to measure performance metrics, and DRL provides a paradigm to learn an optimal control strategy for machines or other actions that can alter the production environment (Bertolini et al., 2021; Panzer et al., 2021). However, applications to a mining production environment are still very scarce and limited.

A mining simulation model provides a rich environment from which an RL agent, the decision-maker, takes actions such as dispatching trucks along a particular path or allocating shovels to a given mining area, and observes the results receiving a reward or punishment in relation to the desired goal to achieve. At every step, the agent receives a representation of the state of the mining system provided by the simulation and, by continuous interaction, learns an optimal policy, a mapping from environment states to actions to maximize the expected total rewards obtained. The agent itself is a Deep Neural Network (DNN) which, thanks to recent advances in DL, enables efficient learning of complex and non-linear relations for optimal and robust decision making.

Two specific problems in open-pit tactical planning are addressed here: the dynamic truck dispatching in daily operations and the planning of shovel allocations to mining faces over a production-quarter time horizon.

## 1.2 Summary of Literature Review

### 1.2.1 Open-pit short-term production planning

Short-term planning differs from long-term applications by emphasizing operational-level decisions dealing with equipment and resource allocation over a shorter time scale on a monthly, weekly, or shift-by-shift basis, usually under the guidance of the long-term plan. At these shorter time scales, mine operations are modeled with greater detail, considering the available equipment and different tasks required to execute the long-term strategic vision of the mine.

Model formulations for the open-pit mine short-term planning (OPSTP) vary amongst researchers but commonly seek to minimize deviations from production targets, minimize operating costs or maximize Net Present Value (NPV), and include a more detailed mathematical representation of equipment interaction. Common formulations aim to obtain decisions on shovel allocations to mining areas and production scheduling of development and extraction activities such as drilling, blasting and preparing the working area. Mixed Integer Programming (MIP) methods predominate in the literature on open-pit short-term planning (Blom et al., 2019). Although the level of details incorporated varies amongst researchers, the short-term planning decisions usually involve the sequencing of mining blocks, or aggregations of blocks called cuts, from the mineral resource block model and the allocation of equipment and other resources to comply with the long-term strategic plan.

Eivazy & Askari-Nasab (2012) proposed a MIP formulation that generates a monthly mining extraction schedule accounting for multiple destinations, along with a horizontal directional mining scheme and selection of ramp to haul ore out of the pit. L'Heureux et al. (2013) incorporate drilling and blasting decisions, shovel allocations and movement between the mining areas to minimize the total cost of operations. Kozan & Liu (2015) also incorporate drilling and blasting activities into the planning optimization process to generate a feasible timetable for shift operations and seek to maximize throughput and minimize expected equipment idle time. Blom et al. (2016) present a multi-objective optimization approach to short-term planning across a range of over 40 objective measures. However, their implementation relies on the use of a sophisticated concurrent-rolling-horizon algorithm, also discussed by Blom, Pearce, & Stuckey (2016) to solve the resulting complex mathematical model efficiently. The main limitation with these approaches is the large complexity of the model required to represent the mine production environment, which then requires manually crafted heuristics to solve it and limit the application to purely deterministic cases. Both & Dimitrakopoulos (2020) integrate uncertainty in fleet production capacity by simulating production capacity scenarios based on the mining block location, truck cycle uncertainty, and metal uncertainty. The authors developed a simulated annealing approach to solve the problem, remarking it was impractical to solve via an exact solver like CPLEX.

More recent research efforts aim to combine discrete simulation with optimization engines to obtain operational schedules that explicitly account for equipment interaction within the mine layout. Integrating Discrete Event Simulation (DES) can allow more robust and data-driven schedules. Upadhyay & Askari-Nasab (2017) present a detailed discrete simulation of mining operations that uses CPLEX engine to obtain optimal shovel allocations to mining faces. They extend their approach in Upadhyay & Askari-Nasab (2018a) to optimize mining faces extraction sequences, truck and shovel allocations using a multi-objective optimization approach within the simulation engine. Shishvan & Benndorf (2019) propose a similar framework for simulation-optimizing operational decisions for a coal continuous mining system in Germany. The simulation captures the details of the excavation and dumping practices of the mining site, however in both cases the optimization engine reacts to the system's state at every point a decision is needed rather than learning the behavior to plan accordingly to the uncertainty of the system.

### 1.2.2 Open-pit truck fleet dispatching

A truck fleet dispatch system in open-pit mines assigns trucks to shovels operating at a mining face within the pit to transport the extracted payload to different destinations outside the pit. The goal of the dispatch system is to route trucks to successfully achieve the shift production targets at the different destinations and maximize the fleet's productivity by minimizing delays and idle times. Open-pit operations are highly dynamic environments where the material is mined from multiple faces to be dumped at multiple destinations based on a production schedule that sets different targets for each one. As a result, the fleet operating cycle of transportation, loading, and dumping are highly stochastic activities that lead to considerable uncertainties in estimating times or costs required for mathematical optimization modeling. In addition, the operations are constrained by limited loading and dumping capacities at each service point, leading to the formation of queues, and resulting in idle and non-productive times.

The problem is usually split into two optimization stages. The first, or upper stage, sets target flow rates and tonnes to move between mining faces and destinations to achieve the shift targets using average estimates of the shovel and truck productivities and operating cycle times. Afterwards, a second or lower stage implements a real-time truck dispatching algorithm that matches trucks with shovels whenever an assignment is needed. This multi-stage approach was proposed by White & Olson (1986) and Olson et al. (1993) and successfully commercialized under the name DISPATCH®, which has become the most widely used truck fleet dispatch system in mining operations. Their system, as described in the available literature, formulates the upper stage problem as a Linear Programming (LP) model to set the production target flows and the number of trucks required at each path for a fixed time horizon to meet targets, then implement a dynamic programming-based approach for the lower stage, real-time, truck assignment. However, the real-time truck dispatching heuristic uses average estimates of fleet cycle activities based on previously recorded data without accounting for the impact on the future truck cycles and system performance due to the assignment returned or the uncertainties in the system.

The upper-stage problem of setting target rates for the different paths between material sources and destinations in the mining system has received the most attention in the literature. LP and MIP approaches have been commonly proposed to solve this stage (Bonates & Lizotte, 1988; Gurgur, 2011; Mena et al., 2013; Mohtasham et al., 2021; Soumis et al., 1989; Ta et al., 2005; V. Temeng et al., 1998). These methods are, in general, developed to obtain the optimal number of truck trips or total tonnage to be moved between each path over a period to meet the given production targets, with different additional objectives considered explicitly in the objective function, such as maximizing shovel and trucks utilization and minimizing idle times or operating costs. The main challenge for the applicability of these approaches is the scalability of LP and MIP models to real-world scenarios with large and heterogeneous equipment fleets and stochastic operating cycles where decisions are needed in real-time. To deal with these issues, most methods assume a homogeneous equipment fleet and use average estimates to formulate and solve the problem. Therefore, since these methods model the highly dynamic open-pit environment as a deterministic system, their solutions could be rendered obsolete or suboptimal.

Readers are directed to Alarie & Gamache (2002) and Moradi Afrapoli & Askari-Nasab (2019) for a detailed review on the advantages and disadvantages of these methods.

Once path targets are set, the lower stage problem deals with the real-time dispatching of trucks. Early work was dominated by heuristic rules that modeled the decision-making process as solving one of the following allocation problems: 1-truck-for-N-shovels, M-trucks-for-1-shovel, or M-trucks-for-N-shovels, as described by Chaowasakoo et al. (2017b, 2017a). In the 1-truck-for-N-shovels strategy, whenever a truck needs a decision, the available shovels are ranked in order based on a certain metric indicating the need for a truck to keep up with the production targets, and the truck is sent to the shovel at the top of this ranking. These approaches consider one truck at a time without measuring the decision's impact on the system's future state and can be considered a purely greedy solution. The M-trucks-for-1-shovel strategy follows a similar logic but considers one shovel at a time. Shovels are ranked based on a metric indicating how far they are behind production targets; trucks are ranked based on their expected time to complete their current cycles, and the next M trucks that satisfy the production shortfall are assigned to that shovel. These approaches are combined in the M-trucks-for-N-shovels to consider the fleet as a whole (sources). These heuristics, however, lack the foresight to gauge the impact of the allocation decision in the future states of the mine and cannot handle the stochasticity inherent in the mining fleet operation. Integer programming (IP) and MIP approaches for the real-time truck dispatching problem have been proposed as well (Moradi Afrapoli & Askari-Nasab, 2020; V. A. Temeng et al., 1997; Yeganejou et al., 2021). These usually aim to dispatch the next truck, or allocate the next M trucks, to minimize deviations from production targets and equipment idle times. Goal programming (GP) approaches provide an extension to these to account for multiple goals such as meeting production targets explicitly, minimizing shovel idle times, and minimizing truck waiting times (Mirzaei-nasirabad et al., 2023; Mohtasham et al., 2022; Moradi Afrapoli et al., 2019). These methods suffer from the same drawbacks discussed in the upper-stage model.

A major gap in the literature for truck dispatching in open-pit mining is the incorporation of uncertainties in decision-making optimization. Current approaches use historical data to calculate average equipment performance values to solve a simplified linear system. To overcome these drawbacks, a RL framework is proposed to train a neural network (NN) truck dispatching agent to meet shift production targets and minimize equipment idle times considering equipment performance uncertainty by learning on a stochastic simulation of the open-pit production system to incorporate the system as a whole. The advantage of using an RL approach lies in the fact that the agent can be trained in a stochastic environment representative of the open-pit operations, where advances in data acquisition and the rise in adoption of Internet of Things (IoT) devices in mining operations, as described by Hazrathosseini & Moradi Afrapoli (2023), allows the development of data-driven simulations of pit operations representing the stochasticity at every activity of the fleet cycle. Moreover, agents trained under an RL framework learn to take actions based on system state observations to maximize a discounted cumulative reward signal, which allows the truck dispatching agent to gauge the impacts of its decision on the future states of the mine equipment traffic, queues, and productivity.

### 1.2.3 Deep Reinforcement Learning

RL is a branch of ML that implements a computational approach to learning an optimal policy and sequence of decisions through interactions with an environment (R. Sutton & Barto, 2018). In an RL framework, an agent, an abstraction for the decision-maker, interacts with an environment at different time steps. At any time-step $t$ where the agent must act, it observes the current state of the system, $s_t$, and makes an action, $a_t$ based on it. The environment then responds to this action by transitioning into a new state in the next time step $s_{t+1}$, and providing a reward $R_{t+1}$ for the agent. This sequential decision-making behavior (Figure 1-1) repeats itself until the environment transitions into a final state and the interaction ends.

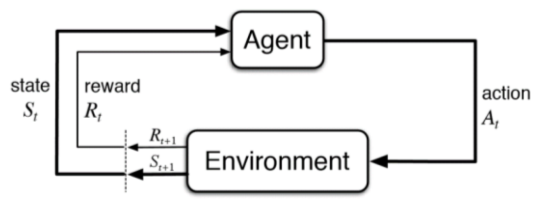


Figure 1-1. RL conceptual framework (After Sutton and Barto (R. Sutton & Barto, 2018)).

RL aims to enable the agent to learn an optimal decision-making policy that maximizes the cumulative reward received throughout its interaction. The objective function that RL algorithms optimize is the total discounted reward accumulated by the agent by interacting with the environment over the entire episode. The RL agent seeks to learn a relation from state-action pairs to value, denominated Q-value, $Q(s, a)$ which is defined as the expected total cumulative reward achieved by the agent (Eq. 1-1). The cumulative nature of the return allows the agent to understand the impact of actions over long sequences of time rather than acting greedily at every time step.

$$Q(s, a) = E\left[\sum_{n=0}^{N} \gamma^n R_n \middle| \, s_t = s, a_t = a\right] \tag{1-1}$$

During training, the agent aims to find the optimal policy $\pi^*$, which is an action selection strategy based on the system state at any time, that yields the maximum Q-value, as expressed in Eq. 1-2.

$$\pi^* = \arg\max Q^{\pi}\,(s, a) \tag{1-2}$$

The rewards returned by the environment are designed to reflect the desired goals to be achieved in each application. In the context of open-pit short-term planning, this can be profited from ore deliveries to the crusher and penalties from deviations in production targets. Estimating a Q-value function for every state and action pair would be impossible in real-world problems; therefore, the Q-value function is usually approximated as a function, which then can be trained with any supervised learning method on the collection of experiences the agent generates. The agent interacts with the environment trading off exploring it, discovering the impact of unknown actions, and exploiting well-proven high-reward decisions. In order to create the optimal policy, it is necessary to have numerous interactions with the environment. So, it is typical to simulate agents and environments on computers before testing them in real-world situations (Naeem et al., 2020).

The training process of RL agents is implemented as an iterative updating approach following dynamic programming techniques. The value of each state, or policy function, is initialized arbitrarily before any interaction, and as the agent acts and observes the environment, it updates its current value estimates from $Q(s, a)$ towards the direction of the new value observed for that same state, the target, $R_t + \gamma \max_a Q(s_{t+1}, a)$, until a convergence criterion is met, such as state value estimates not changing below a small threshold at successive updating steps.

The use of DNN as function approximators in RL frameworks, along with recent developments for the efficient training of DNN (Alzubaidi et al., 2021), provides the ability to learn complex and non-linear relations between environment states, agent actions, and future cumulative rewards. In this framework, each agent's interaction with the environment, is denominated an experience and expressed as the tuple $(s_t,\ a_t,\ R_t,\ s_{t+1})$ for state, action, reward, and next state, that serves as an observation input for the training of the DNN that approximates the policy or the Q-value function.

Mnih et al. (2015) proposed a practical framework to implement a DNN to approximate the optimal Q-value function $Q(s, a)$ within a RL framework. Their algorithm, Deep Q-Learning (DQL), was able to learn a complex policy to play 49 different Atari games, with different goals and controls, using only image inputs, highlighting the capabilities of learning to excel at a diverse array of challenging tasks. This is deemed the first successful application of NN within RL, DRL, and the authors discussed some of the challenges of using DNN as function approximators in RL. When using a non-linear function approximator for the Q-value function such as a NN, the RL algorithms are well known to be very unstable and to diverge due to the highly correlated sequence of observations from interacting with the environment at sequential time steps (Tsitsiklis & Van Roy, 1997). The authors addressed these issues by implementing an experience replay buffer, where experiences are stored and randomly sampled at every DNN updating step to break the correlation from strictly successive experiences. The experience replay buffer serves the function of the training set to train the Q-value DNN. The training of the decision-making NN relies on the agent interacting with the environment collecting experiences in the form of vectors, $e_t = (s_t, a_t, r_t, s_{t+1})$, that represent each transition observed from taking action in the environment along with the reward obtained. The experiences are continually stored in the memory replay buffer, from where batches of experience samples are randomly drawn to update the weights, $\theta$, of the NN agent. The updating process is performed via gradient-based optimization techniques to minimize a loss measure $L(\theta)$ between the action-value predictions obtained by the NN agent and the actual action-values observed from interacting with the environment. Commonly, this loss measure is chosen to be the mean square error (MSE) between the observed returns (targets) and the predicted returns form the network at training step $i$.

Moreover, the authors introduced a target-network, a copy of the DNN, that is used to estimate the value of the value targets but is updated only periodically to break the correlation between the Q-values and the target values that could harm training mechanics. Although numerous DRL algorithms have been proposed since, including modifications to the same DQL approach, these components remain a staple and a key part of any DRL framework (Obando-Ceron & Castro, 2021). Algorithm 1 describes the general DQL framework.

van Hasselt et al. (2016) showed that the original implementation of DQN tends to overestimate action values due to estimation errors of any kind that induce an upward bias as the maximization operator uses the same weights to both select and evaluate action. Double Deep Q-Learning (DDQL) was proposed to decouple the action selection from its evaluation by using the target network to estimate the target Q-value for each action. The authors reported that this simple tweak reduced the overestimation bias significantly and improved the performance of the DQN agents and is a widely used modification to the original Deep Q-Learning framework used in practice.

**Algorithm 1.** General DQL framework

Initialize replay memory $D$ to an initial capacity $N$. Initialize action-value function $Q$ with weights $\theta$ and target action-value function $Q^{tgt}$ with weights $\theta^{tgt} = \theta$.

For each episode:

For $t = 1, \dots, T$:

Observe environment state $s_t$

With probability $\varepsilon$ select a random action $a_t$ otherwise $a_t = argmax_a Q(s, a)$

Execute action $a_t$ in environment. Observe reward $r$ and next state $s_{t+1}$. Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in memory replay buffer $D$

Sample a random batch of transitions from the replay buffer $D$.

For every transition in the batch, calculate target $y = r$ if episode ended at this step or $y = r + \gamma max_{a_t} Q^{tgt}(s_{t+1}, a_{t+1})$ otherwise

For every transition calculate loss $L = (y - Q(s_t, a_t))^2$

Update $\theta$ to minimize loss with respect to model parameters

Every $C$ steps set $Q^{tgt} = Q$

#### 1.2.3.1 Applications in open-pit mine planning

RL approaches for open-pit planning were initially proposed by Askari-Nasab & Awuah-Offei, (2009) for long-term planning under the name of intelligent agent-based open-pit mine planning (IOPS) to determine the optimal combination of pushbacks that maximized the expected return over the pit life-of-mine. The authors developed a discrete simulation engine to model pit pushback expansions and how it impacted the project's economics to train the scheduling agent, as detailed by Askari-Nasab et al. (2007). Although they highlighted the potential of RL techniques to address complex decision-making in long-term open-pit planning, there were no follow-up revised methods or attention from other researchers. Paduraru & Dimitrakopoulos (2017, 2019) introduced RL for open-pit short-term planning, in which an RL agent is trained to learn optimal destination decisions for each mining block for a given production schedule. Although it does not capture the full dynamics of truck-shovel operations and focuses more on the global supply chain, a DES serves as an environment. Furthermore, Kumar et al. (2020) and Kumar and Dimitrakopoulos (2021) extend this research to account for real-time new information obtained through sensors or other monitoring technologies, focusing on a mechanism to incorporate new information on mineral grades and characteristics. They highlighted the potential of RL for adaptive and self-learning mining systems. These DRL applications so far, however, focus on decisions regarding the sequencing and destination of mining cuts and do not include equipment management and allocation which is another key aspect in open-pit short-term planning. Avalos & Ortiz (2023) proposed a DQL algorithm to optimize the long-term open-pit mine plan considering metallurgical variables and metal grade uncertainty. The DQL selects blocks to extract at daily time step intervals based on geotechnical constraints by keeping track of the available blocks when an action is required, with the goal of maximizing the NPV of the mining project. The authors highlight the potential of DRL approaches to integrate multiple types of variables as well as handling uncertainties while achieving practical results. Levinson et al. (2023) proposed an Actor-Critic reinforcement learning algorithm for the stochastic optimization of planning decisions, including extraction sequence, destination policy, stockpiling and preconcentration of a copper open-pit mining complex. The reinforcement learning agent successfully learns a strategy to optimize the short-term planning decisions of the copper open-pit under metal grade uncertainty.

Some efforts have been made to introduce data-learning approaches in the truck assignment workflow. Ristovski et al. (2017) proposed using ML models to learn probability distributions of equipment activity times based on historical data. Then they formulated an IP assignment problem that is solved every time a truck needs a decision, using the activity times predicted by the ML model. However, the algorithm behind the truck assignment is still a deterministic IP model that suffers from the same drawbacks as discussed above.

Zhang et al. (2020) proposed a DRL approach to dispatch heterogeneous truck fleets in open-pit mines to maximize fleet productivity. The agent controls each truck individually and decides where to dispatch them after a cycle is completed. Each truck is treated as an individual agent that collects experiences independently, which are then pushed to a global experience buffer to train a centralized truck dispatching policy. The model is trained based on one 12-hour production shift and is found to over perform common industry heuristics. The mine environment does not model the interaction of trucks in the shared road network where overtaking is not allowed for safety reasons, and road conditions affect the cycle of the truck, which leads to corrupted experiences that the authors flag and remove from the training stage. However, it would potentially affect the agent's performance in a real-world scenario. Moreover, the approach focuses on maximizing total material mined with no regard to differential targets. Therefore, it would not be practical in a mining setting where waste dumps and different processing streams have different production targets to achieve. de Carvalho & Dimitrakopoulos (2021) proposed a similar DRL approach but integrated with a production schedule to achieve the mine targets. The mine environment is modeled as a DES that transitions between trucks' operating cycles and breakdowns. No other interactions between the fleet are considered between transitions, and the time between transitions is modeled directly based on historical cycle times. This represents a problem as truck cycle times directly respond to the dispatching policy; therefore, the agent is getting a biased performance measure of its learning dispatching policy. The authors trained the dispatching agent for episodes consisting of 5 days of production and reported improved truck fleet performance. On the other hand, Huo et al. (2023) presented an RL framework to dispatch truck fleets with a focus on reducing greenhouse gas (GHG) emissions. The authors developed a multi-agent tabular model that doesn't use a Neural Network but instead learns a value for each state-action pair, where each truck learns its dispatching policy. The main limitations are using a value table and learning one policy for each truck, which would severely limit the model's applicability to real-world size examples, as the authors only tested their framework on small fleets with a simple open-pit simulation environment.

The methodology presented in this research attempts to address the main shortcomings of the available literature in DRL applications for truck dispatching in open-pit mines by considering the interactions of the truck fleet as it travels through the shared road network, linking the dispatching policy to a production schedule with different targets at different destinations, heterogeneous shovel and truck fleets, as well as verifying the learned dispatching policy for a month of production rather than a few shifts.

## 1.3 Objectives of the study

In relation to the problem statement, this study's main objective is to develop and verify a DRL framework for loading and hauling equipment allocation for open-pit short-term planning. The framework is required to account for uncertainties throughout the equipment operating cycles and achieve a robust performance. The main operational objectives to be considered are loaders and destinations, ore processing plants and waste dumps, production goals and ore feed quality targets. The framework is verified by using a case study with a real equipment dispatch database to model the equipment operating cycle uncertainties.

Three specific objectives are set to achieve the main objective of this research and are described below.

- Develop a DES model using equipment dispatch database records to model equipment cycle uncertainties to use as a virtual environment for an Artificial Intelligence (AI) agent to learn trucks and loaders allocation policies.
- Develop and verify a DRL framework with a case study to train an AI-based agent to learn a truck fleet dispatching policy that achieves production and feed quality targets under uncertainty of equipment cycles.
- Develop and verify with a case study a DRL framework to train an AI-based agent to learn a shovel allocation policy over a three-month production quarter, 3 three months, that achieves production goals under uncertainty of equipment cycles.

## 1.4 Scope and Assumptions of the Study

The scope and assumptions of this research stem from three major areas: the details of the DES model used to train the AI-based equipment allocation agents, the details of the truck fleet dispatching framework, and the details of the shovel allocation framework.

The scope and assumptions of the DES model are described below.

- Only shovel and truck operations are considered; no ancillary equipment or operations such as drilling and blasting are considered. This simplifies the environment but keeps the level of detail required for truck fleet dispatching and shovel allocation decisions for the short-term plan.
- The mining faces, aggregation of individual blocks from the resource model, are considered as a whole, and as shovels dig, the loaded bucket material has the same characteristics as the whole mining cut, such as density and grade. No finer divisions are considered.
- Shovel's bucket cycle time, truck spotting times, dumping time, and velocities are probability distributions.
- The mine road network is modeled as connected edges along which trucks travel, where each edge has a given gradient. The mine road network is assumed to be split into edges, or individual road segments, of approximately equal grade between intersections or destination nodes, holding either a loading or a dumping area.
- Processing plant components are not modeled, and trucks dump into a hopper that feeds the processing plant, which is considered the end point of the truck operating cycle and

used to record production statistics. Hoppers have a given capacity to hold material delivered from the mine, and a fixed output flow. If the hopper does not have enough capacity for the upcoming truck payload, this has to wait until there is enough room to dump.

The scope and assumptions of the truck fleet dispatching DRL framework are described below.

- In the DES model, truck movement is simulated across each individual road segment. Each road segment holds a queue of trucks to account for potential bunching of trucks.
- The agent is tasked with achieving loader and destinations, both ore processing streams and waste stripping, shifting production targets and keeping the ore feed quality within a given range.
- The agent can decide to which ore processing destination to dump its payload. There are no fixed ore processing streams for the material coming from each ore mining face. Material from a given ore mining face can be dumped into any ore processing stream by any truck cycle.
- The shovel movement schedule is assumed to be known beforehand. During the simulation once a shovel has depleted a mining face, it takes some time to move to its next assignment, during which it is marked as unavailable for new truck assignments until it reaches its new mining area.
- Equipment failures and breakdowns are not explicitly incorporated in the decision-making process, as rerouting or reallocations. However, the agent's state observation uses current assignments to suggest allocation decisions and a mask to avoid incompatible loaders.

Therefore, by indicating a shovel as unavailable or removing a truck from the system, the agent is able to suggest actions with the reduce fleet.

The scope and assumptions of the DRL shovel allocation framework are described below.

- The agent is tasked with achieving shift production targets over a production quarter, three months horizon. Targets as expressed as tonnes delivered by the truck fleet to each destination by the end of each shift.
- The mining faces have a fixed destination policy.
- Each shovel has a fixed truck fleet throughout the proposed time horizon.
- Truck movement is not simulated at each individual road segment, but rather by calculating the time required considering the entire path between its source and destination. The collection of road segments to travel is retrieved, and a total travel time is calculated by sampling the velocity distributions and correcting for the road segment gradients. This allows faster simulation runs to model the 3-month period at each iteration.

It is important to note that the systems proposed here for truck fleet dispatching and shovel allocation decisions are independent of each other, as described by their scope. In practice, a sequential implementation could be devised, learning the shovel allocation plan in the first stage, and then the dynamic truck fleet dispatching following the shovel schedule. Further research should address strategies to develop a nested or integrated system.

Additionally, the framework investigated in this work only encompasses loading and hauling activities, which are the key productivity drivers in open-pit mining. Ancillary operations, such as drilling and blasting, are not considered and left for future research. Moreover, this research does not extend into aspects of real-world deployment and monitoring of the DRL agent within a mine IT system. Although it is a critical aspect for adopting AI-based technologies, there is a large corpus of information about AI-applications deployment, which would be the protocols to follow for the application of the results of this research as it is an AI-based decision-support system like any other. The focus of this research lies in developing the DRL agents and procedures for training it for open-pit operational planning.

## 1.5 Research Methodology

As discussed earlier, this research aims to develop a DRL framework for a robust allocation and dispatching of loading and hauling equipment for short-term open-pit mine planning. The two specific problems that are being targeted here are the dynamic truck fleet dispatching and shovel allocation for short-term planning. The framework must account for and optimize the system, including all uncertainties along the equipment operating cycle, to find a robust strategy, and is verified using a case study.

The first part of this study involves reviewing the literature on the application of AI and data-driven techniques in open-pit mine planning, and analyze the research trends in this general area which contains the specific methods proposed here. The review was carried out by systematically searching and cataloging research works using an appropriate keyword selection to identify sub-areas and their relevance. Moreover, to analyze the prevalence of specific techniques and algorithms throughout time, to understand what has already been tried and what techniques and algorithms have been most successful in this area.

RL algorithms require the use of an environment to let an AI-based agent interact with. In this research, a DES model is proposed to serve as a virtual environment to let the AI-based agent interact with and receive feedback in the form of a reward signal for its performance in the desired tasks. Since the goal is to learn a loading and hauling equipment dispatching and allocation policy, the DES model simulates the open-pit loading and hauling production cycles, keeping track of the relevant key performance indicators (KPIs), to measure the agent's performance. The DES model considers uncertainties along the entire load and haul cycle, including truck spotting at loading and dumping points, truck dumping, loader loading cycles, and truck hauling. The road network is integrated in the DES model by simulating the movement of trucks along the road segments.

For the dynamic truck dispatching model, the DES model simulated the movement of trucks along each of the individual road segments, keeping track of the order of trucks as they enter the different road segments to maintain a queue simulating the bunching, as overtaking is not allowed. This allows to incorporate of the truck bunching mechanism that has an impact on truck productivity, as discussed by Soofastaei et al. (2016), and the lack of is a main limitation of the DRL-based truck dispatching algorithm proposed by Zhang et al. (2020) as discussed by the authors. The shovel allocation model considers truck movement over the overall path and does not model the truck bunching mechanism to accelerate training as simulation realizations, or episodes, are much longer in nature. More details on the DES model for both applications are presented in their respective chapters.

This research proposes a purely data-driven framework for dynamic truck dispatching based on a DRL approach to train an AI agent to learn an optimal truck dispatching policy. The AI agent is designed to use information typically available to mine dispatching and equipment tracking and control systems, such as location and equipment characteristics and current rate of production targets completion, to ensure the feasibility of its industrial implementation for real-time truck dispatching. The AI-based truck dispatcher uses the DES environment to learn the patterns in the equipment's uncertain operating cycles and the system's dynamics to learn a robust policy, which is verified across multiple realizations to ensure its consistency. The goal set for the truck dispatching is to achieve shift production target goals, equipment productivity targets, and material blending targets at the ore processing streams.

The shovel allocation model aims to determine a shovel allocation plan for a time period of a quarter, or three months, of production. The shovel allocation plan details the assignment of shovels to mining faces during the time horizon. A mining face is a collection of blocks from the underlying resource model aggregated for operational purposes, defining practically mineable areas. This model assumes that each mining face has a homogeneous density and metal grade calculated from the underlying resource model. Additionally, the destination for the material of each mining face is predetermined. This is commonly defined before allocating equipment resources to mining activities as part of the mine planning process based on cut-off grade policies and production targets. The problem is modelled sequentially by simulating the open-pit truck and shovel production environment, where every time step corresponds to the depletion of a mining face by a shovel and the requirement of a new assignment. The NN-based agent is called to provide the next assignment for the shovel in need by reading information about the system's state. The agent's goal is to consistently achieve tonnage production targets at each destination, including waste stripping targets, for the period considered.

A case study based on an iron ore open-pit mining operation is used to verify the models. Equipment activity records are available based on a dispatch database containing one year of equipment activities, which is used to model the uncertainty in the different equipment cycle activities with probability models. The DRL frameworks are then implemented and trained on the DES model of the case study, and their performance is verified. Both the DES model and the DRL algorithms were implemented in Python, using the Pytorch library for the AI-based agent training. The general methodology of this research is presented in Figure 1-2.

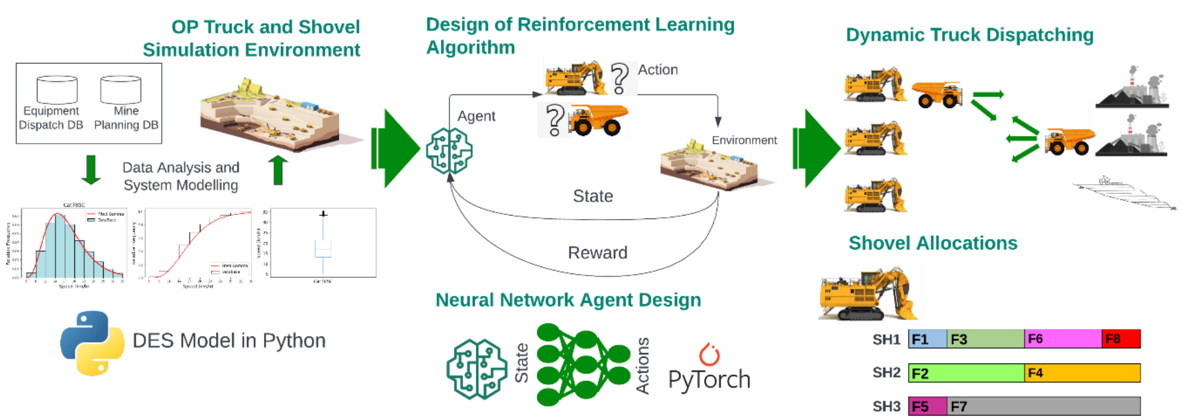


Figure 1-2. General methodology of the research.

## 1.6 Scientific Contributions and Industrial Significance of the Research

The main scientific contribution of this research to the current state-of-the-art is the development and verification of a practical framework for uncertainty-based tactical open-pit planning based on Deep Reinforcement Learning. This research contributes to creating new knowledge, understanding, and innovative technologies required to bridge the gap for adopting data-driven and AI applications in the mining industry.

The proposed formulation and methodology developed allow equipment production records to incorporate uncertainty to develop robust short-term equipment allocation policies. Specifically, the main scientific contributions of this work are described below.

- Data-driven approach to open-pit short-term planning using equipment dispatch and mine planning databases. AI agents are developed to leverage the data in an open-pit production simulator to learn and propose robust equipment allocation policies.
- AI-based truck dispatching policy learned by an agent interacting with a virtual open-pit production simulator. The AI framework uses a multi-agent centralized policy scheme, where each truck is considered as an individual cooperating agent and a single central dispatching policy is learned to handle heterogeneous fleets and different production targets. This approach allows to incorporate uncertainties across the entire operating cycle and achieves loaders, destinations production goals, and quality feed targets.
- AI-based approach to shovel allocation planning. The AI agent leverages the open-pit production simulator to find a robust shovel allocation plan to mining faces over a production quarter that achieves production target goals.
- A prototype application developed in Python using the Pytorch deep learning library was developed to allow knowledge transfer to industry practitioners.

## 1.7 Organization of Thesis

Chapter 1 of this thesis provides background and introduction to the problem considered in this research and its significance. A summary of the literature on short-term planning for open-pit mining is provided, identifying the current state of the art and its limitations. The objectives of this research are stated, as well as its scope and limitations, and the general research methodology encompassing both the AI-based approach to the truck fleet dispatching and shovel allocation problems is described.

Chapter 2 presents a detailed literature review on AI and data-driven approaches in open-pit mine planning.. The main research trends in applying different AI techniques to solve open-pit mine planning problems are identified and described, identifying potential gaps and opportunities for further development.

Chapter 3 describes the proposed DRL framework for open-pit truck fleet dispatching. The DRL framework trains a NN-based agent that learns to dispatch a heterogeneous truck fleet to achieve loaders and destination production targets and keep the quality of the ore feed within desired bounds by interacting with an open-pit production simulator. The open-pit production simulator incorporates uncertainties over the entire operating cycle, allowing the NN agent to learn a robust dispatching policy.

Chapter 4 describes the proposed DRL framework for shovel allocations in open-pit mine planning. The DRL framework allocates shovels to mining faces over a time horizon of a production quarter, to achieve shift production targets. The agent proposes a shovel allocation schedule, identifying the sequence of mining faces each shovel extracts over time.

Finally, Chapter 5 presents concluding remarks, the research contributions, and suggestions for future work.

# CHAPTER 2

# Review of Artificial Intelligence and Data-Driven Approaches in Open-Pit Mine Planning

*This chapter aims to identify the most recent research trends on utilizing AI and data-driven techniques in strategic planning for surface mining operations. The objective of this chapter is to enhance comprehension of the present utilization of these technologies, their future potential, and any potential limitations in the field.*

*The contents of this chapter were published as a peer-reviewed journal.*

*Noriega R., Pourrahimian Y. (2022). A systematic review of artificial intelligence and data-driven approaches in strategic open-pit mine planning. Resources Policy, Vol 77.*

## 2.1 Introduction

## 2.2 Research Methodology

This chapter aims to identify current research trends in applying AI and data-driven approaches for the strategic planning of surface mining operations to understand better the current state of adoption, future potential, and potential flaws of these new technologies in this field.

To fulfill the objectives of this study, a systematic literature review was carried out following the guidelines given by Tranfield et al., (2003), who transfers systematic review methods from the medical field to the management sciences, and Xiao & Watson, (2017), who propose a rigorous methodology for literature reviews in the planning sciences. The main steps for the systematic literature review presented in this study include the formulation of the problem as research questions, the development of the search protocol, including the search query and selection of databases, the definition of screening criteria for inclusion and rejection of documents and the synthesis and analysis of the information retrieved.

The review focuses on the following research questions:

1. How and within which main research areas have AI and data-driven technologies been adopted for the strategic planning of surface mining operations?
2. Which are the most common AI and data-driven approaches for strategic planning in surface mining operations?
3. How have AI and data-driven approaches been applied in the strategic planning of surface mining operations over time?

Research question 1 deals with uncovering the main application areas in which AI and data-driven methods have been applied within the strategic planning of surface mines to understand better where most of the research effort is put on. Research question 2 is more specific to the AI and data-driven approaches to understand which methodologies have been more successful when applied to this field. Finally, research question 3 is concerned with synthesizing the evolution of AI and data-driven specific techniques (e.g., neural networks, genetic algorithms) in the literature relating to strategic planning of surface mining operations to point out potentially favorable and possibly obsolete techniques.

The search included papers from the year 2000 up to June 31$^{st}$ of 2021 in the following scientific databases: Science Direct, Springer Link, Scopus, IEEE Xplore, and Taylor and Francis. These databases include most scientific peer-reviewed work in engineering applications, with some containing relevant mining engineering journals. The dates presented here are in line with the research literature database building and analysis at the start of this research.

The general structure of the search query is presented below, adapted to match the format for each scientific database.

(OR[keywords for surface mining]) AND (OR[keywords for AI and Data-driven approach]) AND (OR[keywords for strategic planning])

The **[ ]** indicates the following set of relevant keywords for the search:

- Keywords for surface mining: Surface mining, open pit mining.
- Keywords for AI and Data-driven approach: Artificial intelligence, machine learning, deep learning, reinforcement learning, data analysis, intelligent system, metaheuristic, simulation.

- Keywords for strategic planning: strategic planning, production scheduling, production monitoring, equipment management, equipment monitoring, grade control.

The **OR[ ]** notation indicates that the query targets at least one of the keywords from that particular set. Therefore, the query targets papers containing at least one keyword from each set corresponding to surface mining, AI and data-driven approaches, and strategic planning.

Afterwards, the literature records obtained were screened based on the following inclusion criteria:

- Only peer-reviewed journal papers or conference proceedings.
- Only publications from the year 2000 onwards.
- Unique studies with duplicates or similar studies by the same authors on different journals or conferences were removed.

Moreover, to stay within the scope of strategic planning and operations management, the following topics related to surface mining that partially appears as part of the search query were not considered in this review: geological exploration, mining rock mechanics, mining equipment reliability, blasting, and mineral processing. These topics can be considered a whole field on their own, and although critical for the success of mining projects, they are out of the scope of this specific research, and to be able to cover them, a different search strategy would be needed systematically. For interested readers, an overview of research trends in rock mechanics is presented in Lawal & Kwon, (2021) and mineral processing in McCoy & Auret, (2019). To the best of the authors' knowledge, there is no systematic literature review work in geological exploration, mine safety, or rock blasting; however, there is a significant body of specific applied research in those areas.

By applying the search-query and inclusion conditions, 87 papers were retrieved for a detailed analysis of the research areas and trends. Figure 2-1 illustrates the general overview of the literature search and compilation.

## 2.3 Systematic Review

### 2.3.1 Classification of Literature

The literature database obtained from the systematic search was categorized based on the main research area, application, AI, and data-driven technique used. Then the results were analyzed to answer the research questions posed. This introduced different abbreviations to deal with the variety of applications and methods found in the literature. To facilitate the reader's comprehension, a list of all the abbreviations introduced in this section is presented in Table 2-1.

All 87 selected papers were reviewed in detail. Then, to answer the research questions, they were classified based on the research area they targeted and the specific mining application and based on the AI and data-driven approach used and technique applied.

- Research area (RA): General area of interest targeted in the publication.
- AI and data-driven approach (AIA): General AI approach from which the techniques used in the publication belong.

The RA observed from the corpus acquired are the following: Production Planning and Scheduling (PPS), Grade Control (GC), and Equipment Management (EM).

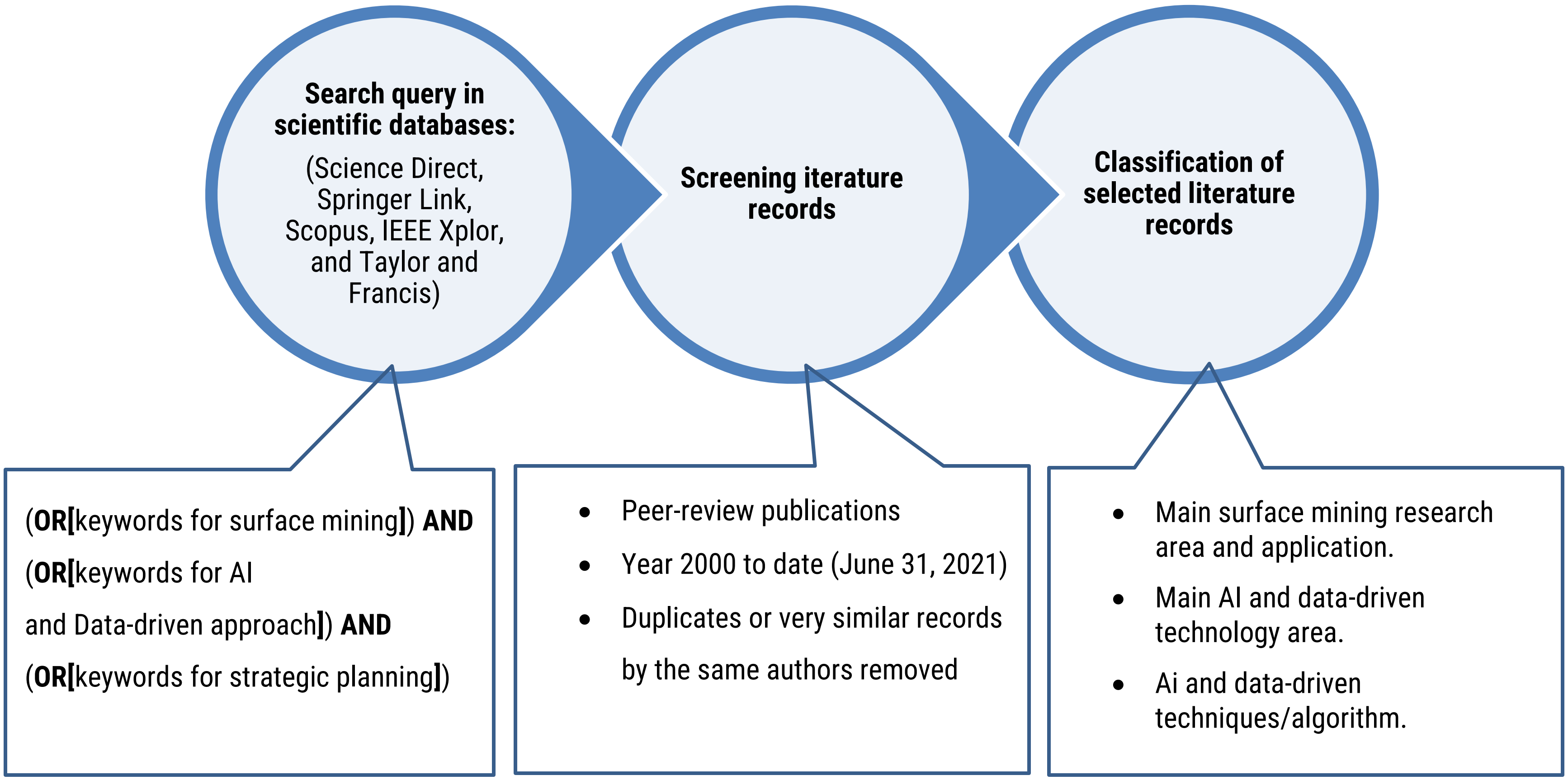


Figure 2-1. Methodology for literature database compilation.

All papers target a particular application within these broad fields of interest for the strategic planning of surface mining operations. The AIA considers SL, USL, agent-based approaches, RL, MTH, and DES.

Table 2-1. List of abbreviations for terminology used in the classification of literature.

| **Research Areas (RA)** | | **AI & Data-Driven Approach (AIA)** | |
|---|---|---|---|
| *EM* | Equipment Management | *DES* | Discrete Event Simulation |
| *GC* | Grade Control | *MTH* | Metaheuristic |
| *PPS* | Production Planning and Scheduling | *RL* | Reinforcement Learning |
| | | *SL* | Supervised Learning |
| | | *USL* | Unsupervised Learning |
| **AI & Data-Driven techniques** | | | |
| *ACO* | Ant Colony Optimization | | |
| *BA* | Bat Algorithm | | |
| *CLS* | Clustering | | |
| *CNN* | Convolutional Neural Network | | |
| *FA* | Firefly Algorithm | | |
| *GA* | Genetic Algorithm | | |
| *HOG* | Histogram of Oriented Gradients | | |
| *ICA* | Imperialist Competitive Algorithm | | |
| *KNN* | K-Nearest Neighbors | | |
| *NN* | Neural Network | | |
| *PSO* | Particle Swarm Optimization | | |
| *PH* | Progressive Hedging | | |
| *RL* | Reinforcement Learning | | |
| *RL* | Reinforcement Learning | | |
| *S-B* | Search-based Algorithms | | |
| *SA* | Simulated Annealing | | |
| *SVM* | Support Vector Machine | | |
| *T-B* | Tree-based algorithms | | |

Moreover, within each RA, the mining application the research targeted was identified and tabulated. Table 2-2 summarizes the number of research papers by RA, application, and AIA.

Table 2-2. Number of research papers by RA and application (in parenthesis) and AIA.

| **Research Areas and Applications** | **AI & Data-Driven Area** | | | | |
|---|---|---|---|---|---|
| | **SL** | **USL** | **RL** | **MTH** | **DES** |
| Production Planning and Scheduling (67) | 8 | 1 | 6 | 41 | 11 |
| *Long-term planning (41)* | 1 | 1 | 2 | 35 | 2 |
| *Short-term planning (22)* | 4 | 0 | 4 | 5 | 9 |
| *Production capacity forecasting (11)* | 1 | 0 | 1 | 0 | 9 |
| *Cost forecasting (4)* | 3 | 0 | 0 | 1 | 0 |
| Grade Control (14) | 3 | 3 | 1 | 7 | 0 |
| *Cut-off grade strategy (5)* | 0 | 0 | 0 | 5 | 0 |
| *Grade Control and Ore delineation (14)* | 3 | 3 | 1 | 7 | 0 |
| Equipment Management (21) | 6 | 0 | 5 | 2 | 8 |
| *Equipment tracking (10)* | 6 | 0 | 2 | 0 | 2 |
| *Equipment dispatch & sizing (13)* | 0 | 0 | 3 | 2 | 8 |

Figure 2-2 visually represents the number of papers by category. PPS is the RA that dominates research efforts, including long-term and short-term or operational production planning and scheduling, and forecasting production capacities and capital costs.

The principal AIA taken has been the development of MTH algorithms to tackle the large-scale and complex problems of real-sized mines, with SL impacting cost forecasting applications. RL approaches were tested initially in 2009 by Askari-Nasab & Awuah-Offei, (2009) for long-term planning and resurfaced again by 2017 over multiple research efforts. Discrete simulation is used extensively for planning and scheduling at an operational level where the interactions between equipment considerably impact production Key Performance Indicators (KPIs).

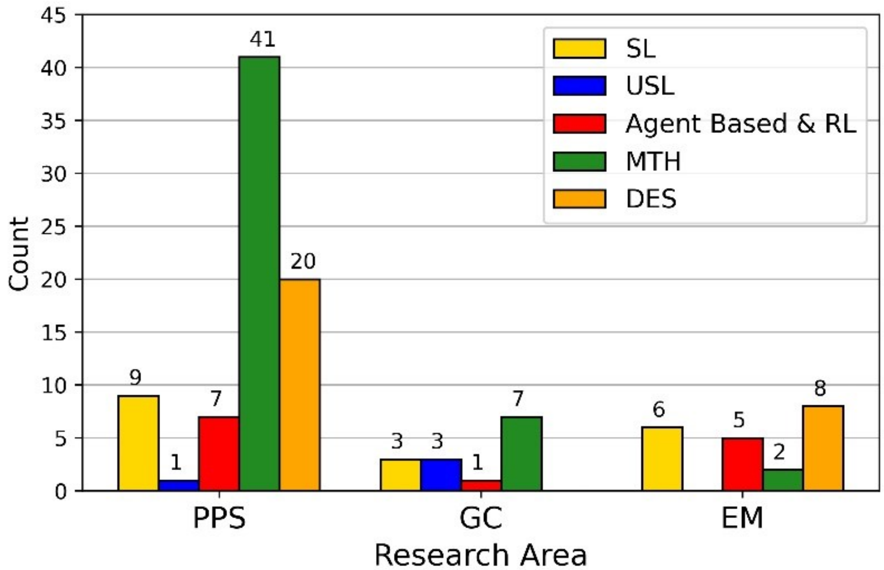


Figure 2-2. Number of papers by RA and AIA.

The EM area includes research publications that deal explicitly with mining equipment where tracking, consumption control, and equipment dispatching are the main applications. Multiple types of AIA have been tested in which DES methods are the most favored approach by researchers. It can potentially exploit large datasets that are more commonly available with the development of sensors and monitoring technologies for mining equipment. Agent-based approaches and RL appear with research focused on the dispatching and optimal routing of trucks and shovels. SL techniques also play a key role here, where large mine records can be used to predict equipment behavior and consumption.

Research on applications for grade control in open-pit mining operations appeared significantly in the database. Papers under this category cover applications that aim to find techniques to discriminate ore from waste better and delineate ore zones for improved mine planning and determining cut-off grade strategies for the operation. MTHs appear to be favored algorithms in this area to solve the complex problems of delineating ore boundaries and determining cut-off strategies.

Following the research questions, Figure 2-3 shows the number of papers by specific mining applications and AIA to get some insights into which AI and data-driven approaches have had a broader adoption for mining applications. MTHs, such as genetic algorithms (GA), significantly impact long-term planning and grade control research. These applications solve large and complex computational models for surface mines' scheduling and decision-making processes. On the other hand, DES is commonly used for more operational and short-term planning where equipment cycles are more concerned. SL approaches have seen some adoption across multiple applications, RL and agent-based approaches which have been tried for long- and short-term planning, and equipment dispatching.

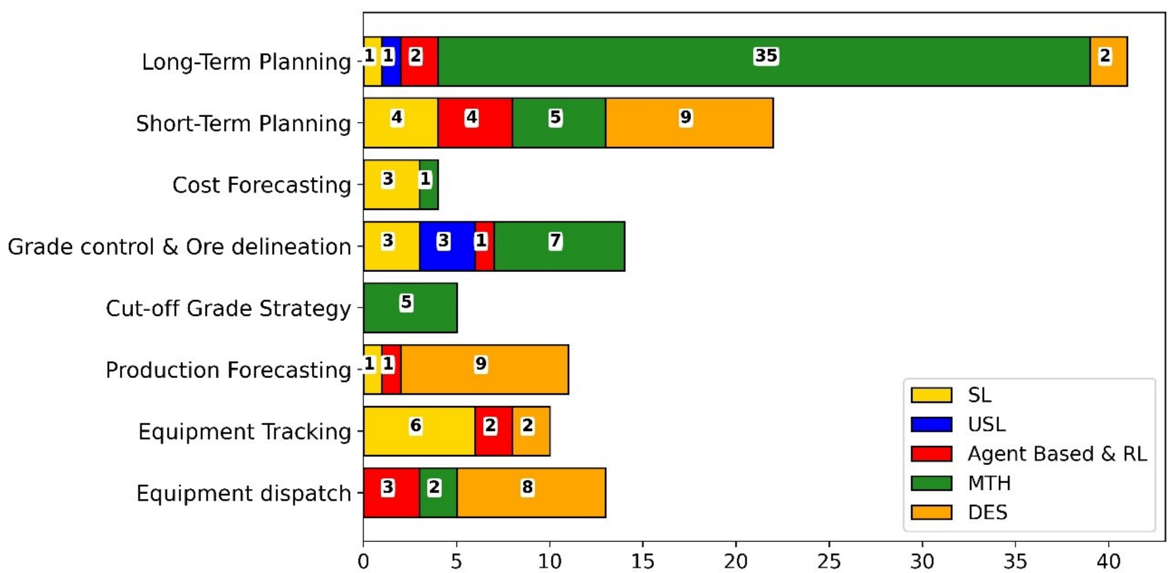


Figure 2-3. Number of papers by specific mining application and AIA.

### 2.3.2 Trend Analysis

Topic modeling and trend analysis techniques provide an important tool for researchers to navigate the large corpus of publications and studies within an area and get an overview of the evolution of topics and techniques explored by the research community (Kavvadias et al., 2020). To answer research question 3 and get an idea of the evolution of adopting different AIA within surface mining strategic planning, Figure 2-4 shows the number of publications by AIA throughout the period in question, 2000-2021. The publications were grouped in bins of 3 years to allow for better visualization, including the last year, 2021, in the last group.

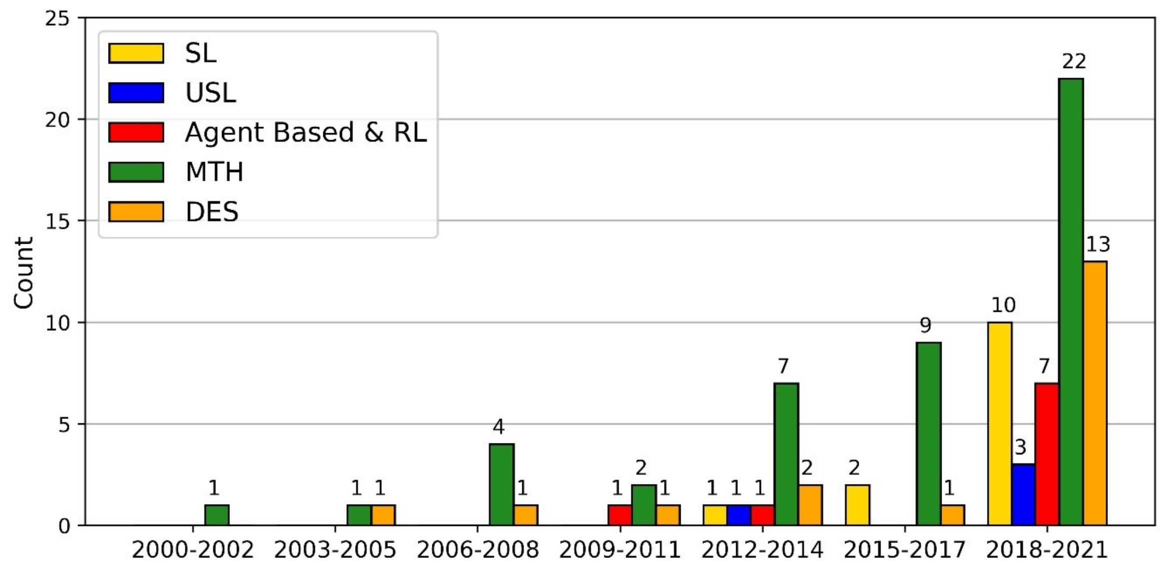


Figure 2-4. Trend of publications by AIA.

Figure 2-4 shows that the past few years have seen considerable efforts directed in applying AIA in the mining industry, along with trends in other sectors. MTH approaches have seen the largest positive trend, showing an exponential growth pattern in the number of publications. This type of intelligent computation approach has benefitted immensely from the general increase in computational power and seem to be a reliable option to solve large scale mine production planning and scheduling problems, both at a long-term and operational scale, which require the evaluation of many possible combinations of resource allocation and mining extraction patterns decisions. DES approaches have seen extensive adoption, especially in the past four years. These methods require large databases to reproduce equipment production cycles and interactions accurately and have benefited from the large-scale adoption and focus on data-driven applications within the mining industry.

SL has the largest adoption from the ML approaches, increasing within the past four years. SL requires large, labeled datasets to work efficiently, which have recently become more readily available with monitoring technology advances. On the other hand, USL seems to be the least adopted approach, appearing just after the 2012-2014 period. USL tries to discover insights from unlabeled data and is particularly challenging to bring into practical applications since the lack of a ground truth label (e.g., machine failure, ore grade) makes its interpretation challenging. This is a complication in the adoption of USL in other industries as well. Finally, RL approaches appeared as early as 2009-2011 but faded away from the literature, making a significant comeback in the past four years. RL is benefited from very recent key breakthroughs that promise to make their application feasible in real-world settings. RL high complexity remains a hurdle for industry adoption; however, it shows great potential as it explicitly combines data-driven learning capacity with decision-making processes.

To get further insight into the specific techniques tried by researchers, a trend evolution map was created for the techniques identified in each paper, shown in Figure 2-5. Analyzing the research trends of particular AI and data-driven techniques and their evolution through time can provide researchers with a better understanding of what techniques have been already tried and are starting to fade away, what techniques have seen consistent success in their applications, and what are some of the new hot topics in the literature, which greatly supports the directions for future research and work as it has been applied in other industries (Akbari et al., 2021). An example was applied by Bertolini et al., (2021) to model the topic trend evolution of ML adoption in industrial processes and understand which techniques have seen a more successful adoption in the field interest and which are becoming obsolete. Bertolini et al., (2021) identified five main clusters of techniques based on their position in the trend evolution map denominated: Question Marks, Hot Topics, Consolidated, Stars, and Obsolete. As detailed below, the AI and data-driven methods identified in the compiled literature database are classified in similar clusters based on their trend evolution throughout time. This trend score captures both number of appearances in publications and how consistent they appear throughout time, to differentiate methods that appeared in short bursts in past years but then faded away, and methods that are consistently applied by researchers throughout time to tackle a variety of challenges in the strategic planning of surface mining operations.

The SL techniques include convolutional neural network (CNN), tree-based classification and regression (T-B), support vector machine (SVM), neural networks (NN), k nearest neighbors (KNN), and histogram of oriented gradients (HOG). USL techniques include clustering (CLS), and RL techniques account for a single group of RL and agent-based algorithms. MTH techniques include ant colony optimization (ACO), particle swarm optimization (PSO), genetic algorithm (GA), simulated annealing (SA), search-based algorithms (S-B), bat algorithm (BA), imperialist competitive algorithm (ICA), firefly algorithm (FA) and progressive hedging (PH). DES techniques account for a single group.

Each topic is represented as a bubble whose size is proportional to the number of publications that use that technique. In Figure 2-5, the x-axis (Age) indicates the number of years since its first appearance in the literature. The y-axis (Trend) shows a percentage deviation from the technique publication life's 'center of gravity'. A stable topic that has appeared consistently in the literature since its first publication without a recent surge in a short amount of time would have a trend value near zero. A positive trend indicates that a topic is appearing more frequently in recent years or has had a significant comeback after initially fading away. A negative trend indicates a topic that is disappearing from the recent literature. From these definitions, six topic clusters can be identified.

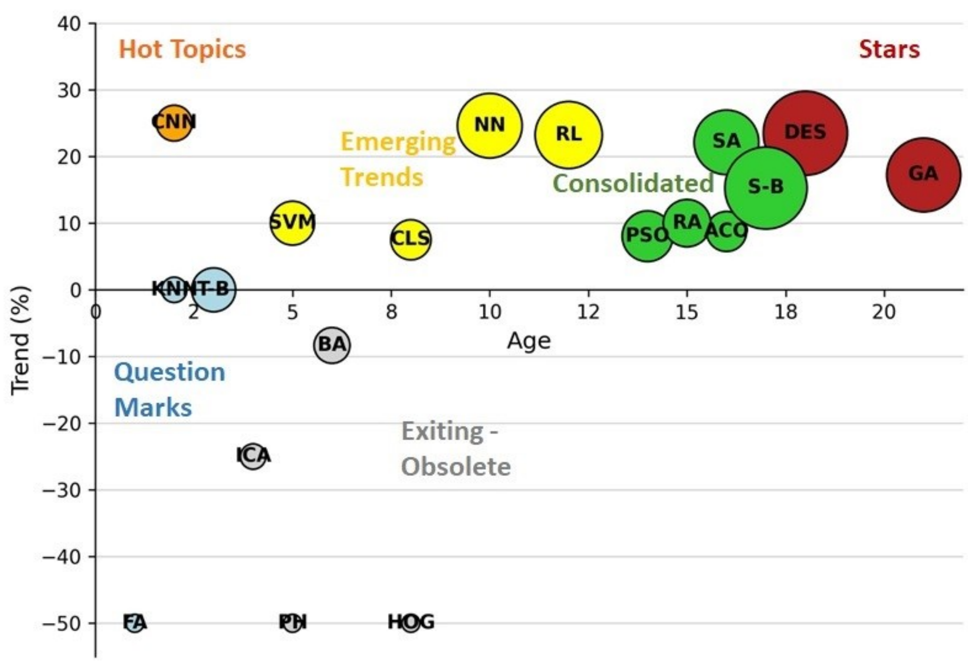


Figure 2-5. Trend evolution of specific AIA techniques during the research period 2000-2021.

- ***Stars (High age and positive trend):***
  Techniques that have appeared consistently since early on the research time period and experiencing a surge in applications include GA and DES. These techniques seem to have

the most success and are reliable in solving problems within surface mining strategic planning.

- ***Consolidated (Medium-to-high age and positive trend):***
  Techniques applied for a long time in the literature with still significant research interest include SA, S-B, ACO, and PSO. These techniques have proven to be successful in research efforts for a long time and are a solid choice to tackle complex problems within this field.

- ***Emerging trends (Low-to-medium age and positive trend):***
  Techniques that have been recently adopted and seem to have had some success with increasing research interest include RL, NN, and CLS. Given due time, these techniques could either move to become consolidated choices in the field or fade away.

- ***Hot topics (Very low age and positive trend):***
  Very recent techniques that have seen a large interest. Only includes CNN, a very recent deep learning technique that has also seen a surge in applications across multiple fields. These techniques are new promises that are yet to stand the test of time to become solid choices within this field.

- ***Question marks (Very low age and zero to negative trend):***
  Very recent techniques that have seen limited introduction and could potentially see a follow-up in the coming years include FA, KNN, SVM, and T-B.

- ***Exiting/Obsolete (Medium to high age and negative trend):***
  Techniques that have been tried in research for some time now but that have faded away include ICA, BA, PH, and HOG. These techniques do not seem to give good results within surface mining strategic planning or have been displaced by newer developments. For

example, HOG is an approach to computer vision problems that have been replaced by the appearance of CNN in general use.

## 2.4 Detailed Review by Research Area

### 2.4.1 Production Planning and Scheduling

The production planning and scheduling area concerns applications in which AI and data-driven techniques support tactical decision-making for the mining operation strategy, both long-term and short-term, including decisions on resource allocation and ore extraction to achieve economic and production targets. Research classified into this area includes specific long-term planning, short-term planning, production and cost forecasting applications.

One of the earliest efforts is presented by Pendharkar & Rodger (2000). They developed a Genetic Algorithm (GA) to determine the production, transportation, ore blending schedules, and selection of markets for multiple coal mines, highlighting the potential of GA for complex decision-making processes within the mining industry. GA has become a reliable technique for solving open-pit long-term production scheduling (OPS) problems. Moosavi et al. (2014), developed a hybrid model using a GA and augmented Lagrange multipliers to solve OPS for two pushbacks of an iron mine containing 6770 blocks. Alipour et al. (2019), compared a GA approach with the commercial software SimSched DBS for OPS, where they reported the GA achieves a 4% increase in the net present value (NPV) for the Marvin mineral resource dataset. In this research, the authors state that the commercial solver IBM CPLEX, a state-of-the-art optimization engine, could not solve the OPS after 25 days, whereas the GA reached a competitive solution within 20 to 30 minutes.

GA has also been extensively used to introduce uncertainty and extend stochastic optimization models to the OPS problem. For example, Samantha et al. (2013), formulated a multi-objective GA for OPS with mineral grade uncertainty, represented via orebody conditional simulations, for an iron deposit. The objectives of the GA were defined to obtain a schedule that minimizes deviations from targeted grades of iron, silica, and alumina elements. Moreover, Franco-Sepulveda et al. (2018), incorporated market uncertainty as well in the future prices of the minerals of interest, with a GA formulated to maximize NPV and minimize its standard deviation. Additional GA-based methods to solve the OPS problem under uncertain inputs are presented by Alipour et al. (2018), and Paithankar & Chatterjee (2019), highlighting GA's flexibility as a technique for robust decision-making.

Other successful evolutionary computing approaches for long-term planning include Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). ACO methods are based on the ability of ants to find the shortest paths to food and are efficient algorithms to search for the shortest paths over-weighted graphs. The earliest ACO application found was by Riff et al. (2005), who named their approach Miner Ants Colony. They tested their model on 50 artificial mine block models, which were similar to a confidential real mine. They reported positive results in obtaining close to optimal solutions for some of the largest and more complex datasets in about one hour. Shishvan & Sattarvand (2015) presented a similar but more detailed presentation of an ACO algorithm for OPS. Their model provides some insights into the calibration of the different parameters of the algorithm and obtains good results within reasonable computing times for a large-scale problem. Gilani & Sattarvand (2016) developed an ACO framework to integrate geological uncertainty via multiple conditional simulations of the ore deposit. The framework was tested on a large-scale dataset (about 2.5 million blocks), obtaining an NPV improvement of about 8% from a commercial software solution.

PSO algorithms follow a similar approach where solutions to a problem dubbed ‘particles’ are moved around searching for an optimal solution; it was inspired by the movement of collective organisms in nature. Ferland et al. (2007) presented an early attempt to adapt a PSO solution for OPS, where the only constraints considered were slope and mining capacity. Furthermore, Khan & Niemann-Delius (2015) designed a PSO that coulalso handle processing capacity, testing on a 7,836 blocks orebody. Results were benchmarked against an exact solution using CPLEX, which after 22 hours, reported a solution with a 4.5% optimality gap. In contrast, the PSO achieved a better optimality gap in under 12 minutes for different parameter settings. A stochastic approach was developed by Gilani et al. (2020) for mining sequence decisions under mineral grade uncertainty. Under different PSO strategies, improvements around 9% to 12% in NPV were achieved with a required time of about 15 hours. An application by Gu et al. (2020) described a PSO method for an in-pit crushing and conveying system to determine the optimal crusher location that minimizes transportation costs.

Simulated Annealing (SA) is another successful metaheuristic applied in the long-term planning and production scheduling of surface mines, which is a method inspired by the annealing technique in metallurgy that deals with the heating and controlled cooling of materials. The earliest application found by the search query was done by Kumral & Dowd (2005). They detailed a SA algorithm for OPS considering three objectives: minimizing deviation from required tonnage, penalty, and opportunity cost, and mineral content variability. The authors report a case study on a Western Australia iron ore body containing 2,773 and considering iron, silica, and alumina variables, obtaining a result in approximately 25-30 minutes, although no benchmark was presented. Danish et al. (2021) considered the single OPS to integrate stockpiling management with material mixing. They presented three test cases with the largest comprising 12,822 blocks, where the CPLEX was unable to generate a solution, whereas the SA framework proposed achieved a solution with a 7.78% optimality gap within 2 hours.

SA techniques have been especially successful for the OPS problem under uncertainty. Leite & Dimitrakopoulos (2007), integrated geological uncertainty by multiple orebody simulations and a SA detailed that seeks to find a production schedule that minimizes deviation from production targets, reporting a 20% increased NPV and better risk management than deterministic counterparts on a copper deposit test case. Albor Consuegra & Dimitrakopoulos (2009) analyzed the same stochastic SA algorithm's sensitivity, reporting no significant improvement after 10 orebody simulations and an increase of 17% of mineral reserves due to an increased final pit limit. Montiel & Dimitrakopoulos (2013), presented a similar SA to handle multiple process destinations depending on material types (e.g., acid leaching, bio-leaching) and tested on the Escondida Norte mine in Chile, a massive copper deposit. Thy benchmarked against a schedule generated by commercial software and reported a 4% increase in NPV and average deviations in the mill and waste production smaller than 5%, whereas the commercial software schedule yielded average mill production deviations of 20% and 12% for waste. On the same problem, Montiel & Dimitrakopoulos (2015) integrated multiple material transportation options. Considering metal uncertainty, Kumral (2013) used a SA to jointly solve the block sequencing problem with the ore-waste classification problem. The SA approach uses multiple orebody simulations to determine whether a block should be considered ore or waste rather than relying on a previous cut-off.

Goodfellow & Dimitrakopoulos (2017) proposed a stochastic SA to optimize the whole mineral value chain, including multiple pits, processing streams, transportation options, and markets under geological uncertainty. Two test cases were reported for nickel laterite and copper-gold mineral value chains obtaining an increased NPV and a better production risk profile in both cases. Multiple extensions to this algorithmic framework appear in the literature. Saliba & Dimitrakopoulos (2019) incorporated market uncertainty, Kumar & Dimitrakopoulos (2019) integrated geo-metallurgical variables, Levinson & Dimitrakopoulos (2020) added waste management decisions and Saliba & Dimitrakopoulos (2020) tailings management of acid-generating material, including the capital and operating costs involved.

Local search-based MTHs have seen some success in the literature as an alternative to solve the OPS problem, particularly tabu search and variable neighborhood search algorithms. Both are local search methods that explore immediate neighbors of a potential solution to discover an improved one. These methods have been particularly appealing to the stochastic version of OPS to obtain a near-optimal mining schedule robust to mineral grade uncertainty. The particular implementation of these search-based strategies is discussed by Lamghari & Dimitrakopoulos (2012), Senécal & Dimitrakopoulos (2019), Lamghari et al. (2014), and Lamghari et al. (2015).

Although other MTHs have been tested to solve the OPS problem for strategic planning, researchers have not seen similar levels of attention, suggesting that they may not be efficient in tackling the structure of OPS. These are bat algorithm (BA) by Moosavi (2020), the imperialist competitive algorithm (ICA) by Mohammadi et al. (2017), and progressive hedging (PH) by Lamghari & Dimitrakopoulos (2016). Tolouei et al. (2020) compared BA and firefly metaheuristic algorithm (FA) to solve OPS under metal uncertainty, reporting that the FA achieved better results.

RL approaches to solve the OPS problem were initially proposed Askari-Nasab & Awuah-Offei (2009) under the name of intelligent agent-based open pit mine planning (IOPS), to determine the optimal combination of pushbacks that maximized the expected return over the pit life-of-mine. The authors developed a discrete simulation engine to model pit pushback expansions and how it impacted the project's economics to train the scheduling agent, as detailed in Askari-Nasab et al. (2007). Although they highlighted the potential of RL techniques to address complex decision-making in long-term OPS, there were no follow-up revised methods or attention from other researchers. Lamghari & Dimitrakopoulos (2020) reintroduced some RL concepts for long-term OPS within a hyper-heuristic framework. The hyper-heuristic approach is described as a heuristic selection framework, in which given multiple heuristic choices for solving the OPS problem, the framework learns which is better at each iteration to produce an optimal solution.

Souza et al. (2010) presented different search-based MTHs for short-term mine scheduling and truck and shovel allocation plans to minimize deviations from production goals and number of trucks used, which were benchmarked against an exact solution obtained using the CPLEX solver and found to be competitive but requiring significantly lower time. On the other hand, Alexandre et al. (2015) reported a GA that obtained better short-term schedules than the search-based MTH for the same problem. Mousavi et al. (2016) introduced shovel allocation decisio, proposed a tabu search, and simulated annealing hybrid metaheuristic to solve the problem. Both & Dimitrakopoulos (2020) integrated uncertainty in fleet production capacity by simulating production capacity scenarios based on the mining block location and truck cycle uncertainty, along with metal uncertainty, by orebody simulations, in the OPMOP. The authors develop a SA approach to solve the problem, remarking it is impractical to solve via an exact solver like CPLEX.

More recent research efforts aim to combine discrete simulation with optimization engines to obtain operational schedules that explicitly account for equipment interaction within mine layout.

Integration of DES could potentially allow more robust and data-driven based schedules. Upadhyay & Askari-Nasab (2017) presented a detailed discrete simulation of mining operations that uses CPLEX engine to obtain optimal shovel allocations to mining faces. They extend their approach in Upadhyay & Askari-Nasab (2018b) to optimize mining faces extraction sequences, truck and shovel allocations using a multi-objective optimization approach within the simulation engine. Shishvan & Benndorf (2018) proposed a similar framework for the simulation-optimization of operational decisions for a coal continuous mining system in Germany. The simulation captures the details of the excavation and dumping practices of the mining site. The optimization model seeks to minimize excavators' and spreader downtimes to minimize cost and maximize production.

RL for mining operational decision-making was introduced by Paduraru & Dimitrakopoulos (2017, 2019), in which an RL agent is trained to learn optimal destination decisions for each mining block for a given production schedule. Although it does not capture the full dynamics of truck-shovel operations and focuses more on the global supply chain, a DES serves as an environment. Furthermore, Kumar et al. (2020) and Kumar & Dimitrakopoulos (2021) extend this same research to account for real-time new information obtained through sensors or other monitoring technologies, focusing on a mechanism to incorporate new information on mineral grades and characteristics. They highlighted the potential of RL for adaptive and self-learning mining systems.

Another major application of AI approaches in the literature for PPS is for production forecasting. This includes research directed towards predicting the productivity of a mine given its layout and equipment. It is a problem where uncertainties due to the movement of trucks and the operation of shovels within a shared mine layout (roads, mining faces, crushers) can have a significant impact and lead to overestimated production capacities or unfeasible production schedules. The favored approach to tackle this problem seen in the literature is using data-driven DES to reproduce the equipment interaction within the mine layout and evaluate multiple scenarios for strategic decision-making.

Awuah-Offei et al., (2003) present an early practical application to estimate truck and shovel requirements for a production period of 4 years in an African mine. A DES model was built using historical operation records in the SIMAN programming language. More recent applications have transitioned to using the Rockwell ARENA software to build discrete simulation models. Multiple variations in data sourcing and KPI targets have been proposed to build and use the DES of truck-shovel production cycles for operational decision-making. For instance, Tan et al., (2012) proposed using GPS data from mining truck control systems along with a DES to evaluate dispatching strategies. Soofastaei et al. (2016) proposed a DES of truck-shovel cycles to evaluate the effect of truck payload variance on cycle times and productivity for a mine in Arizona, USA. Other similar DES research applications are presented in Upadhyay et al. (2019) for an accurate estimation of Tonne per Gross Operating Hour (TPGOH), a critical productivity KPI in open-pit mines, and by Ozdemir & Kumral (2019) to evaluate the productivity improvement of a proposed dispatching model benchmarking against historical mine records. Ozdemir & Kumral (2018) proposed integrating the capability of evaluating dynamic variablesnd discrete events under an agent-based Petri net simulations framework. They highlighted the possibility of tracking dynamic variables such as equipment fuel consumption more accurately under this approach. Different applications for data-based DES models are presented by Ugurlu et al. (2020) for surface drilling operations productivity and Yaghini et al. (2020) to evaluate the impact of shovel operator performance on different mine productivity key performance indicators.

A different approach to production forecasting was proposed by Choi et al. (2020). A large dataset collected by Internet-of-Things (IoT) devices installed in an open-pit was analyzed using supervised learning techniques to predict ore production. The authors reported that SVM achieved the best performance amongst the techniques tested. This recent effort highlights the possibilities of fully utilizing data generated by mine monitoring systems.

Estimating capital costs for mining projects is another recurring application where AIA appears as a promising method in the literature. Nourali & Osanloo (2018) tested tree-based regression methods on a dataset comprising 28 copper porphyry mines reporting annual waste and ore production and capital cost. They reported encouraging results in predicting capital costs based on rock production; however, the dataset used was of small size, and conclusions from it may not be entirely accurate for new mining projects. The authors extended their research in Nourali & Osanloo (2019) to a dataset comprising 52 copper porphyry mines, recording annual mine and mill production, reserve tonnages, and stripping ratio. A support vector regression (SVR) algorithm was tested to predict mining capital cost from these parameters. Guo et al. (2019) tested multiple techniques to predict mining capital costs based on annual mine and mill production, reserves average grade and mine life for a dataset of 74 open-pit copper projects, and found a NN predictor to yield the best results with an average error of 7.77%. Zhang et al. (2020) explored the NN method in more detail, combining it with an ACO MTH for the NN training and reported improved results on the same dataset. The main drawback of AI approaches for cost estimation is the availability of datasets, which hinders the performance of more complex AI methods like NN that require tuning a large number of hyperparameters.

### 2.4.2 Grade Control

Grade control and ore delineation is another major area of research interest. Under this category, we found applications dealing with finding the optimal cut-off grade strategy, ore classification, and dig limits delineation. The cut-off strategy for an open-pit mine refers to determining values over which mineral resource units are considered ore throughout the lifespan of the mine. An early application by Ataei & Osanloo (2004) formulated the cut-off strategy problem as a nonlinear optimization problem. They proposed the use of GA to obtain the cut-off strategy for multi-metal mines. GA based approaches are further proposed by Azimi et al. (2011), to incorporate variable commodity prices and in Ahmadi & Shahabi (2018). Other MTH to solve the cut-off grade strategy problem have appeared in recent literature, such as the Imperialist Competitive Algorithms (ICA) and Particle Swarm Optimization (PSO) algorithms described by Ahmadi & Bazzazi (2019).

Beretta et al. (2019) proposed a framework for automatic lithology classification of a mining face. They used unmanned aerial vehicles to obtain imagery of a mining bench and then compared k-nearest neighbors (KNN), SVM, and tree-based methods (T-B) to classify the bench imagery into waste, ore, vegetation, and soil areas. Although they reported promising results, they recommend further investigation of more complex image classifiers like CNN. CNN were studied by Pu et al. (2019) to classify coal images as ore or gangue, reporting accuracy of 82.5% and remarking the potential of CNN methods for ore/waste image-based discrimination.

Aggregation of mineral resource blocks into selective mining units groups blocks of adequate size for the mining method and equipment to be employed. Tabesh & Askari-Nasab (2013) presented a hierarchical clustering algorithm to group mineral blocks into larger units based on grade and rock type similarities, applying a shape control method afterward to adjust for feasible mineable shapes. The approach is extended in Tabesh & Askari-Nasab (2019) to account for geological uncertainty and create mineable units that are less sensitive to metal variability. In Li et al. (2020) the impact of block aggregation in the downstream mineral processing process was considered, testing different clustering techniques. The authors reported a k-means-based clustering algorithm as the top performer that maximized the profits from the mining-mineral processing integrated system. Another application by Williams et al. (2021) focused on developing a CNN to evaluate the quality of mining dig limit clusters generated by a GA. Although they reported multiple hurdles to overcome before a real-world deployment, initial results were encouraging for short-term planning where fast computations are required. One of the main drawbacks of block clustering is the loss in ore-waste discrimination and potential dilution. Lotfian et al. (2020) proposed a GA for the clustering process. They reported long-term planning using their clustering framework achieved at least an 82% of the NPV obtained from scheduling original blocks in some test cases.

RL approaches also see an application in Dirkx & Dimitrakopoulos (2018). A multi-armed bandit framework was applied to select the best infill drilling pattern amongst a set of patterns within a budget, accounting for multiple geological elements' uncertainty. They remarked on the applicability of the method for general infill drilling campaigns.

### 2.4.3 Equipment Management

Mining operations depend on efficient control and equipment allocation to meet production and financial targets. In the equipment management research area (RA) we detail research found in the application of AI and data-driven approaches directed towards mining equipment consumption control and equipment allocation and dispatching.

The allocation and sizing of truck fleets to shovels, and shovels to available mining faces are key decisions in the operational planning of mining activities, where data-driven approaches such as Discrete Event Simulations (DES) have been widely used to evaluate different strategies, and metaheuristics like Genetic Algorithms (GA) have been popular to generate equipment allocation and routing plans. In the strategic planning section, we detailed some applications that overlap with this category but that emphasize short-term production planning; here, the remaining research is described.

Mena et al. (2013) described a simulation-optimization approach for allocating trucks' mine routes, to maximize the expected productivity of each truck on each route. They proposed a detailed DES simulation based on historic mine data to interact with the optimization engine and remark the need for accounting of equipment productivity and reliability in operational planning. Moradi Afrapoli et al. (2018) combined an optimization model for truck dispatching with a rich data-driven DES of an operating mine and processing plant. They applied the framework in a test case to determine an optimal truck fleet configuration, reporting meeting production targets with 13% fewer trucks than the configuration estimated without using a DES to account for uncertainties. Moradi Afrapoli et al. (2019) detailed a DES built to benchmark a proposed dispatch optimization model against commercial alternatives applied to a mine test case, which remarks the potential use of DES as a powerful tool for accurate data-driven scenario and mine strategy evaluation.

Agent-based approaches have also been explored for the truck-dispatching problem in which, rather than posing a global optimization problem, trucks are considered individual agents that receive information from the mining system and seek to optimize a goal. The first record of this application retrieved by the query is by Bastos et al. (2011), in which an agent-based optimization algorithm is proposed to find the optimal routing of loaded trucks between shovels and dumping stations, using a DES of the upcoming mining shift as the training environment. On the other hand, Icarte et al. (2020) proposed a novel approach in which truck dispatching problem as a multi-agent system in which trucks, shovels, and unloading points (e.g., crushers, dumps) are represented by independent intelligent agents, and these collections of agents interact with each other in the shared mine environment. The truck-shovel interaction was modeled using a Contract Net Protocol (CNP). In CNP a shovel sends a call for proposals to the truck agents, which check their current state and the condition of the unloading agents and send a proposal to the shovel. The shovel then selects the best proposal amongst trucks for the assignment. They benchmarked their approach against a heuristic rule and mathematical optimization model using a DES of a real copper mine in Chile and reported achieving production targets with an 18% decrease in operating costs. Furthermore, the researchers extended their work in Icarte et al. (2021), to add a mechanism to handle machine failures by rescheduling trucks optimally.

Researchers have also used AI and data-driven approaches to accurately predict mining truck fuel and energy consumption. Siami-Irdemoosa & Dindarloo (2015) reported good results when testing a NN to predict fuel consumption per operating cycle of mining trucks based on truck payload, loading times, idled while loaded, and idle while empty times. Soofastaei et al. (2016) developed a NN to predict truck fuel consumption (liters/h) based on gross vehicle weight, truck velocity, and total road resistance using data from a coal mine in Australia.

Some applications were found that proposed a practical implementation of AI systems for equipment tracking and visual sensing. Rezazadeh Azar & McCabe (2013) described a framework for identifying and tracking mining trucks throughout the production cycle in real-time video recordings. The authors proposed a Histogram of Oriented Gradients (HoG) computer vision technique and presented an application to recognize and count hauling trips. Yao et al. (2021) proposed a CNN – NN framework to estimate the piled-up status and payload distribution (PSPD) of bulk materials in a dump truck from camera images. The PSPD describes the alignment and amount of bulk material in a dump truck's body and helps determine dumping positions to improve stress state and equipment service life. The authors presented some successful pilot tests.

Ali & Frimpong (2021) proposed a framework to improve autonomous truck steering capabilities named DeepHaul. An object recognition module was proposed to detect mining equipment, humans, and animals using a CNN from images and video recordings in the haul truck’s path. Afterwards, a RL framework was used to optimize the truck steering decision capabilities based on the visual sensing detection by putting the truck in multiple scenarios involving different objects in its path throughout a haul road.

## 2.5 Discussion

The vast majority of research is directed into the open-pit production planning and scheduling problem, with a big focus on developing metaheuristics and intelligent computation techniques to solve complex large-scale production scheduling for the life-of-mine strategic plan. The specific problem of long-term and short-term planning has received the most attention with a large variety of solution methods, mostly metaheuristics. The challenge with metaheuristic methods is that their implementation tends to be very problem-specific, and their performance could vary wildly between problem instances. However, Genetic Algorithms (GA) and Simulated Annealing (SA) have proven to be the most consistent techniques used throughout the period analyzed. Although metaheuristics are a promising approach to tackling these complex problems, the presentation of new metaheuristic techniques should follow some good practices such as those proposed by Osaba et. al. (Osaba et al., 2018) for a clear statement of assumptions, implementation details, and results reporting to encourage transparency and reproducibility of methods.

Discrete Event Simulation (DES) has also been widely adapted as an approach to support data-driven decision-making for mine planning and operation. Researchers have used DES to improve mine plans by providing an environment that simulates the interaction between the different processes and equipment during the mine operation and build algorithms that incorporate this dynamic to improve on decision-making to achieve the desired goals. DES also plays a key role in truck fleet management, especially for research applied for the truck dispatching problem, which requires near real-time decisions that significantly impact mining production.

A large amount of historical data on equipment behavior is required to successfully implement a DES model. Data compilation and cleaning from raw databases is one of the main hurdles for adopting AI and machine learning techniques for any industrial case (Bertolini et al., 2021). So it also represents an important challenge here. Future work should also focus on guidelines and good practices for how to best handle mining operation databases to build a DES or digital twin model to support decision-making.

From the more traditional Machine Learning (ML) domain, Supervised Learning (SL) techniques are the most widely used across applications such as short-term planning, cost forecasting, grade control, and equipment tracking. SL techniques rely on the availability of large amounts of labeled data to implement algorithms that learn patterns on it to make accurate predictions. For this purpose, equipment tracking and control applications seem particularly fitting for SL techniques, where problems such as forecasting truck fuel consumption and payload, and estimating hauling cycles appear in the literature. These problems use large equipment databases available for a long time and improve internet network connectivity in surface mining operations.

Grade control applications also reap the rewards of recent advances in SL techniques for image processing, which have enabled researchers to present automated rock type and ore classification algorithms using drones and digital camera images and determining optimal ore dig limits. Future work towards a real-time ore and waste discrimination system based on digital images could positively impact the mining production environment to tackle issues such as unplanned dilution.

Cost forecasting applications also appear in the literature; however, in all cases, the authors report use cases with very few data points, usually less than 100, which presents a major hurdle for its potential application. This specific use case reflects one of the major challenges in developing AI and data-driven approaches: the availability of data.

More recent applications involve Reinforcement Learning and agent-based (RL) techniques used for production planning, scheduling, and equipment management. Although the idea of RL has been around for a long time, it has not seen much real-world application and is just starting to show some successful use cases (Dulac-Arnold et al., 2019). More work in this area is needed to showcase its potential on different applications within surface mining systems, as it has seen large volumes of research for production planning and control in dynamic systems (Panzer et al., 2021), vehicle routing (Nazari et al., 2018), problems very similar in structure short-term production planning, and truck dispatching.

## 2.6 Literature Review Update

Since the publication of the results from the literature review analysis presented in this chapter, with a document compilation cutoff date of June 31st 2021, more applications of artificial intelligence and machine learning techniques for strategic mine planning have been published. Deep Learning techniques, in particular, have become an active hot research topic in multiple mining industry fields. Azhari et al. (2023) present a comprehensive review of the applications of Deep Learning (DL) techniques for the different broad segments within the mining value chain, considering exploration, extraction, and reclamation activities. The exponential growth in the research of DL implementations in the mining industry is highlighted, particularly the adoption of CNNs, which were also identified as hot topics here, and discussing some of the major challenges for the further adoption like the availability of open access datasets. To provide an overview of newer developments concerning the two specific problems addressed in this thesis, a summary of recent advances in DRL techniques concerning truck fleet dispatching and open-pit mine planning, beyond the cutoff date of chapter 2, is discussed.

de Carvalho & Dimitrakopoulos, (2021) proposed a DRL approach to dynamic open-pit truck dispatching integrated with a production schedule to achieve the mine production targets under uncertainty. The mine environment is modeled as a discrete event simulation (DES) that transitions between trucks' operating cycles and breakdowns. No other interactions between the fleet are considered between transitions, and the time between transitions is modeled directly based on historical cycle times. This represents a problem as truck cycle times are a direct response to the dispatching policy; therefore, the agent is getting a biased performance measure of its learning dispatching policy. The authors trained the dispatching agent for episodes consisting of 5 days of production and reported improved truck fleet performance. A major challenge for further work would be to extend the application timeframe for a trained model, as the dispatching model was verified to work properly only for a period of 5 days, with the performance starting to fall off. This would require a constant updating or retraining of the model which would hinder its feasibility for adoption. On the other hand, Huo et al. (2023) presented an RL framework to dispatch truck fleets with a focus on reducing greenhouse gas (GHG) emissions. The authors developed a multi-agent tabular model that does not use a NN but instead learns a value for each state-action pair, where each truck learns its dispatching policy. The main limitations are using a value table and learning one policy for each truck, which would severely limit the model's applicability to real-world size examples, as the authors only tested their framework on small fleets with a simple open-pit simulation environment.

Avalos & Ortiz (2023) proposed a DQL algorithm to optimize the long-term open-pit mine plan considering metallurgical variables and metal grade uncertainty. The DQL selects blocks to extract at daily time step intervals based on geotechnical constraints by keeping track of the available blocks when an action is required, intending to maximize the NPV of the mining project. The authors highlight the potential of DRL approaches to integrate multiple types of variables and handle uncertainties while achieving practical results. Levinson et al. (2023) proposed an Actor-Critic reinforcement learning algorithm for the stochastic optimization of planning decisions, including extraction sequence, destination policy, stockpiling, and preconcentration of a copper open-pit mining complex. The reinforcement learning agent successfully learns a strategy to optimize the short-term planning decisions of the copper open-pit under metal grade uncertainty.

## 2.7 Concluding Remarks

This research systematically reviewed applications of AI and data-driven approaches for open-pit strategic planning. The research goals were to uncover trends in AIA adoption in the period 2000-2021, understand which applications in this field are being solved using these approaches, and which specific AIA techniques have been more successful as measured by the number of appearances in peer-reviewed research publications. A comprehensive search query was designed, and 86 publications were reviewed in detail.

The goal achieved by this paper was to establish the current state of use of AI and data-driven technologies for the strategic planning of surface mines, identifying the algorithms and workflows that have been implemented for specific application cases in this domain. Overall, the adoption of AIA within open-pit strategic planning has seen exponential growth within the period considered, with successful applications across different areas of interest. The large adoption of metaheuristic and intelligent algorithmic techniques indicates the attractiveness of fast and reliable computation methods for large and complex problems. There is growing interest in discrete simulation, which involves using extensive historical mining data to create digital twins of mining operations. This can provide valuable decision-making support.The surge in supervised learning and reinforcement learning techniques shows the potential of ML adoption in operational management tasks. Finally, researchers have shown a willingness to adapt state-of-the-art AI and data-driven techniques to solve open-pit strategic planning problems, showing these technologies' potential to unlock value within the mining industry.

# CHAPTER 3

# Truck Fleet Dispatching in Open-Pit Mining using Deep Reinforcement Learning

*This chapter explains the development of a Reinforcement Learning (RL) based truck fleet dispatching system for open-pit mines to achieve production targets and maximize equipment utilization and productivity. A case study is presented in an iron ore deposit where the trained agent manages to learn a robust dispatching policy to achieve the ore and waste mining targets and maintain the metal concentration of the ore feed to the processing plants within a desired range.*

*The contents of this chapter have been submitted and are under review for a peer-reviewed publication by the Computers and Operations Reseach journal.*

*Noriega R., Pourrahimian Y. and H. Askari-Nasab (2023).Deep reinforcement Learning based real-time open-pit truck dispatching system. Computers and Operations Research.*

## 3.1 Introduction

This thesis details the development of a Reinforcement Learning (RL) based truck fleet dispatching system for open-pit mines to achieve production targets and maximize equipment utilization and productivity. The open-pit trucking system is modeled as a multi-agent environment, where each truck in the fleet acts as an individual agent, and a single centralized truck dispatching policy is learned using a RL framework. When a truck dispatch decision is required, the RL-agent observes the state of the mining system, characterized by measurements available in real-time from current mining equipment management systems to generate the dispatch solution in real-time. In order to train the system, a stochastic environment is created for open-pit operations. This involves using equipment activity databases to account for uncertainties throughout the truck-shovel fleet's operating cycle and incorporating them into the dispatch policy learning process.

Truck fleet dispatching in open-pit mines has been historically solved using linear programming (LP) based and heuristic algorithms, where fast dispatching decisions are needed to keep the fleet running (Moradi Afrapoli & Askari-Nasab, 2019). However, heuristics use limited information on the mining system to produce fast solutions and do not account for the large uncertainty sources. In addition, mining operations generate large amounts of real-time data on equipment location and status, which are underutilized by current fleet dispatching solutions (Chaowasakoo et al., 2017a). Therefore, developing a methodology that can account for operational uncertainties, produce fast and reliable dispatching decisions to meet targets, and better utilize the data collected during operations is highly desirable.

Artificial Intelligence (AI) algorithms provide promising data-driven frameworks to solve optimization problems (Mazyavkina et al., 2021) and guide real-time decision-making in complex systems (Hildebrandt et al., 2023). By learning from data, a system does not need to be fully specified through a mathematical model to find an optimal decision-making strategy, which often severely impacts the capacity to model the complexities in the system, account for uncertainty, and obtain solutions in a reasonable time frame. RL as a branch of AI and ML is particularly well suited to sequential decision-making problems in dynamic environments where actions are taken based on available information observed from the system. In an RL framework, an agent interacts with an environment by observing a state description, acting, and receiving a reward as the environment transitions to a new state in response to the agent's action. The agent's goal is to maximize the total reward accumulated during its interaction, reflecting the desired goals to achieve (R. Sutton & Barto, 2018). RL-based approaches have been proposed for the management and dispatching of the truck, or vehicle, fleets in other industries (Alcaraz et al., 2022; Jiao et al., 2021), which suggests its benefits to develop an intelligent fleet management system for the mining industry.

## 3.2 Problem Description

The last step in the production chain in an open-pit mine consists of loading and hauling material between loaders and destinations to execute a short-term mining plan. The overall short-term plan establishes shift requirements on how much material from each mining face, open areas for digging in the pit must be removed to ensure proper development of the pit, and the quantity and quality of the rock that must be delivered to each destination for further processing and refining or waste disposal. The truck fleet dispatching problem deals with the frequent real-time decision, at an operational level, of assigning a free truck to a path between a loader and a destination for its next operating cycle.

Consider an open-pit mine with N mining faces where loading equipment loads ore, valuable material, or waste, and destinations corresponding to dumping sites that receive the truck payloads, which could be one of M processing plants or stockpiles or one of K waste dumps (Figure 3-1), with $O$ trucks to a haul material. Each loader is located at a mining face, $l_i \in \{l_1, l_2, \dots, l_N\}$, has a target production output which can be expressed as a tonne per hour of material loaded, $tph_{l_i} \in \{tph_{l_1}, tph_{l_2}, \dots, tph_{l_N}\}$. These rates can be different depending on the capacity of the loading equipment and the requirements of the short-term plan. Ore processing plant destinations, $d_i \in \{d_1, d_2, \dots, d_M\}$, have a target production input, expressed as a tonne per hour or material received, $tph_{d_i} \in \{tph_{d_1}, tph_{d_2}, \dots, tph_{d_M}\}$, of a certain material quality expressed as the grade of metal concentration in the feed received, $g_{d_1} \in \{g_{d_1}, g_{d_2}, \dots, g_{d_M}\}$. These quantity and quality constraints allow the processing plants to operate within the designed metallurgical parameters. Additionally, the mine also has waste stripping excavation targets to be met at each of the waste dumps $d_w \in \{d_1, d_2, \dots, d_K\}$ to ensure the long-term development of the pit and provide access to deeper ore mining face needed for future operation.

Trucks move between loaders at mining faces and destinations along a shared road network to achieve the quantity and quality targets described above. The problem is defined by deciding a path assignment for each truck as they complete each operating cycle, where a path $p \in \{1, 2, .., P\}$, correspond to a path through the road network between a source and a destination. Trucks can be of different capacities and performance, and variabilities in the trucks and loader activity cycles due to operator skill, road conditions, weather effects, and other external factors introduce uncertainties in cycle and productivity estimations that make the real-time control of the truck fleet a challenging operational problem.

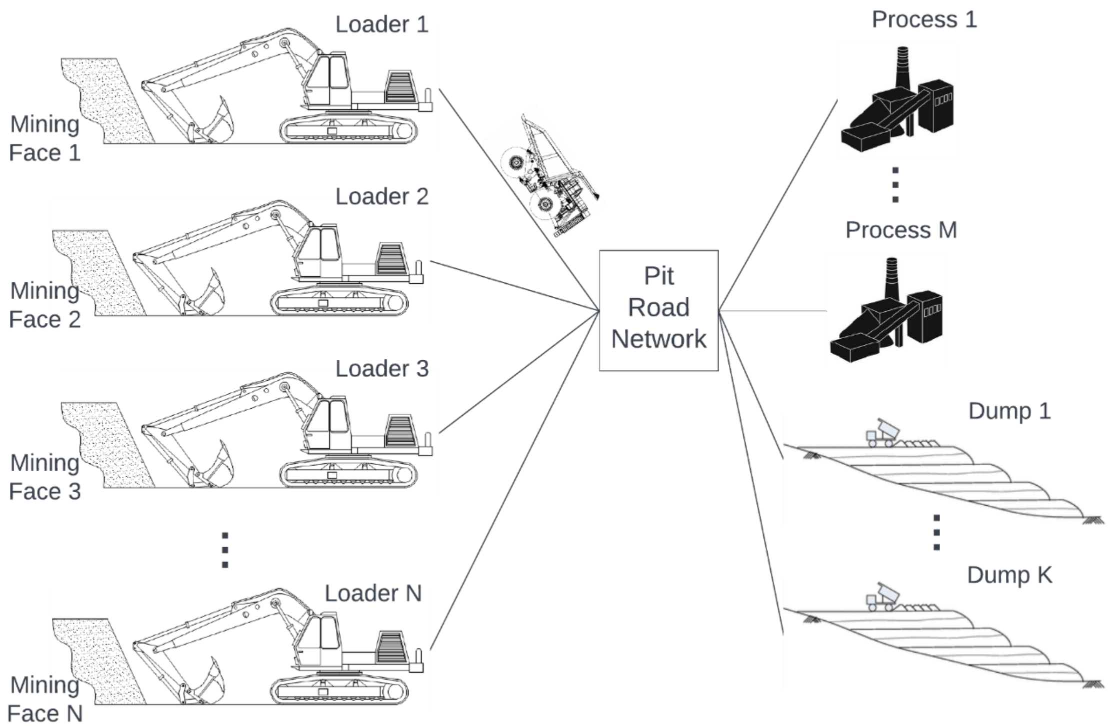


Figure 3-1. Open-pit truck dispatching problem. The truck fleet needs to be constantly routed along paths between material sources at mining faces within the pit and destinations outside the pit.

## 3.3 Methodology

### 3.3.1 RL Framework for Open-Pit Truck Dispatching

A model-free RL algorithm is proposed to solve the real-time open-pit truck dispatching problem. The task is modeled as a Markov Decision Process (MDP) where an agent interacts with an environment acting based on state observations and receiving a reward signal as the environment transitions to a new state. A MDP is formally described by the tuple $\langle S, A, P, R \rangle$ where $S \subset R^s$ is the set of all possible states in the environment with dimensions $s$, $A \subset R^a$ is the set of all possible actions with dimension $a$, $P: S \times A \rightarrow S$ is a transition function that defines the conditional probability, $p(s'|s, a)$, of the environment transitioning to a new state, $s'$, given that the agent took action $a$ from state $s$. This transition function follows the Markov property, indicating that the new state depends only on the previous last state and action taken, not past interactions. Finally, $R: S \times A \times S \rightarrow R$ is a scalar reward value that is returned for a given action taken and transition between states. The agent decision-making strategy is encoded in a policy, a mapping from states to actions, $\pi: S \rightarrow A$ .The goal of RL is to learn an optimal policy, $\pi^*(a|s)$ that maximizes the cumulative discounted return, $G = \sum_{t=1}^{T} \gamma^{t-1} r_t$, where $\gamma \in [0,1]$ is a discount rate applied to balance the present value of future rewards and $T$ is the final step of the agent interaction.

In the framework proposed, the agent represents a centralized truck dispatcher that has access to the location and state of every loader and truck in the mine and the current progress of the different production Key Performance Indicators (KPI), as most mine production and equipment monitoring systems provide this information in real-time. A DES model is built to simulate the truck and loader operations and serve as the environment to train a truck dispatcher agent. The DES is based on the simulation models proposed by Upadhyay et al. (2021), to account for the interaction between trucks moving along the shared road network and the effect of road conditions and gradient on the effect on the achievable velocity. In this environment, the agent is given control to dispatch a truck after it dumps its payload and requires a new path for its next cycle. The transition function is defined based on the operating cycle of each truck, with the initial state corresponding to the observation when the truck finishes dumping its payload and the next state corresponding to the observation after the truck finishes its assigned cycle. Each truck is considered an individual agent that collects transition experiences on its own; however, a centralized policy is learned overall to control the system.

Learning a centralized, single-agent policy is desirable as single-agent learning dynamics are much less computationally complex than multi-agent learning (Wong et al., 2022). For example, in the open-pit truck dispatching task, each truck represents the same type of agent with slight differences in performance due to size, which can be encoded in the state observations. Additionally, a mine monitoring and dispatching system has complete access to information on and control of each truck as desired. The framework proposed here is then centralized learning with decentralized execution (Bahrpeyma & Reichelt, 2022), in which multiple agents are trained with a centralized controller where each agent pushes their collected experiences to a global buffer from which the centralized policy is trained. The dynamics of the decentralized truck experience tracking and pushing towards a centralized global buffer for training are depicted in Figure 3-2.

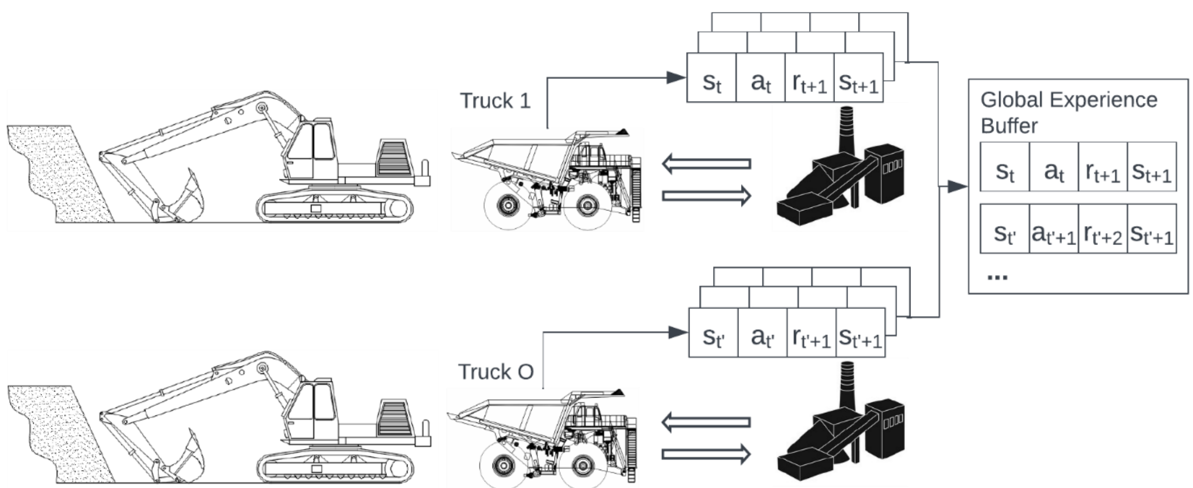


Figure 3-2. Decentralized experience tracking and pushing towards a centralized buffer for training.

### 3.3.2 Open-Pit Environment

The open pit DES environment simulates a production shift where trucks move between shovels and dumping destinations across a shared haul road network. Stochasticity is considered in each component of the truck operating cycle: hauling empty, getting loaded, hauling loaded, and dumping, with queues forming based on the limited serving capacity of both shovels and dumping destinations (Figure 3-3). Furthermore, as trucks travel along the shared road network, they are not allowed to overtake each other for safety reasons, which leads to the bunching of trucks behind each other, a phenomenon that impacts the estimation of operating cycles and productivity (Soofastaei et al., 2016).

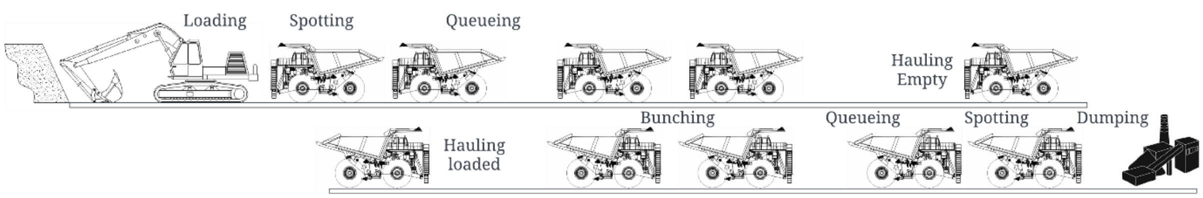


Figure 3-3. Open-pit production cycle.

The DES is based on the models proposed by Goris Cervantes et al. (2019) and Upadhyay et al. (2021). At the start of each simulated shift, shovels are assigned to a mining face based on the mine plan. Truck dispatching actions are required every time a truck dumps its payload, where the system's current state is given as input to the AI dispatched to obtain a truck-shovel assignment for its next cycle. Figure 3-4 shows the general logic of the DES environment and its interaction with the AI dispatcher agent.

When a truck starts traveling across the road network towards either a shovel or a dump destination, it is assigned a path, a collection of connected road segments from the network that is the shortest path between the current location of the truck and its next destination.

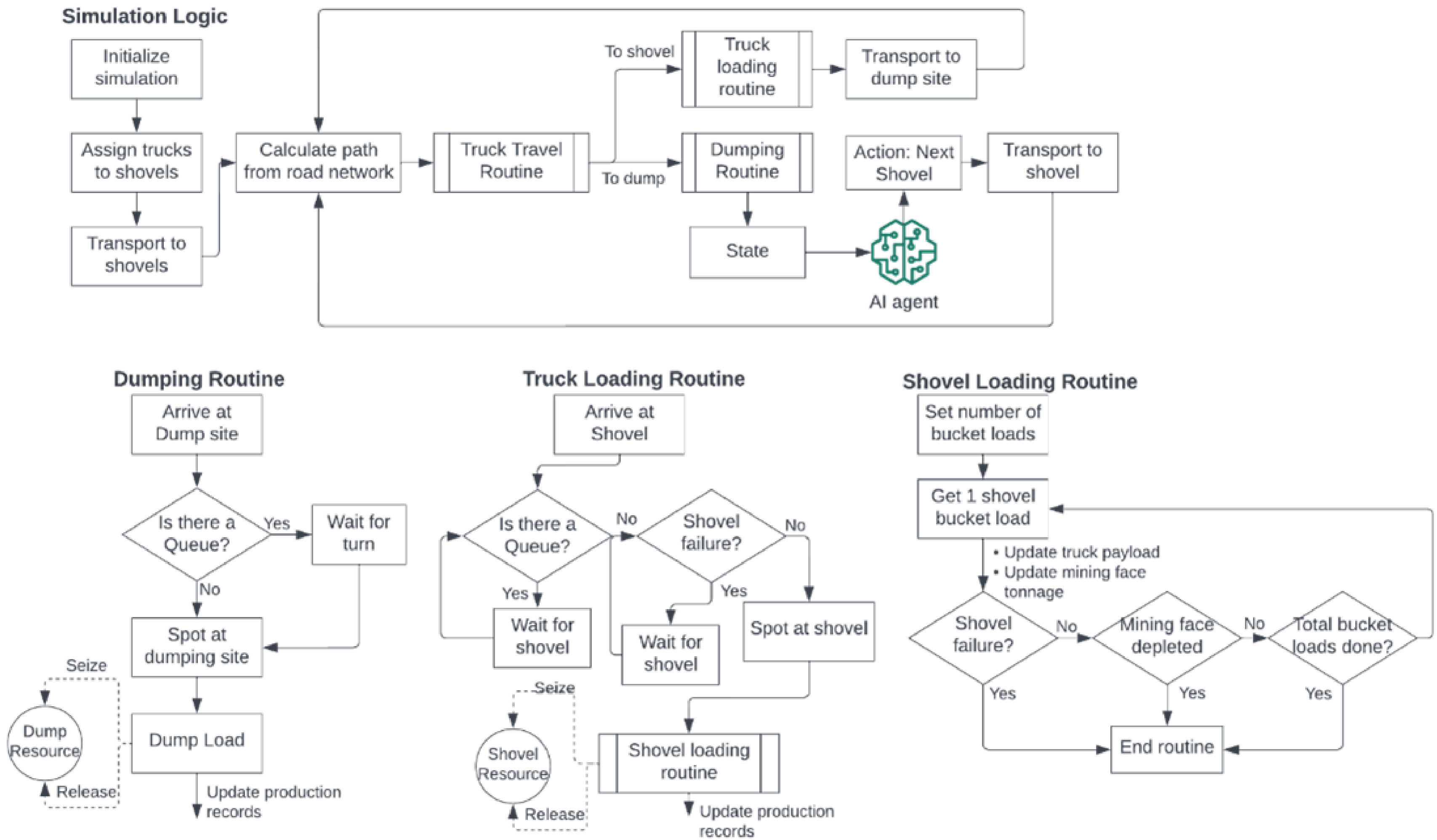


Figure 3-4. DES logic for the simulation environment to train the RL-based truck fleet dispatch agent.

Each road segment in this path contains information about the segment's rolling resistance, grade resistance, and length and maintains a queue of trucks traveling across it, which is used to model truck bunching. Moreover, each segment has a maximum speed limit $v_{road}$ that constrains the speed of any truck traveling through it.Each truck traverses its path by moving along each road segment. Truck velocities are estimated using the equipment dispatch database to model each truck type's empty flat haul movement distribution. This refers to the velocity of trucks when hauling empty over flat segments and provides a good baseline to include the effects of payload and road resistance. After entering a segment, the truck enters the road segment truck's queue and is assigned a velocity based on its truck type empty flat haul velocity distribution adjusted by the road resistance and payload based on the truck manufacturer's rimpull curves. While traveling through the different road segments, trucks are not allowed to overtake and bunch behind adopting its leading truck's velocity. This bunching mechanism is hard to predict analytically and significantly impacts productivity and its variability over the shift. Zhang et al. (2020) did not account for this mechanism in developing an RL-based truck dispatching system, reporting that it led to corrupted or wrong experiences for which a custom algorithm was developed to identify and remove them from the training process.

As described above, the open-pit production environment DES was developed in Python using the SimPy library. This efficient process-based discrete event simulation framework implements generic concepts of resources, managing its queues and waiting mechanisms and handling the scheduling of stochastic events.

### 3.3.3 State and Action Description

The action space encodes the decision to dispatch a truck to any available loader destination paths. It is expressed as a one-hot encoded vector where only one entry can take a value of 1 while all others take a value of 0. The positional index of the entry with value of 1 indicates the path to send a truck for its next cycle. In an open pit mine, not all paths between loaders and destinations are available, as some loaders work in waste mining areas for pit expansion that would not be paired to an ore processing destination. The potential available paths are commonly known before the start of the shift based on the mine plan and mineral resource quality control. The state observation is described with a long vector containing relevant information from the system's current state to make truck dispatching decisions. The information passed in the state information is depicted in Figure 3-5.

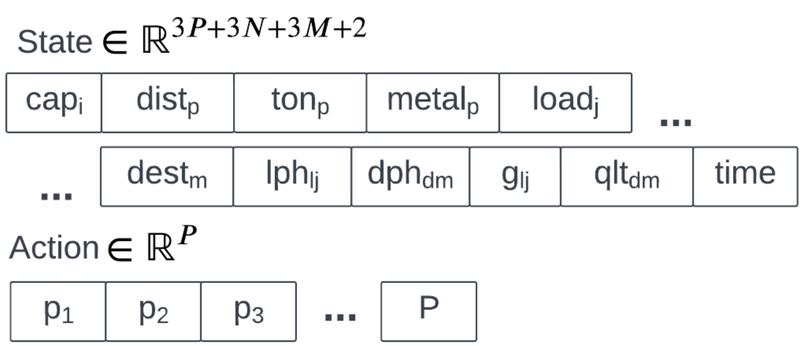


Figure 3-5. State and action representation.

Where $cap_i$ refers to the capacity of the truck dispatched and $CAP$ to the capacity of the largest truck in the system, to allow the agent to identify the increased payload provided by larger trucks. The $dist_p$ component represents the total length in meters for each loader-destination path available, which is standardized by the maximum path length. The $ton_p$ component encodes the total truck capacity currently assigned to each path, normalized by the total truck capacity in the system, and $metal_p$ encodes the total metal payload currently traversing each path based on the truck's loader assignment. The current progress of the shift targets is passed in $load_j$, for the tonnage loaded by each loader, and $dest_m$, for the tonnage delivered to each destination, expressed relative to the shift tonnage targets. The productivity, per hour production, of each loader and destination is encoded in $lph_{l_j}$ and $dph_{d_m}$respectively relative to the loader and destination productivity targets respectively. The quality, or metal grade, of the material being loaded at each mining face is passed in $g_{l_j}$, and the ratio of the current metal grade at the feed of the ore processing plants to their target metal grade, $g_{d_m}$, in $qlt_{d_m}$. Finally, $time$ encodes the current shift time, expressed as fraction.

### 3.3.4 Reward Function

The reward function at every time step, after dispatching truck $i$, is shown in Eq. (3-1).

$$r_t = \frac{cap_i}{CAP} \times (\delta_l + \delta_d + \delta_g) - \sum_{j=1}^{N} l_j^{\ -} - \sum_{k=1}^{M} d_k^{\ -} - \sum_{k=1}^{M} g_k^{\ +-} \tag{3-1}$$

The reward function was designed to guide the agent in accomplishing three main goals (i) achieving the loader production targets, (ii) achieving the production delivery targets at each ore processing destination, and (iii) the quality of the delivered material being within the desired acceptable range. To achieve this, a reward is provided to the agent as positive feedback for meeting all the targets in the episode. However, additional smaller reward signals are required to guide the agent to this success state. For this purpose, $\delta_l$ is a binary flag that indicates if the productivity targets $tph_{l_i}$ are satisfied at all loaders, $\delta_d$ is a binary flag that indicates if the productivity targets $tph_{d_i}$ are satisfied at all ore processing destinations, and $\delta_g$ is a binary flag that indicates if the quality of the material delivered at all destinations is within the acceptable range. The agent is rewarded if it accomplishes the goals after every transition. To guide the agent towards achieving its targets, as well as ensuring a continuous productivity profile, the agent is penalized for the shortfall in productivity at each loader ${l_j}^-$ relative to the targets $tph_{l_i}$, a shortfall in material delivered at each destination ${d_k}^-$ relative the targets $tph_{d_i}$, and the deviation in the quality of the material fed to the processing plants ${g_k}^{+-}$ relative to the quality target. Only shortfalls are penalized in the quantity target components to allow the agent to potentially discover strategies to increase the estimated fleet productivity.

### 3.3.5 Deep Q-Learning Implementation

The RL algorithm proposed to learn a truck dispatching policy for open-pit mines is Deep Q-Learning (DQN), one of the most popular and used RL algorithms. DQN, was developed by Mnih et al. (2015) and uses a neural network (NN) function approximator to learn a state-action value function, $Q_\pi(s_t, a_t)$, representing the expected return obtained from taking action $a$ from state $s$ at time step $t$, following policy $\pi$ after that. The Q-value function is defined by a recursive relation to future state-action pairs following the same policy (Eq. (3-2)), which embeds a look-ahead estimate of how current actions will affect the agent's overall performance.

$$Q_\pi(s_t, a_t) = r_{t+1} + \gamma \max_{a_{t+1}}(Q_\pi(s_{t+1}, a_{t+1})) \tag{3-2}$$

As the agent interacts with the environment, it collects experiences in the form of $\langle s_t, a_t, r_{t+1}, s_{t+1}\rangle$ tuples from which the state-action Q-value is calculated by bootstrapping, estimating the future returns using the current state of the NN function approximator. The goal of the DQN training process is to tune the NN weights, $\theta$, so that its Q-value predictions are close to those experienced by the agent, thereby guiding the optimal action selection strategy. The authors introduced two key mechanisms to stabilize and improve the convergence of DQN algorithms with

NN function approximators. An iterative updating framework where the Q-values are adjusted towards target values that are only updated periodically by implementing a target network characterized by the weights $\theta^{tgt}$. Additionally, an experience replay buffer mechanism is implemented where the agent experiences are stored and from which random batches are uniformly sampled to execute gradient-descent-based training updates on the NN weights. The Q-learning iteration $i$ updates the NN weights $\theta_i$ to minimize the distance from its Q-value predictions to the observed ones using a mean squared error loss function (Eq. (3-3)).

$$L_i(\theta_i) = \mathbb{E}\left[\left(r_{t+1} + \gamma \max_{a_{t+1}}\left(Q_\pi\left(s_{t+1}, a_{t+1}; \theta_i^{tgt}\right)\right) - Q_\pi(s_t, a_t; \theta_i)\right)^2\right] \tag{3-3}$$

Where $Q_\pi(\cdot;\theta^{tgt})$ refers to Q-values estimated by the target network, which are synchronized with the Q-network parameters $Q_\pi(\cdot;\theta)$ every $C$ steps and held fixed between updates.

van Hasselt et al. (2016) showed that the original implementation of DQN tends to overestimate action values due to estimation errors of any kind that induce an upward bias as the maximization operator uses the same weights to both select and evaluate action. Double Deep Q-Learning (DDQL) was proposed to decouple the action selection from its evaluation by using the target network to estimate the target Q-value for each action. The loss function for the DDQL algorithms is shown in Eq. (3-4). The authors reported that this simple tweak reduced the overestimation bias significantly and improved the performance of the DQN agents.

$$L_i(\theta_i) = \mathbb{E}\left[\left(r_{t+1} + \gamma Q_\pi\left(s_{t+1}, \operatorname*{argmax}_a Q(s_{t+1}, a; \theta_i)\,;\ \theta_i^{tgt}\right) - Q_\pi(s_t, a_t; \theta_i)\right)^2\right] \tag{3-4}$$

Another critical component in Deep RL algorithms is the exploration strategy, to allow the agent to try different actions and improve its knowledge of different action values to make more informed decisions. A simple but effective and commonly used technique for exploration is the $\varepsilon$-greedy strategy (Ladosz et al., 2022). Following this exploration strategy, the agent picks a random action with probability $\varepsilon$, and with a probability $1 - \varepsilon$ picks the action with the maximum current action-value estimate. The open-pit truck dispatching system proposed uses the DDQL algorithm, where the agent is represented as a feedforward neural network that receives the state observations described, estimates the Q-values for assigning a truck to the available loader-destination paths and selects the best dispatching strategy.

Algorithm 2 describes the Double Deep Q-Learning implementation.

| **Algorithm 2.** Double Deep Q Learning |
|---|
| Initialize replay memory $D$ to an initial capacity $N$, with random actions. Initialize action-value function $Q$ with weights $\theta$ and target action-value function $Q^{tgt}$ with weights $\theta^{tgt} = \theta$<br><br>For each episode:<br><br>For $t = 1, \ldots, T$:<br><br>Observe environment state $s_t$<br><br>With probability $\varepsilon$ select a random action $a_t$ otherwise $a_t = argmax_a Q(s, a)$<br><br>Execute action $a_t$ in environment. Observe reward $r$ and next state $s_{t+1}$. Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in memory replay buffer $D$<br><br>Sample a random batch of transitions from the replay buffer $D$. |

For every transition in the batch, calculate target $y = r$ if episode ended at this step or

$y = r + \gamma Q_\pi \left( s_{t+1}, \underset{a}{\operatorname{argmax}}\, Q(s_{t+1}, a; \theta_i)\,;\ \theta_i^{tgt} \right)$ otherwise

For every transition, calculate loss $L = (y - Q(s_t, a_t))^2$

Update $\theta$ to minimize loss with respect to model parameters

Every $C$ steps set $Q^{tgt} = Q$

### 3.4 Case Study

The framework is evaluated in an iron ore mining operation. The mine maintains an estimate of the quality of the material, metal grade, being loaded at each mining face for planning purposes, and a database is available with historical records of equipment activity along with timestamps which are used to fit probability models to simulate their operating cycles. The mine feeds two ore processing plants, both requiring a total of 26,400 tonnes of ore per shift, with the first plant expecting a head-grade of $75\%$ and the second plant a head-grade of $65\%$ on their feed. Metal grade within $\pm 5\%$ of the target quality is deemed acceptable for the plant feed. Trucks hauling ore to the processing plants dump into a hopper with a capacity of 500 tonnes and an outflow rate of 2,200 tph, where trucks must wait for enough room in the hopper for their load before dumping their payload. All waste material is dumped into a single waste dump. A total of 5 loaders are available, 2 Hitachi 2500 with a bucket payload of 14 tonnes to work in ore mining faces, and 3 Hitachi 5500Ex shovels with a bucket payload of 22 tonnes for waste digging. Trucks loaded by any of the two ore shovels can be dispatched to any of the two ore processing streams, and trucks loaded by the waste shovels can only go to the waste dump. This defines seven possible paths the trucks can be assigned to for their operating cycle between each loader working in ore faces and the processing plants, and between the three loaders used for waste extraction and the waste dump. The mine has CAT785C trucks with a capacity of 140 tonnes and CAT793C trucks with a payload of 218 tonnes available. The RL agent is trained with a fleet of 20 CAT 785C and 14 CAT 793C, estimated to be required to meet the desired targets based on average productivity calculations, following common industry standards for truck fleet sizing. Figure 3-6 shows the general layout of the mine and the shared road network along which trucks travel. The mining faces areas contain different digging locations for each mining face on which shovels operate based on the mine plan.

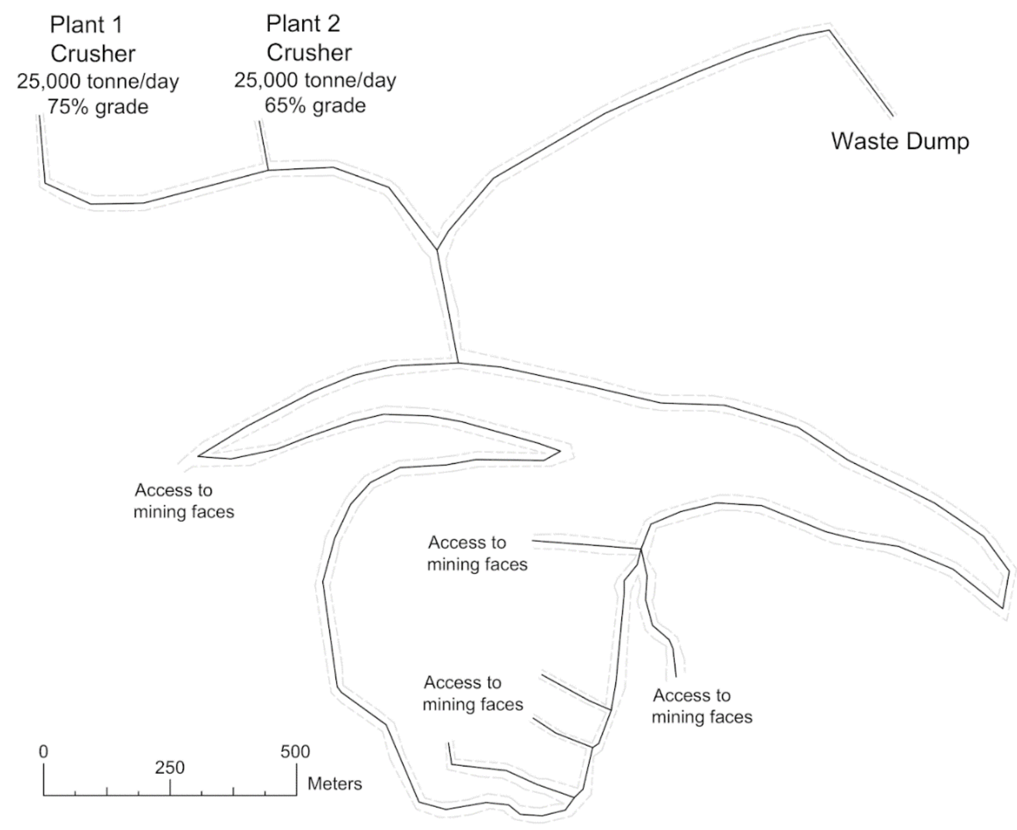


Figure 3-6. Case study mine road network, destination, and mining area locations.

The probability distributions fitted to each equipment activity based on the historical database are shown in Table 3-1. Different probabilities were fitted, and the best fit was selected based on goodness of fit tests. When sampling from the reference speed distributions of the trucks, a minimum value of 5 km/h was adopted, and a new sample was drawn in the case of sampling a value lower than it. The treatment of the historical data is not shown in detail as it is not the focus of the research and does not contribute any novelty or maintain confidentiality.

The NN truck dispatching agent is a feedforward network with 6 hidden layers and 600 neurons using ReLU activations at each layer. The hyperparameters used to implement the DDQL algorithm are presented in Table 3-2. The iteration update frequency of the target network hyperparameter was chosen based on best practices suggested for Deep Q-Learning of about 30 to 50 episodes between synchronization, which in this case represents roughly 25,000 steps.

Table 3-1. Activity time distributions for the equipment in the case study.

| **Activity** | | **Distribution** |
|---|---|---|
| Shovel bucket cycle time | Hit 2500 | Triangular (15, 26, 50) (s) |
| | Hit 5500Ex | Triangular (15, 29, 50) (s) |
| Truck spot time at the shovel | CAT 785C | Gamma (22.54, 1.39) (s) |
| | CAT 793C | Gamma (26.91, 1.36) (s) |
| Truck spot time at the crusher | CAT 785C | 30 (s) |
| | CAT 793C | 30 (s) |
| Truck dump time | CAT 785C | Normal (52, 6) (s) |
| | CAT 793C | Normal (55, 8) (s) |
| Truck reference speed | CAT 785C | Normal (35.6, 8.2) ($km/h$) |
| | CAT 793C | Normal (31.4, 9.4) ($km/h$) |

Table 3-2. Parameters for the training of the DDQL algorithm for the open-pit truck dispatching agent in the described case study.

| Parameter | Value |
|---|---|
| Replay buffer size | 200,000 |
| Batch size for NN training updates | 32 |
| Discount factor, $\gamma$ | 0.99 |
| Learning rate, $\alpha$ | 2.5 × 10-6 |
| Iteration update frequency of target network | 25,000 |

## 3.5 Results and Discussion

### 3.5.1 Training

Mining operations follow a tactical plan that defines the areas to be mined, the assignment of loaders, and productivity targets at different periods, where detailed plans are commonly prepared monthly. The RL-based truck fleet dispatching agent is trained to learn a policy based on the next production month in the case study. For this purpose, each training episode simulates a production day, where the configuration of the loader's assignments for each day is randomly sampled from the planned assignments over the next month. This randomization allows the agent to learn from a diverse set of operational configurations that are planned to be realized in the mining operation. Furthermore, the agent acts in the stochastic open-pit environment where every equipment activity takes a random time to complete based on the historical records available.

Five training runs were carried out with different random seeds to evaluate the proposed algorithm's training dynamics. Besides keeping track of the total episodic reward accumulated by the agent, the total production of the truck fleet, the average truck cycle time, and the average truck fleet utilization, fraction of the time spent on production activities, were also tracked as a problem-specific indicator. Figure 3-7 shows the average value obtained for these metrics across all runs with the bounds defined by $10^{th}$ and $90^{th}$ percentile at each episode. The simulation and deep reinforcement learning framework were developed and implemented in Python using the PyTorch package for deep learning. The training was carried out in the Google Colab cloud platform.

As training progresses, the agent consistently makes better-dispatching decisions achieving higher total episodic reward, which indicates that the truck fleet dispatching strategy achieves the desired productivity targets for the different pit configurations.

After about 1500 training episodes, where each episode requires about 600 dispatching decisions on average, the agent converges to a robust dispatching policy. The improvement in agent decision-making is also observed in the other key productivity metrics. The agent achieves lower truck cycle times as training progresses, managing the truck fleet to spend less time waiting in queues and increasing its utilization, and time spent in production activities. This is also reflected in an overall increase in total production obtained by the fleet dispatching policy. The decrease in the dispatching efficiency and total production observed, after around 500 episodes, is due to the agent learning to achieve the ore metal quality targets set at the processing plants. This reflects the complexity of open-pit production where competing mining, processing quantity, and processing quality targets require a trade-off.

The size of the Neural Network, the number of layers and neurons, is one of the most impactful parameters in the Deep Reinforcement Learning framework.

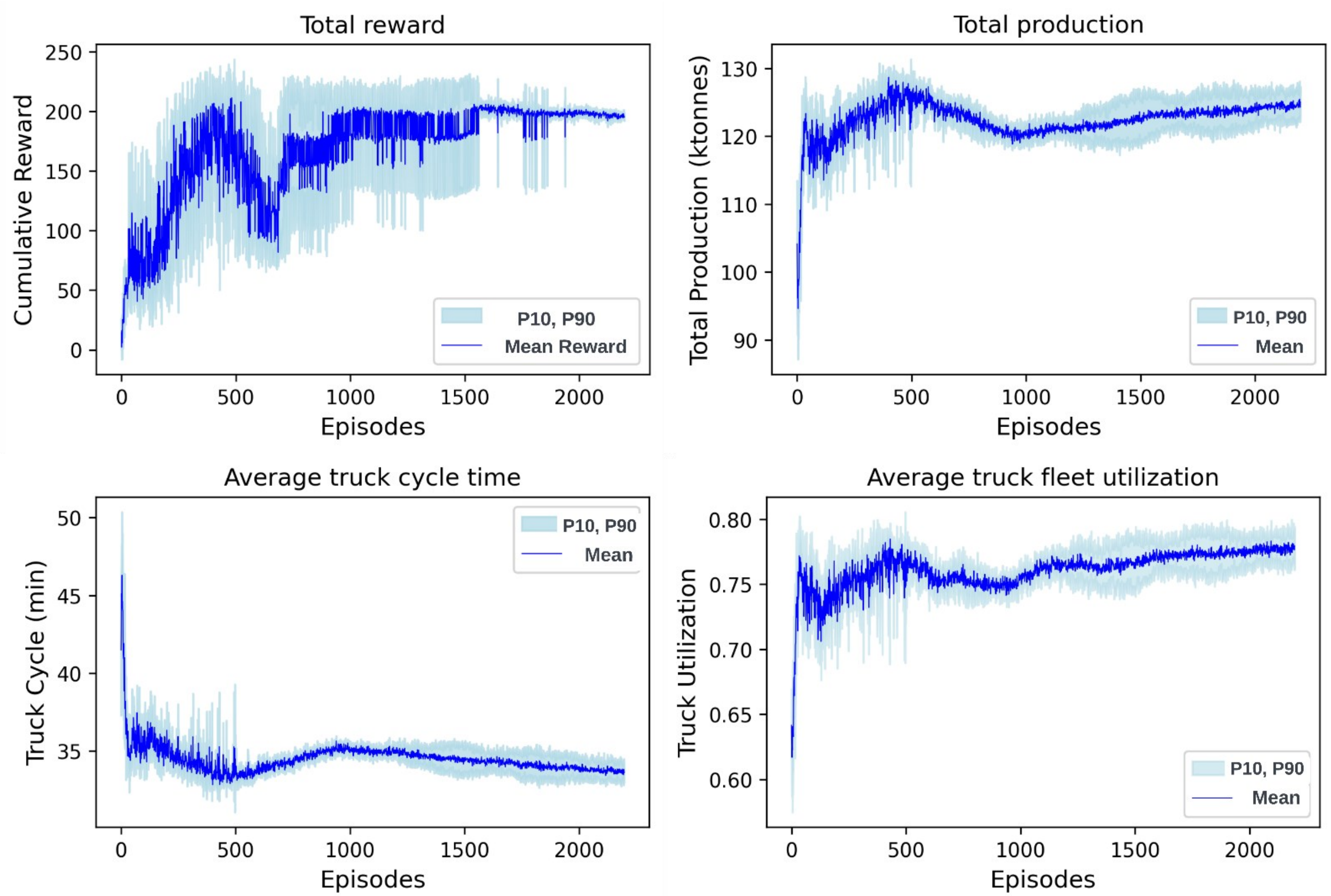


Figure 3-7. Training performance of the open-pit dispatching agent across 5 runs with different random seeds.

Larger networks have more function approximation capacity, allowing the learning of more complex correlations between action and high reward action selection strategies over a larger number of states. However, after some point, larger NN provides diminishing returns in terms of increased performance, with the possibility of incurring in worse performance once the NN becomes too large with too many parameters to train. The Neural Network size parameters were selected by experimenting with different sizes in terms of number of layers and number of neurons and observing the training dynamics.

Figure 3-8 shows the training performance of the agent with different NN sizes, from smaller networks to larger ones. The smallest network, with 3 layers and 200 neurons each, has the worst performance, converging to a low-reward strategy much earlier. This is predictable, as small networks have fewer parameters, hindering their capacity to learn more complex decision-making strategies. As the NN size increases, the training performance improves, however, the largest network tested, with 7 layers and 700 neurons each, does not provide an improvement but rather seems to start decreasing its performance with respect to the NN selected.

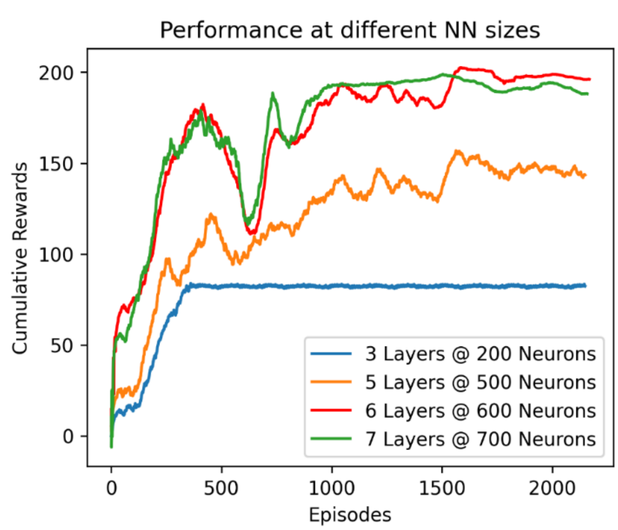


Figure 3-8. Training performance with different NN sizes. Selected NN size in red.

### 3.5.2 Evaluation

After learning a truck dispatching policy that achieves all goals, the agent performance is evaluated continuously throughout the entire production month based on the mine plan. In the evaluation run, once a loader has exhausted all material from its working area, it is moved to the next one on its schedule based on the plan. During this movement, no trucks can be assigned to it as it becomes unavailable. The realized pit configurations during the continuous simulation are not necessarily used for training, as mining areas may be exhausted at a different rate than planned as the agent found a more efficient strategy

. A total of 20 different simulations were carried out, and the average performance on different metrics and a 95% confidence interval were reported to assess its variability.

Figure 3-9 shows the agent's overall performance, as measured by the total reward collected at the end of each day during the month of production. The average performance of the agent is in line with the training performance, with some dips that are related to the exhaustion of mining areas where the loading equipment becomes unavailable for production until relocated to its new working area.

Two commonly used truck fleet dispatching heuristics were implemented as a benchmark to compare the behavior of the truck dispatching agent with respect to overall production and equipment productivity targets. The first is a shortest queue (SQ) heuristic, in which a truck is dispatched to the loader with the lowest number of trucks in its queue and then to the ore processing destination with the shortest queue if loaded with ore, with ties broken randomly. The second one is a target tracking (TT) heuristic in which a truck is dispatched to the loader with the lowest ratio of current rock loaded to its shift target and then to the ore processing destination with the lowest ratio of current feed received to its shift target if carrying ore, with ties broken randomly.

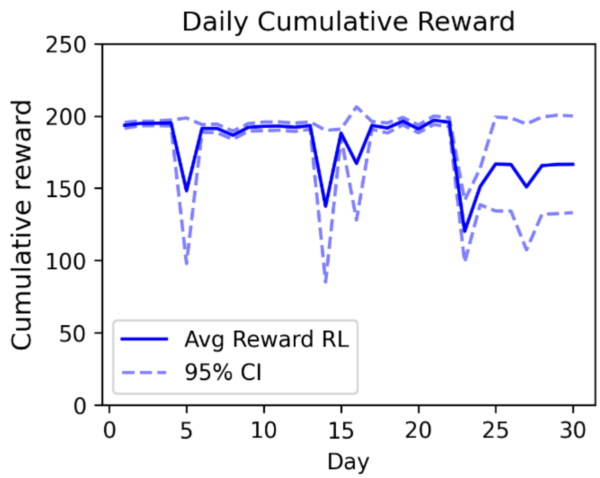


Figure 3-9. Average reward obtained by the agent throughout the month of production was simulated for evaluation.

These heuristics are commonly employed as a baseline to compare the implementation of more sophisticated dynamic dispatching algorithms in academic research, as well as being commonly employed by fleet managers in mines when no proprietary fleet management system is in use and are favored due to their simplicity, however, requiring manual and constant adjustment based on the experience of fleet dispatch engineers (Dendle et al., 2022). These heuristics are severely limited due to being myopic, as they only consider the next step in the decision and do not integrate the uncertainties in the system.

Figure 3-10 shows box plots of the production realized by the agent on each day across all verification runs and Figure 3-11 shows the average, and a 95% confidence interval, total shift production delivered to the ore processing plants and the waste dump in the evaluation simulation. The ore delivered to each processing plant shown here corresponds to the material dumped by the trucks at the crusher and not the flow processed by the plant, which is capped by the outflow rate of the crusher's hopper. This consideration is made to better gauge the impact of the dispatching strategies on overall rock material moved. The RL agent manages to beat all shift production targets and executes a robust dispatching strategy achieving similar performance across all simulations of the production month. The benchmark heuristics cannot achieve all shift targets as their decision-making strategies are myopic and cannot handle the system's multiple competing objectives and uncertainties. The SQ focuses the truck fleet on processing plant 1 at the expense of feeding processing plant 2, which could be due to the configuration of the pit system with the location of the ore plant 1 and the loaders favouring the formation of shorter queues biasing the heuristic.

On the other hand, the TT heuristic cannot achieve any of the targets, as its decision-making strategy is extremely myopic and leads it to an inefficient operation. Although not a fair comparison, since these heuristics only consider one single goal in a greedy manner, they provide a sensible benchmark to demonstrate the complexity of the problem. The RL agent performs adequately in the system with different requirements and service rates at both the mining and processing activities. Besides the overall shift production, each loading equipment and the ore processing plant's hourly productivity targets must be met to avoid excessive loader idle time or empty feeds. Figure 3-12 shows the average daily tonnes-per-hour (tph) productivity achieved by the agent over the simulated production month. The agent keeps all loaders busy, and the ore processing plants are fed as required. Overall, the dispatching strategy is robust across all simulation runs except for the last week of production of loader 4 and 5 working on the waste excavation, which could be due to the agent using any of those two shovels more at the expense of the other as both send their waste material to the same, and only, waste dump in the system.

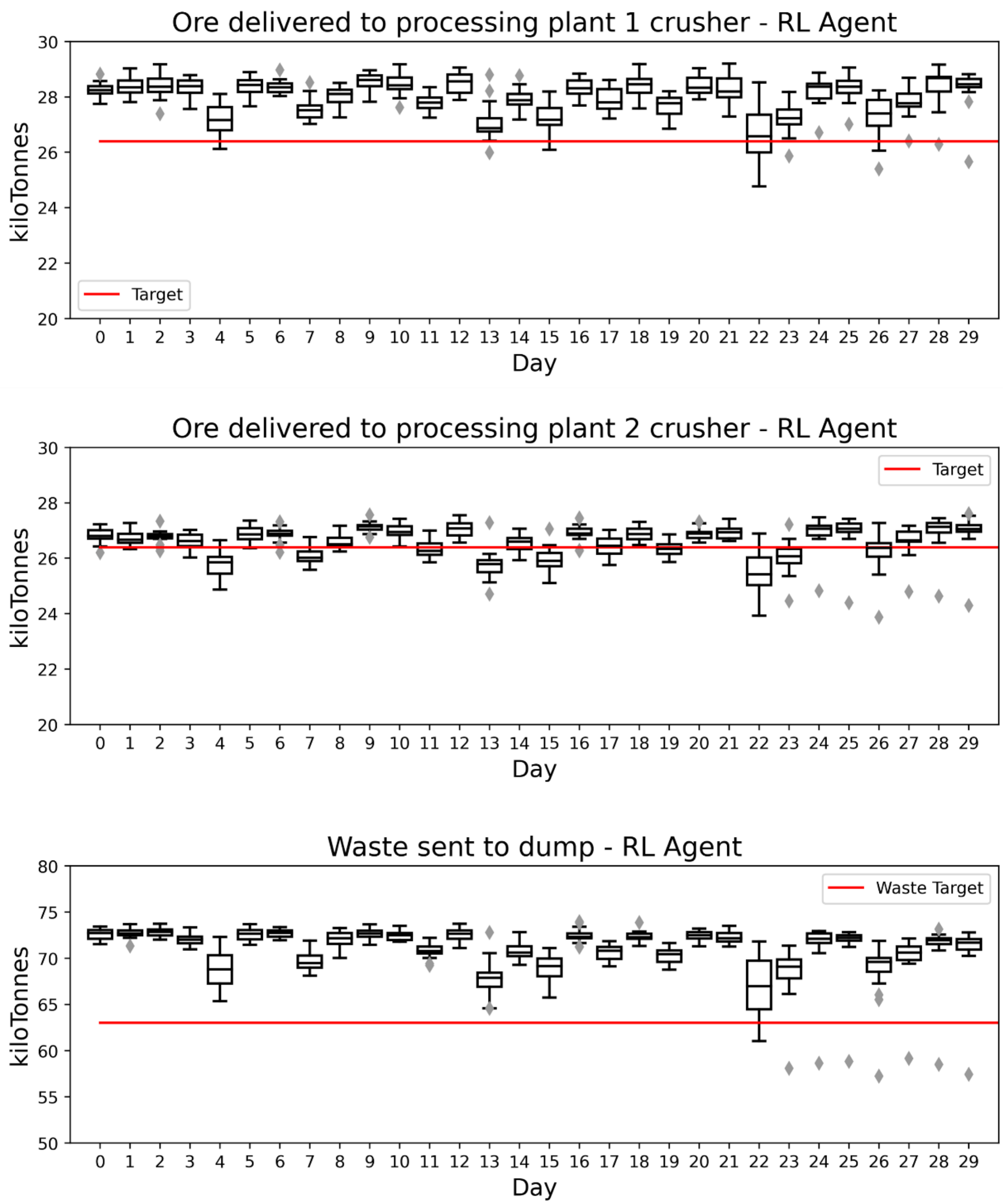


Figure 3-10. Box plots of production realized by the agent over the month sent to the different ore processing plants and waste dump.

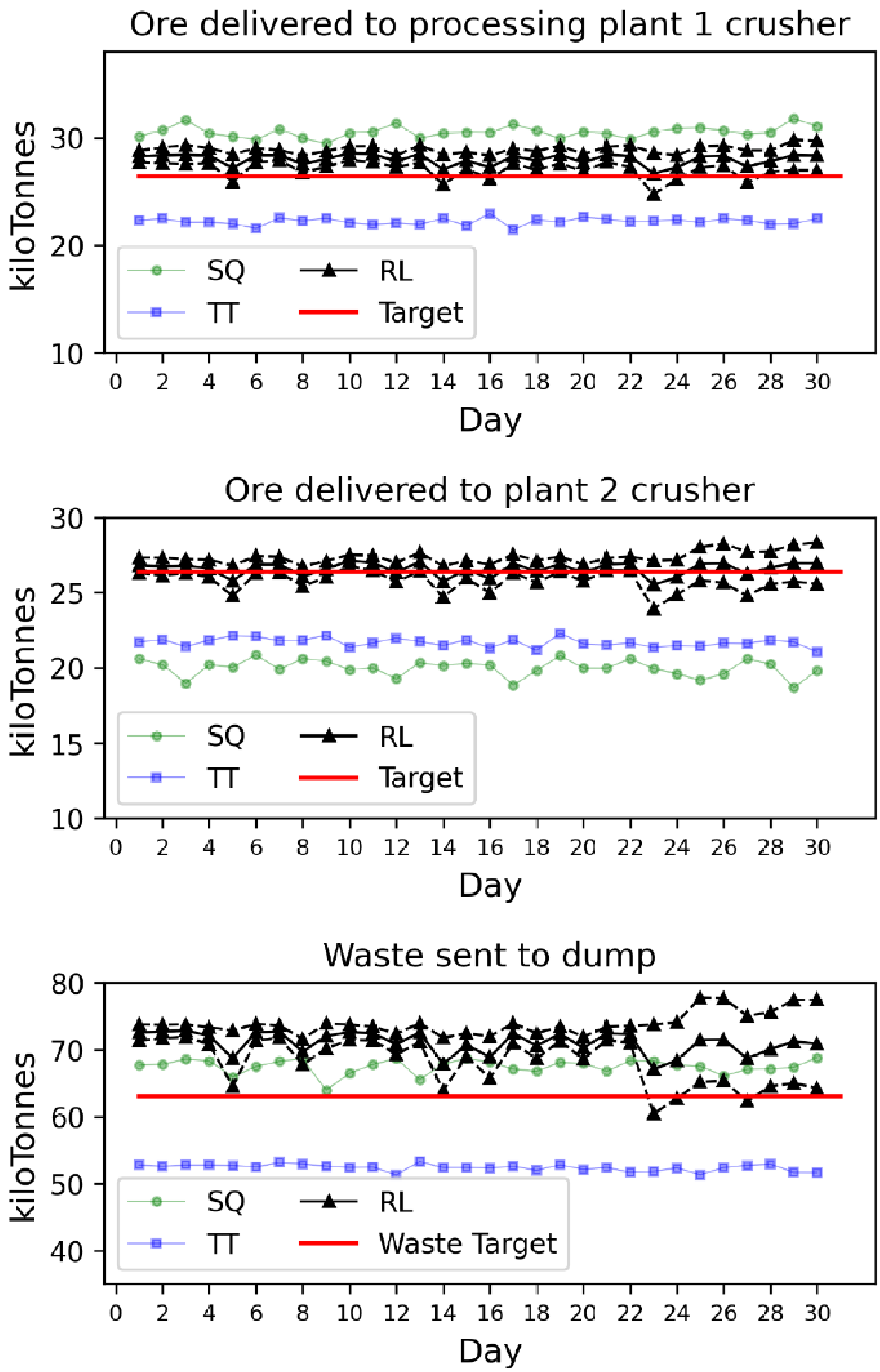


Figure 3-11. Average production realized by the agent over the month sent to the different ore processing plants and waste dump. Dashed lines around the RL performance indicate a 95% confidence interval. SQ: Shortest Queue heuristic; TT: Target Tracking heuristic, RL: Reinforcement Learning agent.

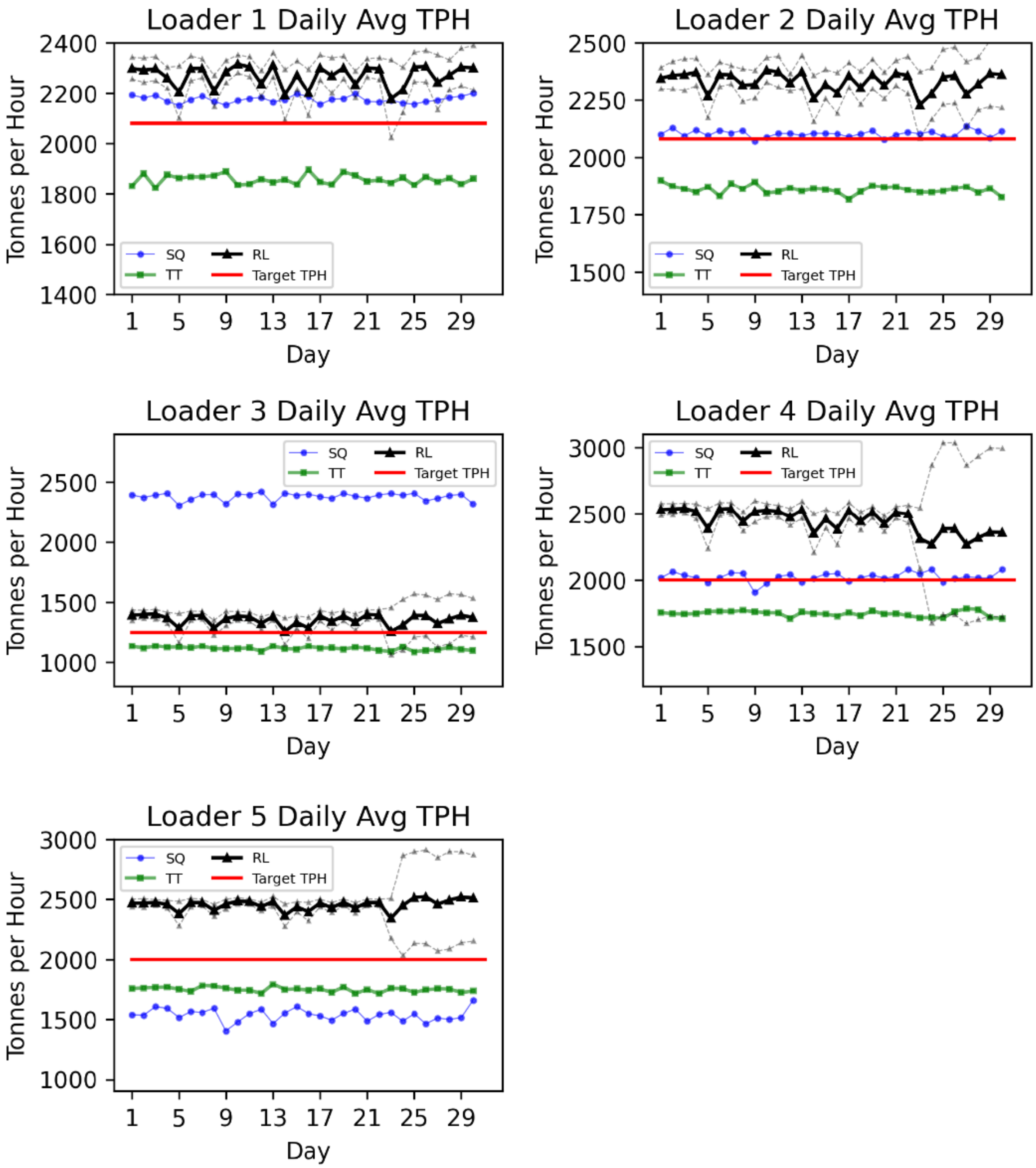


Figure 3-12. Daily average productivity rates achieved at each loader by the truck dispatching agent over the month for evaluation. Dashed lines around the RL performance indicate a 95% confidence interval. SQ: Shortest Queue heuristic; TT: Target Tracking heuristic, RL: Reinforcement Learning agent.

Based on the mine plan, the SQ and TT heuristics have a hard time meeting all targets, with the SQ strategy is not capable of handling different productivity requirements at the loaders, as it seeks to keep all loaders busy as possible with Loader 3 productivity significantly over its target.

The agent consistently accomplishes the ore processing plants feed quantity requirement across all simulations of the production month (Figure 3-13). Overall, the agent achieves all loaders and ore processing plants operational targets, ensuring an efficient operation. Finally, Figure 3-14 shows the average quality of the ore feed sent to the processing plant. The agent manages to keep the average iron ore grade sent to the processing plants within their respective desired limits hourly. This ensures that the processing plants operate within their designed metallurgical parameters.

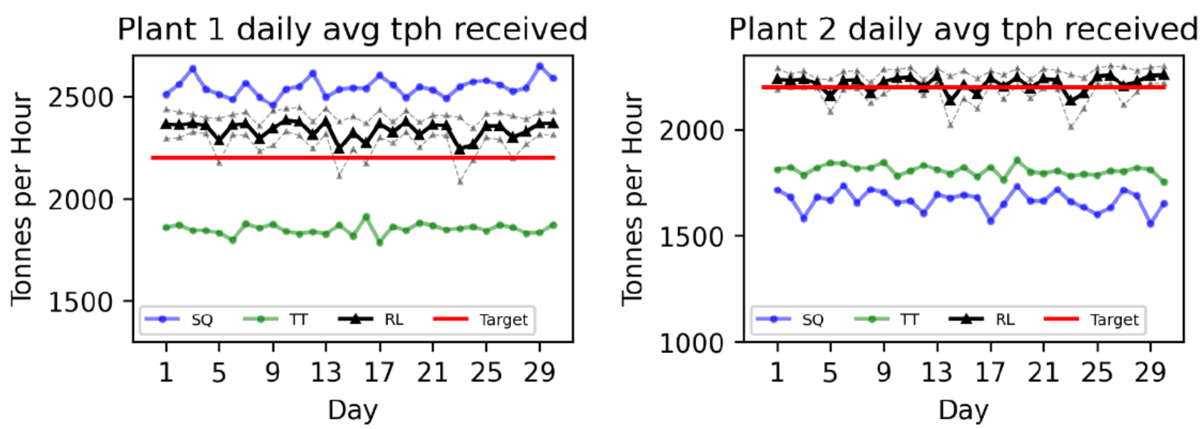


Figure 3-13. Daily average productivity rates achieved at each ore processing plant by the truck dispatching agent over the month for evaluation. Dashed lines around the RL performance indicate a 95% confidence interval. SQ: Shortest Queue heuristic; TT: Target Tracking heuristic, RL: Reinforcement Learning agent.

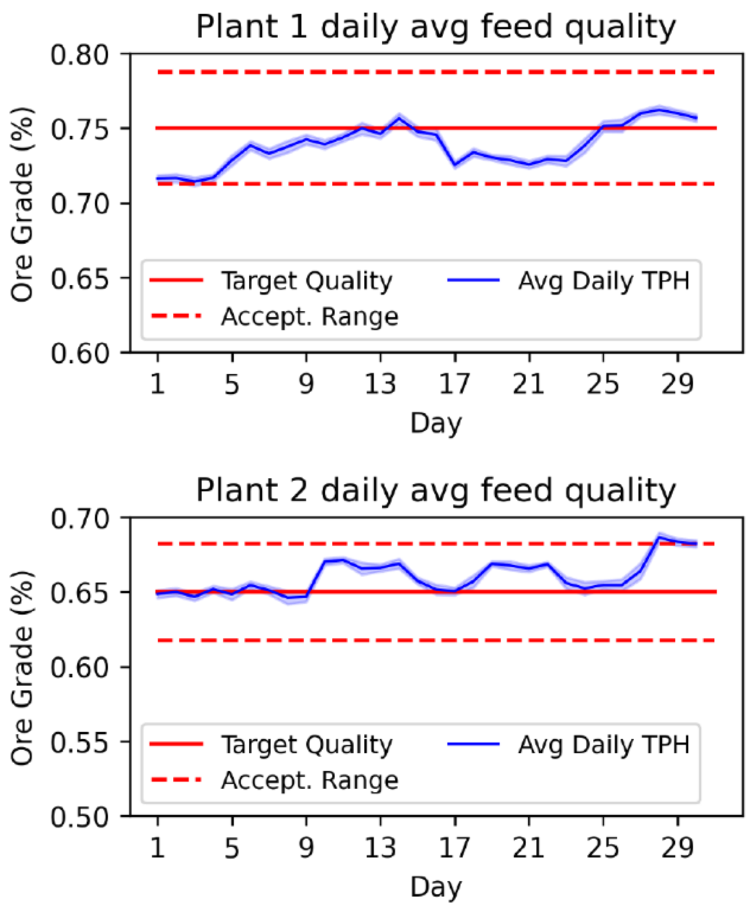


Figure 3-14. Daily average ore processing plant feed quality achieved by the truck dispatching agent over the month for evaluation. RL: Reinforcement Learning agent.

Individual trucks can become mechanically unavailable in mining operations due to failures and breakdowns of different components. Therefore, the RL truck dispatching agent must perform adequately with different truck fleet sizes. The month of production is simulated with different truck fleet sizes to evaluate the agent's performance, which is not retrained. Figure 3-15 shows the performance of the RL dispatcher at varying truck fleet sizes over the 20 simulations of the month of production. The scenarios considered are described below and were prepared considering the matching between shovels and trucks to avoid completely infeasible scenarios:

- 16T1; 10T2: 16 CAT 785C and 10 CAT793C trucks

- 17T1; 10T2: 17 CAT 785C and 10 CAT793C trucks
- 17T1; 11T2: 17 CAT 785C and 11 CAT793C trucks
- 18T1; 11T2: 18 CAT 785C and 11 CAT793C trucks
- 18T1; 12T2: 18 CAT 785C and 12 CAT793C trucks
- 19T1; 12T2: 19 CAT 785C and 12 CAT793C trucks
- 19T1; 13T2: 19 CAT 785C and 13 CAT793C trucks
- 20T1; 13T2: 20 CAT 785C and 13 CAT793C trucks
- BASE: 20 CAT 785C and 14 CAT793C trucks
- 20T1; 15T2: 20 CAT 785C and 15 CAT793C trucks
- 21T1; 15T2: 21 CAT 785C and 15 CAT793C trucks

The average cumulative reward obtained by the agent throughout the production month decreases overall as fewer trucks are available in the system, as expected due to less equipment to haul material out of the pit. On the other hand, adding extra trucks to the fleet does not improve the agent's performance as the total reward and production curves start to plateau, and increase in variability, indicating that the system becomes too congested. The agent manages to meet production targets at a lower truck fleet that it was trained with, which indicates that it is generalizing well and suggests that fewer trucks can be used in the system to achieve the targets consistently. The results of this sensitivity can also be used to size a stockpile or other material handling method for the ore processing plants. The over-production of ore would determine the required size of the stockpile, which could be selected to handle potential expansion scenarios of increased truck fleets.

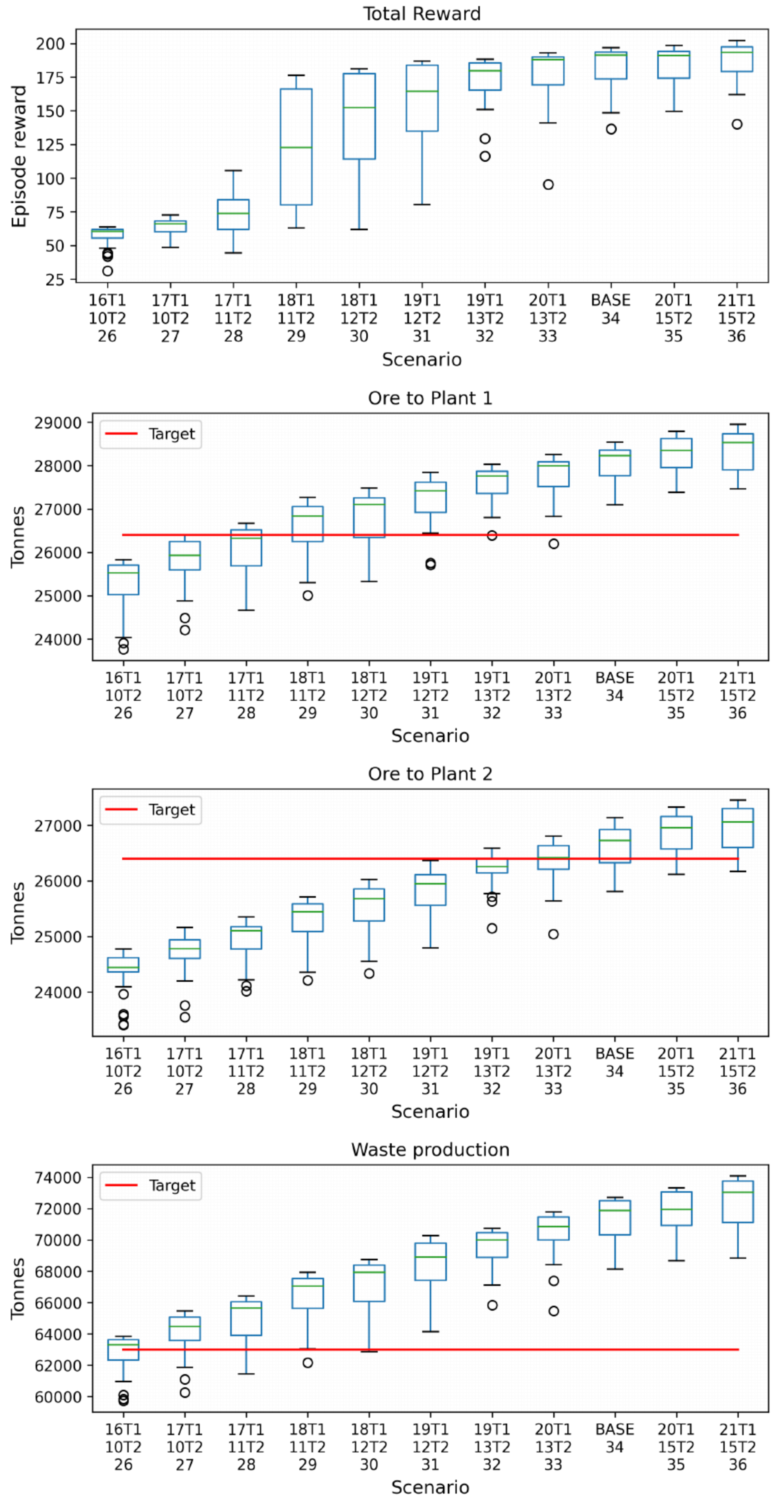


Figure 3-15. Box plot of the average daily production achieved by the agent under different truck fleet sizes. Each scenario is labeled based on the number of CAT 785C trucks, T1, and the number of CAT 793C, T2, in the fleet, with the total number of trucks below.

### 3.6 Conclusions

This paper developed an AI agent for real-time truck fleet dispatching in truck and shovel open-pit mines based on Double Deep Q-Learning and a neural network to approximate action-value functions. A discrete event simulation model was developed to serve as an environment to train the agent to meet mining production and ore processing feed quantity and quality targets. In the proposed framework, each truck is considered an individual agent that collects an experienced vector after completing an operating cycle. Each experience vector contains information about the pit's state configuration, which can be obtained from industry-standard fleet management and production tracking systems, the dispatch action is taken, its impact on the pit environment after execution, and a reward signal. The agent uses the environment's state description in the simulated training to build a correlation between state actions and rewards to guide decision-making; however, in deployment, it is information that should be available to the agent in real-time to use the learned model. The components of the state described here are all available in real-time with modern mining productivity and equipment tracking systems. Each truck agent pushes these experiences to a global buffer from which the neural network-based dispatch policy is trained. The training process consists of having the agent control the truck fleet in a simulation of production shifts where the activity time of trucks and shovels are randomly sampled from distributions modelled based on historical data. This allows the agent to learn a robust dispatching strategy that consistently meets all targets at each shift simulation. The reward function described in this research was designed to encourage the agent to achieve the three competing goals in the system as a single scalar objective. In the literature, multi-objective reinforcement learning approaches have been proposed to facilitate the design of reward functions for problems with conflicting objectives. However, the most common strategy adopted to deal with multi-objective problems is to expand the Q-values to a vector representation with each element representing the value under a particular objective, then applying a scalarization function that represents the user's preference between each goal to identify and learn optimal policies. This approach is equivalent to the one adopted here of designing a reward function encompassing multiple objectives. The field of multi-objective reinforcement learning is still relatively new but could help integrate additional competing goals in the learning loop.

In the training loop developed in this research, the agent's NN starts learning from scratch, from a random initialization, the task of dispatching trucks. However, the historical data of mining operations can be better used to pre-train the NN model in a supervised learning way to learn the truck dispatching policies in place. The same effect could also be achieved by initializing the experience replay buffer of the Deep Q-Learning algorithm to incorporate the historical records. Using the operation's historical records could help accelerate training and develop models with a strong baseline behaviour based on the operation's history.

The framework was evaluated on an iron ore open-pit case study over a month of production, where it achieved a satisfactory performance across all production targets over multiple simulation runs. The RL-based framework proves to be a promising technique for developing autonomous fleet management controllers as the industry moves towards adopting autonomous fleets.

Future work should be directed to evaluate the performance of other deep reinforcement learning algorithms successfully applied in similar domains, such as actor-critic and policy optimization methods. Moreover, applying the framework to integrate additional related decision-making capabilities, such as fleet preventive maintenance decisions and investigating the potential to learn dispatching policies to minimize fuel consumption and greenhouse gas emissions, would interest the mining industry.

# CHAPTER 4

# Shovel Allocation Planning in Open-Pit Mining using Deep Reinforcement Learning

*This chapter explains the development of a Reinforcement Learning (RL) based shovel allocation planning system for open pit mines to achieve production targets under equipment operational uncertainty. A case study is presented in an iron ore deposit where the trained agent manages to learn a policy to provide shovel allocation plans for a production quarter, 3 months, to achieve mining extraction targets.*

*The contents of this chapter have been submitted and are under review for a peer-reviewed publication by the International Journal of Mining, Reclamation and Environment.*

*R. Noriega, Y. Pourrahimian (2023), Shovel allocation and scheduling for open pit mining using Deep Reinforcement Learning, International Journal of Mining, Reclamation and Environment.*

## 4.1 Introduction

Short-term mine planning involves making operational decisions related to allocating equipment and resources on a monthly, weekly, or shift-by-shift basis. The overall long-term plan guides this type of planning. At these shorter time scales, mine operations are modeled with greater detail, considering the available equipment and different tasks required to execute the long-term strategic vision of the mine.

Model formulations for the open-pit mine short-term planning vary amongst researchers. However, these models typically aim to minimize deviations from production targets, minimize operating costs or maximize Net Present Value (NPV). They also involve a detailed mathematical representation of equipment interaction. Common formulations aim to obtain decisions on shovel allocations to mining areas and production scheduling of development and extraction activities such as drilling, blasting, and preparation of the working area. Mixed Integer Programming (MIP) methods predominate in the literature on open-pit short-term planning (Blom et al., 2019). Although the level of details incorporated varies amongst researchers, the short-term planning decisions usually involve the sequencing of mining blocks, or aggregations of blocks called mining faces or cuts, from the mineral resource block model and the allocation of equipment and other resources for it, to comply with the long-term strategic plan.

## 4.2 Problem Description

The system considered in this model is an open-pit mine with truck-shovel operations. The model aims to determine a shovel allocation plan for a time period of a quarter, or three months, of production. The shovel allocation plan details the assignment of shovels to mining faces during the time horizon. A mining face is a collection of blocks from the underlying resource model aggregated for operational purposes, defining practically mineable areas. This model assumes that each mining face has a homogeneous density and metal grade calculated from the underlying resource model. Additionally, the destination for the material of each mining face is predetermined. This is commonly defined before allocating equipment resources to mining activities as part of the mine planning process based on cut-off grade policies and production targets.

The problem is modeled sequentially by simulating the open-pit truck and shovel production environment, where every time step corresponds to the depletion of a mining face by a shovel and the requirement of a new assignment. The NN-based agent is called to provide the next assignment for the shovel in need by reading information about the system's state. The agent's goal is to consistently achieve tonnage production targets at each destination, including waste stripping targets, for the period considered.

## 4.3 Methodology

In this work, the allocation of shovels to mining faces is modeled as a sequential decision-making problem. The production system of the open-pit mine moves between steps defined by shovels completing the loading of materials in a mining cut and needing a new work assignment. The environment at time step $t$, has a shovel depleting its current mining cut assigned and requiring an action $a_t$, representing the movement to a new mining cut assignment. The environment executes the action, moving the shovel to its new working area and simulating the overall mine production until another shovel depletes its current assignment transitioning to time step $t+1$.

A highly customizable DES model of the mining production environment focusing on loading and haulage activities was developed. This model helps simulate the mining system's production and the impact of shovel allocation decisions. The DES models the interaction between loading and hauling equipment within the mine haul road network, loading and depletion of mining faces, and dumping into crushers and waste dumps with the possibility of accounting for stochasticity via probabilistic distributions for the different activities of the equipment. In addition, relevant production KPIs of the system are tracked for reporting, such as tonnage delivered at crushers, tonnage received by the plant, equipment cycle times, and utilizations, amongst other common KPIs used in mining. The DES was developed in Python, being completely open-source and easy to interface with other core ML and data analytics libraries. Figure 4-1 illustrates the general logic of the open-pit DES environment.

The DES starts with assigning shovels to their initial mining face, obtained by calling the RL agent, and simulates the movement of trucks along the road network to get loaded by the shovels and dump their payload at the set destinations for each mining face. When a mining face is depleted, the shovel needs to be relocated to a new mining face to keep production going; at this point, the AI agent is called to decide which of the available mining faces at that time the shovel will be assigned. Then, the shovel takes some time to move to the new mining area and resumes its operation after arriving.

Truck haulage is modeled by calculating the travel time through the different segments in the road network that form a path between a destination and a shovel, a haul route. An approach like Goris Cervantes et al. (2019) is used to estimate truck speeds using the equipment's historical records and the mine haulage network characteristics.

In this approach, empty flat haul velocity records for the different truck types in the system are collected and used to determine a probability distribution for empty flat haulage velocities.

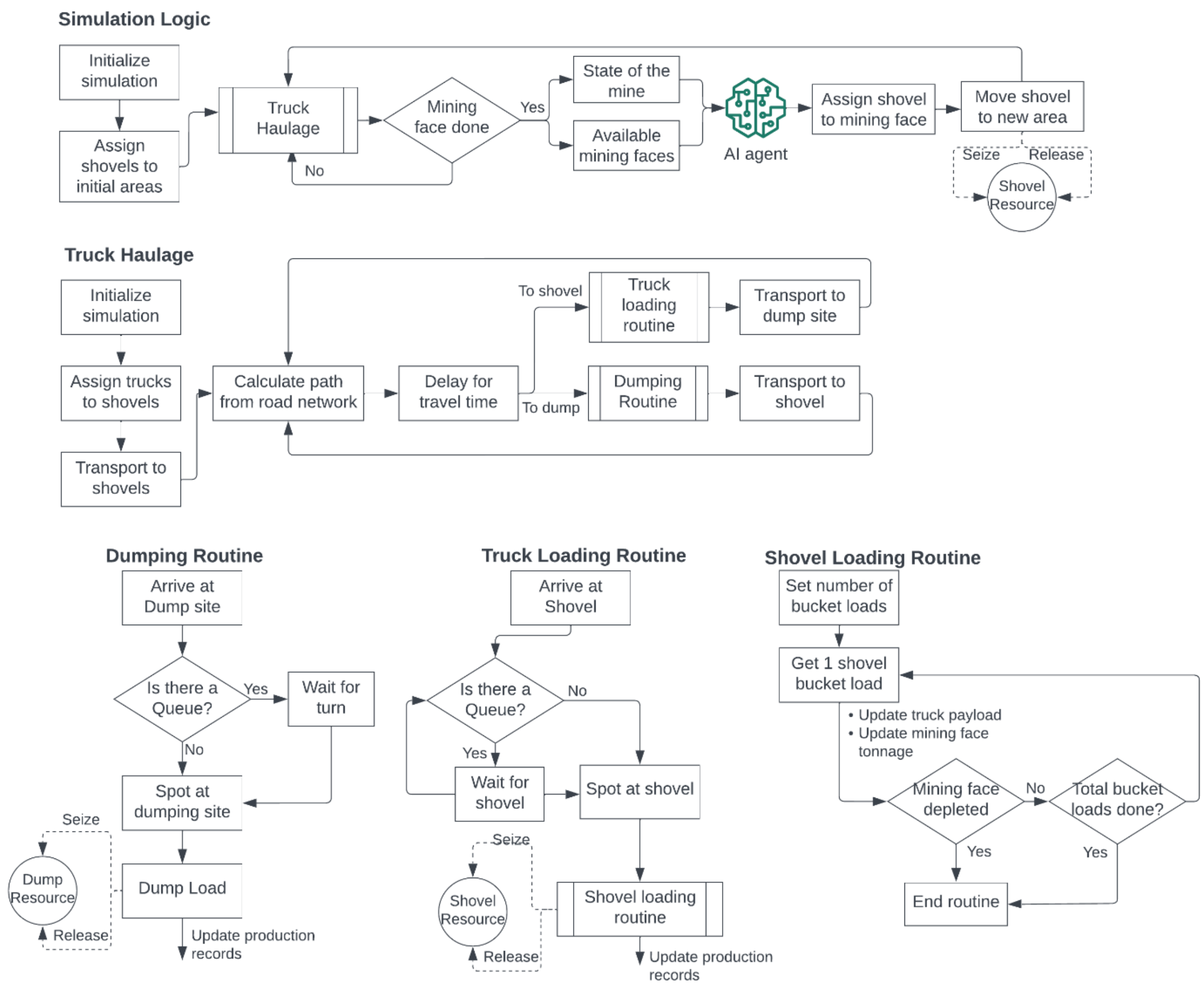


Figure 4-1. Open-pit DES simulation environment logic depiction. The main simulation logic controls the movement of shovels between mining faces assignments, and a Truck Haulage module simulates the movement of trucks across the road network between shovels and destinations.

These records should be obtained only from sections of the mine where empty haulage through a flat road is available without turning corners to define a baseline velocity distribution for the trucks. When a truck enters a road segment, the maximum achievable velocity based on the rimpull curve of the truck, its current weight, and the road segment total resistance is estimated along with the maximum achievable velocity of the truck as if it were hauling through an equivalent flat road using the same information. The ratio between the rimpull velocity and the equivalent flat haul velocity serves as an adjustment factor that accounts for the truck type's performance along a road segment. The truck is assigned a velocity equal to a velocity sampled from the empty flat haulage velocity distribution multiplied by the rimpull to equivalent flat haul velocity ratio. This data-driven methodology accounts for the impact of the road network characteristics on the truck haulage cycles and has been found to estimate productivity rates accurately (Upadhyay et al., 2021; Upadhyay et al., 2019).

When a truck arrives at the shovel, it waits in line for it to be available. Once available, the truck seizes the shovel and receives multiple bucket-loads of material from the mining face until it is full. Once the truck receives its load from the mining face, it travels through the haul road network until it reaches its destination, which could either be a crusher or a waste dump, depending on the type of material received. Crushers have a maximum ore processing rate that determines the amount of material they can handle in a given time, expressed as tonnes per hour. Trucks dump their payload onto a hopper por bin that regulates the flow to the crusher. The hoppers have a fixed capacity and a outflow rate, with trucks waiting until enough room in the hopper is available to dump the payload.

The open-pit production environment DES, as described above, was developed in Python using the SimPy library, an efficient process-based discrete event simulation framework which implements generic concepts of resources, managing its queues and waiting mechanisms, as well as handling the scheduling of stochastic events.

### 4.3.1 RL Framework for Shovel Allocations

During each iteration, the RL agent modeled as a NN, learns how to link the description of the system state and the actions taken with the accumulated reward. This helps the agent to recognize the most valuable actions and develop the optimal policy. The actions taken by the agent are defined as shovel allocations that happen when a mining face is depleted, and its assigned shovel requires a new mining face allocation. The state of the system at a given time $t$, when an action is required, must encode all features needed for the agent to learn its relationship with the desired objective to be maximized. For this purpose, the state of the system at time $t$ is encoded as a vector, $s_t$, with components $s_t = [MF_i, SH, t]$. Where $MF_i$ is a vector that encodes information about the mining faces in the system for the production period to be analyzed and is defined as $MF_i = [ton, g, dist, act, avlb]$. The component $ton \in \mathbb{R}^C$ is a vector that includes information about the remaining tonnage in each mining cut to be extracted expressed as a fraction, $g \in \mathbb{R}^C$ is a vector that includes information about the metal grade in each mining cut, $dist \in \mathbb{R}^C$ is a vector of the distance between each mining cut and its assigned material destination normalized by the maximum distance in the set, $act \in \mathbb{R}^C$ and $avlb \in \mathbb{R}^C$ are vectors of binary values that indicate whether each mining cut is active, currently being mined, and available, respectively, with all its precedences having been mined. $SH$ is a one-hot encoded vector that indicates which shovel should be assigned at this time step. It is a vector where all elements are zero except for one element with a value of one, indicating the specific shovel requiring an assignment. Finally, $time \in \mathbb{R}$, is the current simulation time expressed as a fraction. The action is represented as one hot encoded vector indicating which mining cut the shovel is assigned next. Figure 4-2 shows the general interaction between the state, RL agent, and assignment actions.

The reward function provided at each step directly serves as the objective function to be maximized by the agent during its interaction with the environment. As stated in the problem definition, the goal is to consistently meet production targets for both ore and waste at every location in the system. This should be done while minimizing the distance the shovels need to travel between mining assignments on different faces.

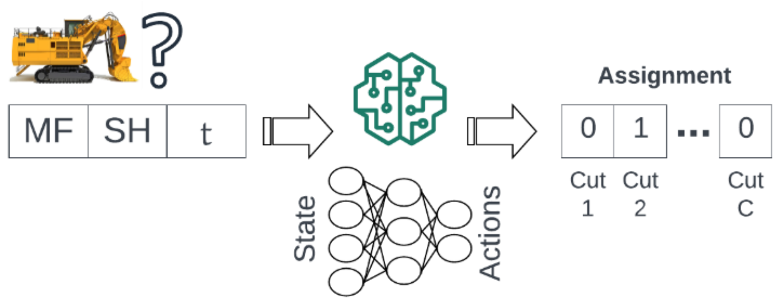


Figure 4-2. RL shovel allocation overview. When a shovel requires an assignment, the agent observes the system's state that serves as input to a NN model to predict the best mining cut for its next job assignment.

The reward at time step $t$, $r_t$, for assigning shovel $i$ to move from the depleted mining cut $c'$ to a new mining cut $c$, is defined in Eq. 4-1.

$$r_t = -\frac{K_{c',c}}{\max(K)} + \sum_h \sum_d \frac{\delta_d - dev_d^-}{D} \tag{4-1}$$

Let $K$ be a distance matrix where entry $K_{i,j}$ is the distance between mining cut $i$ and mining cut $j$ along the mine road network. The first term in the reward penalizes the agent for moving shovels along large distances. The penalty is divided by the maximum distance to keep it bounded between $[0,1]$. In the second component, the agent is rewarded or penalized based on the hourly production output of the system in the time between assignments. For every hour $h$ between time steps $t$ and $t+1$, the agent is penalized by the shortfall, or negative deviation $dev_d^-$, expressed as a fraction of the hourly production output relative to the desired one at each destination $d$. If the hourly production at destination $d$ satisfies the required production, then the agent is provided a $+1$ reward value, indicated by the binary variable $\delta_d$ that takes the value of $+1$ if this condition is met. This second part of the reward is normalized by the number of destinations in the system, $D$, to fix the upper bound of the overall reward at $one$ at every hour between assignments.

Figure 4-3 summarizes the general interaction of the agent with the environment to observe the system's state, execute actions and collect rewards.

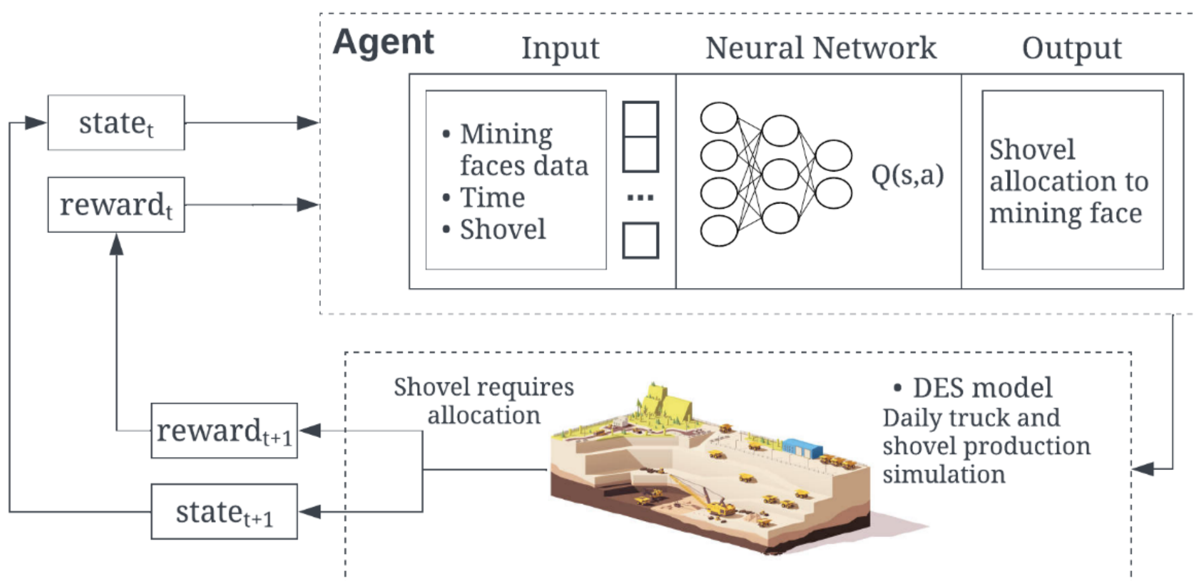


Figure 4-3. Agent-environment interaction for implementing the DQL shovel allocation agent.

### 4.3.2 Deep Q-Learning Implementation

Since the publication of the original DQN method (Mnih et al., 2015), many improvements have been proposed to enhance learning efficiency, significantly improving convergence, training stability and sample efficiency. Hessel et al. (2018) tested some of the most significant improvements to the original DQN and combined them into a single framework called Rainbow Deep Q-Learning, showing a significant increase in overall performance. The components adapted to general DQL described in Algorithm 1 are multi-step returns, double DQL, noisy networks for exploration, and a duelling networks architecture described next.

Sutton (1988) proposed using multi-step returns rather than single-step rewards to provide the agent with information about the impact of its actions over longer periods of time. The implementation is straightforward; once an action has been taken from a given state, the discounted cumulative reward observed after $n$ steps from that action is used as the target for the action-value prediction. Values of $n = 2$ to $n = 5$ have been commonly found to provide faster learning behaviours. The original DQN tends to overestimate action values leading to training instabilities and convergence problems. This is due to the maximization step in estimating the returns that serves as the target for training. Double Q-Learning was proposed by van Hasselt et al. (2016) as a solution to the maximization bias, where at every time step choosing actions is done from the Q-network $Q(s, a)$, but the target network $Q^{tgt}$ is used to evaluate the target for the updates. The epsilon-greedy strategy for exploration can be limiting in complex environments. Fortunato et al. (2018) proposed a simple but improved exploration strategy by adding noise to the weights of the NN agent rather than relying on the epsilon-greedy strategy. The noise in the NN model leads to some randomness in the agent's action selection but is adjusted automatically as an additional parameter by backpropagation during training. As training progresses, the NN can learn to ignore the noisy paths through the network at different rates in different parts of the state space, allowing for a form of state-conditional exploration. Wang et al. (2016) developed a novel NN architecture suited for value-based RL methods that features two streams of computation based on the observation that the action-value function $Q(s,a)$ can be decomposed as the sum between the value of the state $s$, $V(s)$, and the advantage of taking action $a$ from state $s$, $A(s,a)$, denominated duelling DQN. The advantage of actions can be interpreted as how much extra reward some particular action yields from a given state from the baseline state value. The duelling DQN architecture takes the feature vector and processes it through two independent paths: one for predicting the state's value and another for predicting each action's advantage. After that, the values can be summed up to obtain the Q-function. This architecture resulted in better training stability and faster convergence.

The integrated agent for shovel allocations to meet production targets in open-pit mining developed in this paper follows the basic DQN algorithm incorporating all the abovementioned improvements. The loss function to train the NN used is the MSE, and the optimizer used in training is ADAM, which has become a reliable NN optimizer that typically requires little tuning (Li et al., 2023). The combination of this particular loss function and the optimizer has demonstrated strong performance for DQN agents across various simple and intricate environments (Obando-Ceron & Castro, 2021).

The agent is a fully connected NN with four layers and 200 neurons each. Each layer has a Rectified Linear Unit (ReLU) activation function, which helps to speed up gradient calculation times and control vanishing/exploding gradient problems. Moreover, at every step, the gradient norms are clipped to a norm within 10 to stabilize training further, a common practice described by Zhang et al. (2020) to accelerate training; this means that rare extreme experiences will not cause extreme shifts in the NN parameters. The use of the Duelling DQN architecture means that there will be two network paths: one for predicting the state values $V(s)$ and another for predicting the advantage of taking each action from a given state $A(s,a)$. The NN agent architecture is depicted in Figure 4-4. The agent and the entire framework were implemented using the Python's Pytorch package.

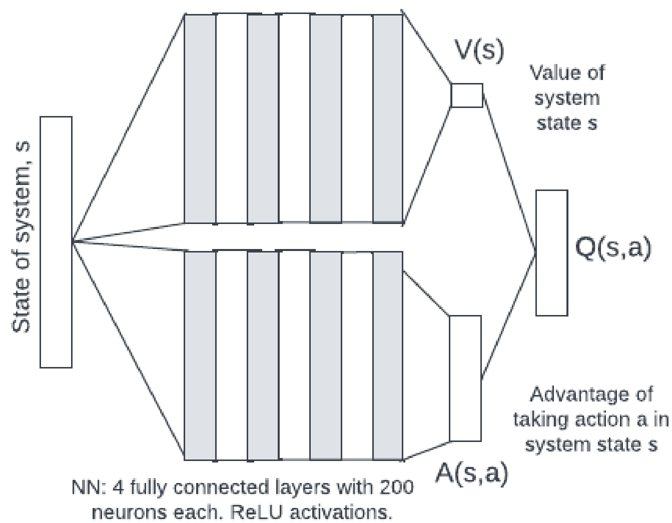


Figure 4-4. Shovel allocation Neural Network agent architecture.

## 4.4 Case Study

The shovel allocation AI agent proposed in this paper was tested in an iron ore mining operation through a case study. At the mining site, materials are loaded using five shovels. Two Hitachi 2500 shovels, with a bucket payload of 12 tonnes, are used for ore production. Meanwhile, three Hitachi 5500 Ex shovels are utilized for waste production with a bucket payload of 22 tonnes. A fleet of 33 trucks is available to haul the material from the pit to either of two crushers or a waste dump. The mine uses 18 CAT785C trucks, with a payload of 140 tonnes, to work with the ore shovels and 15 CAT793C trucks, with a payload of 218 tonnes, to work with the waste shovels.

Each ore shovel is allocated a fixed fleet of 6 CAT785C trucks, while each waste shovel has a truck fleet of 5 CAT793C. The framework could be extended in future work to optimize the integrated truck fleet and shovel allocation decision. The mine operates one 12-hour shift per day, seven days a week. The mine ore production target for both crushers' feed is 2,200 tonnes-per-hour (tph), or 52,800 tonnes of ore per day for the whole system, and a stripping ratio target of 1.8, or 95,040 tonnes of waste.

The agent's goal is to define a shovel allocation plan that would enable the crusher to achieve its production target in the upcoming quarter of three months. This plan would be based on the factors such as the mine layout, equipment performance, and availability of mining faces. The set of mining faces to be extracted in this period is based on the long-term strategic plan of the mine. This plan outlines broad sets of mining faces to be extracted over coarse periods of time without indicating a detailed sequencing. A total of 110 mining faces were considered, representing the set of mining faces available and expected to be mined in the next four months, an extra month of mining areas, to provide the agent with extra mining faces and allow flexibility to adapt the mine plan. Each of these faces has a set of physical precedences that represent the physical space required to start extraction, which is enforced by presenting to the agent only the available faces at each step when an action is required. Figure 4-5 shows a plan view of the mine layout. It includes the location of the crusher, waste dump, and access points to the mining faces. The distance to each mining face from the access points is assumed to be the linear distance between its digging coordinate and the closest access point in the road network.

To model the equipment production behaviour, statistical distributions were fitted to recorded historical data from an available equipment dispatch database to the activities comprising the equipment's load and haul operating cycle in the case study. Table 4-1 shows the statistical distributions fitted to the different equipment activities and parameters used in the DES model.

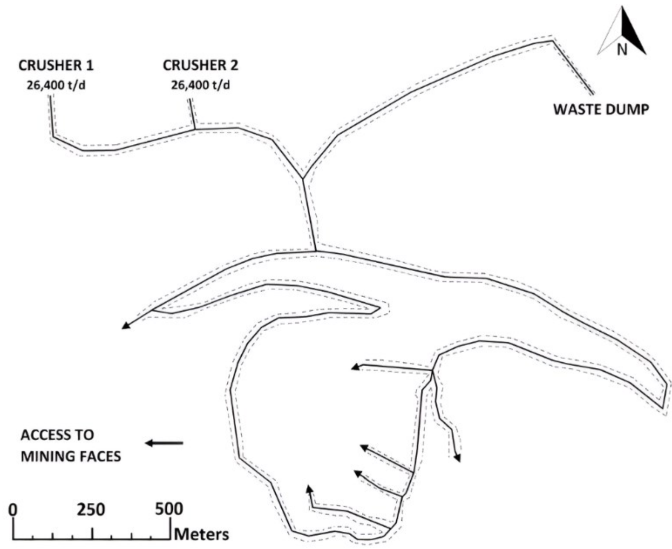


Figure 4-5. Mine layout for the case study.

Table 4-1. Distributions fitted to different activities in the productivity cycle of the load and haul equipment.

| Activity | | Distribution |
|---|---|---|
| Shovel bucket cycle time | Hit 2500 | Triangular (15, 26, 50) (s) |
| | Hit 5500Ex | Triangular (15, 29, 50) (s) |
| Truck spot time at the shovel | CAT 785C | Gamma (22.54, 1.39) (s) |
| | CAT 793C | Gamma (26.91, 1.36) (s) |
| Truck spot time at the crusher | CAT 785C | 30 (s) |
| | CAT 793C | 30 (s) |
| Truck dump time | CAT 785C | Normal (52, 6) (s) |
| | CAT 793C | Normal (55, 8) (s) |
| Truck reference speed | CAT 785C | Normal (35.6, 8.2) ($km/h$) |
| | CAT 793C | Normal (31.4, 9.4) ($km/h$) |

## 4.5 Results and Discussion

In DRL frameworks, the NN agent is trained by collecting experiences from interaction with the environment over multiple episodes. An episode here refers to a 91-day simulationof the open-pit production environment, equivalent to three months or a production quarter. During each episode, the agent allocates shovels to available mining faces as they complete their previous assignment at each time step and receives a reward based on the proposed reward function. The environment is restarted once an episode is completed, and a new, independent simulation episode is started. The total cumulative reward obtained at the end of each episode provides a measure of the agent's performance at allocating shovels to mining faces to achieve the daily production targets. When the NN agent is first trained, its weights are randomly initialized, and its performance is initially poor. As the training progresses and the NN is updated to learn effective decision-making strategies, the agent's performance is expected to improve with each episode Table 4-2 presents the parameters used in the training of the DQL agent.

Table 4-2. Parameters used to train the DQL algorithm for shovel allocation in open-pit mining.

| **Parameter** | **Value** |
|---|---|
| Replay buffer size | 8,000 |
| Batch size for NN training updates | 32 |
| Discount factor, $\gamma$ | 0.99 |
| Learning rate, $\alpha$ | 5 × 10-4 |
| Iteration update frequency of target network | 1,000 |
| $n$ for multi-step returns | 4 |

To ensure consistent training dynamics, the training of a DQL agent was evaluated through five runs using different random seeds to generate simulated events. Figure 4-6 shows the average cumulative reward and upper and lower bounds defined by the standard deviation across the five training runs.

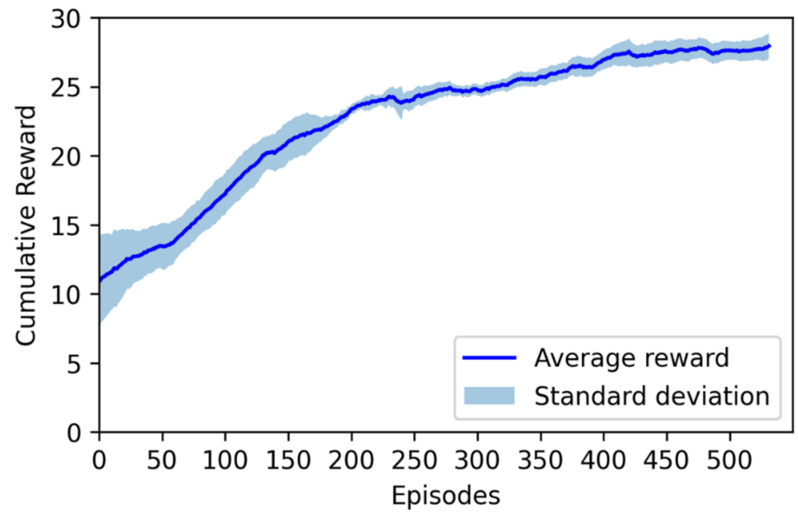


Figure 4-6. Training performance of the shovel allocation DRL agent across five runs. An episode consists of a 3-month simulation of the case study truck and shovel open-pit mining operation.

Throughout the 3-month production period, the cumulative reward reflects how well the agent meets production goals and reduces total shovel movement distance. During the initial training phase, the agent's performance is low due to the limited experiences the NN have had to train on. As the training continues, the neural network gradually becomes better at choosing actions with higher value. This results in a consistent improvement in the agent's performance in all runs. Once the agent has undergone approximately 400 episodes, it acquires a strategy that yields consistent results. Consequently, the training is discontinued since no further improvement can be observed. By the end of the training, the agent can step through a simulation of the mining production environment and allocate shovels to achieve a high total reward at the end of each 3-month simulated period.

The sensitivity of the performance of the shovel allocation agent against different NN sizes was evaluated. The size of the NN is one of the most impactful parameters in a Deep Reinforcement Learning framework, as larger networks allow the learning of more complex relationships between actions and rewards, however, after some point, larger NN hinder performance as they have too many parameters to train. Figure 4-7 shows the training performance of the agent with different NN sizes.

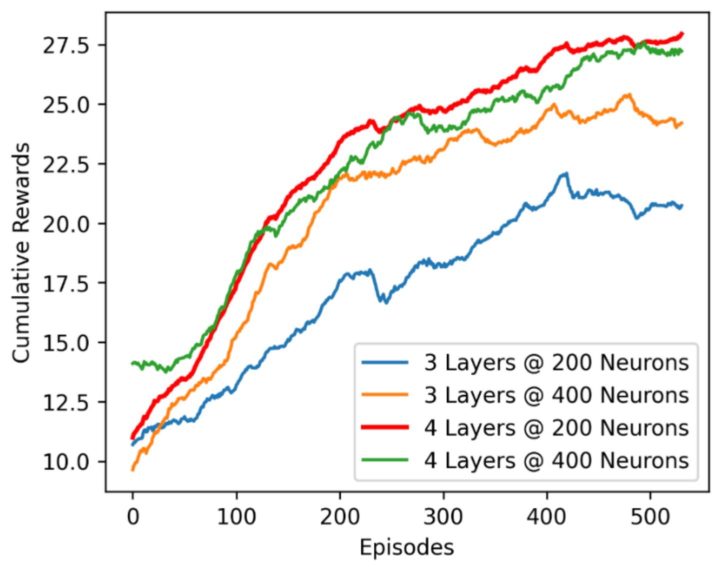


Figure 4-7. Performance of the shovel allocation agent under different NN sizes. Selected NN in red.

It can be observed how smaller networks achieve lower cumulative returns overall, over the same training period. The performance of the agent increases with larger networks, however, increasing the size after a network with 4 layers and 200 neurons each does not provide any improvement, whilst adding more parameters too train. Therefore, the selected NN size was of 4 layers with 200 neurons each.

In order to evaluate the agent's performance regarding production targets, a new simulation run is performed with the final trained agent selecting shovel allocations at each step. The shovel schedule is recorded, showing where they will be used for mining, when they will start and finish, and how much ore and waste will be produced at each location per day. Figure 4-8 shows the shovel allocation plan proposed by the agent. The numbers inside the bars indicate the mining face where the shovels work on.

Figure 4-9 shows the weekly ore and waste production. This information is calculated by aggregating the daily production recorded over each week of the 3 months, consisting of 13 weeks or 91 days. The agent has generated an allocation for shovels that will enable the production of enough ore to meet the targets for the next three months. All waste stripping targets were successfully met during the first two months of production. However, in the last month, the agent experienced some difficulties due to the expansion of the open-pit and the increasing distance between mining faces and destinations. These challenges could be resolved through the addition of more haul trucks. The performance of the DRL agent is considered satisfactory in achieving the production goals during the time horizon considered.

Figure 4-10 shows the average daily tonnes-per-hour (tph) of ore delivered at each crusher throughout the 3-month period. The ore delivered at the crusher by the truck fleet includes ore in the crusher’s bin with a capacity of 500 tonnes. The shovel allocation plan manages to keep the crushers operating at the target capacity over the 3-month production period.

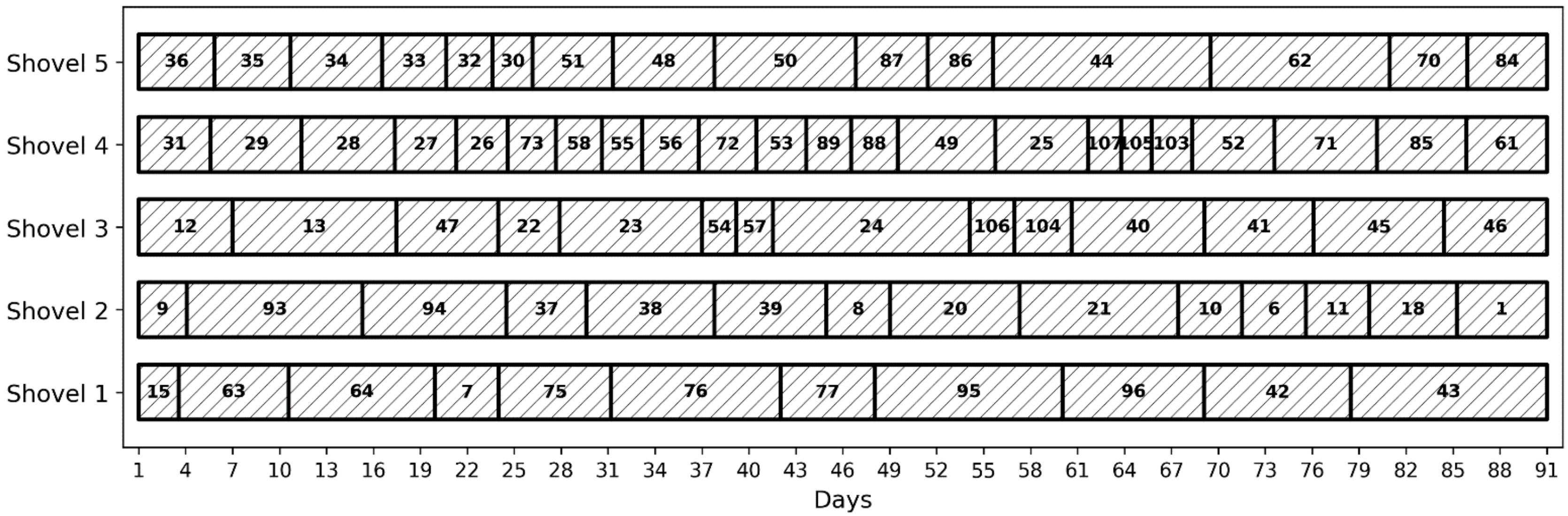


Figure 4-8. Shovel allocation plan proposed by the agent for the next production quarter.

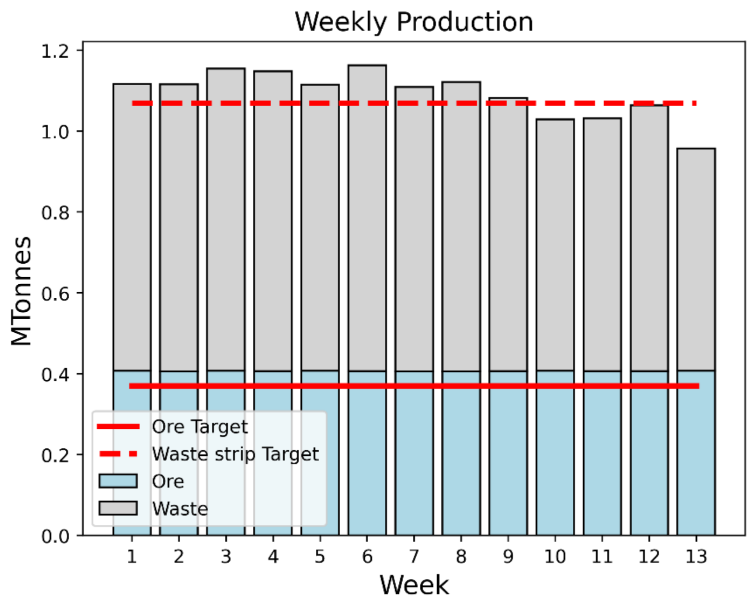


Figure 4-9. The DRL agent obtained Weekly ore and waste production. The ore production refers to the truck fleet's delivery at the processing stream.

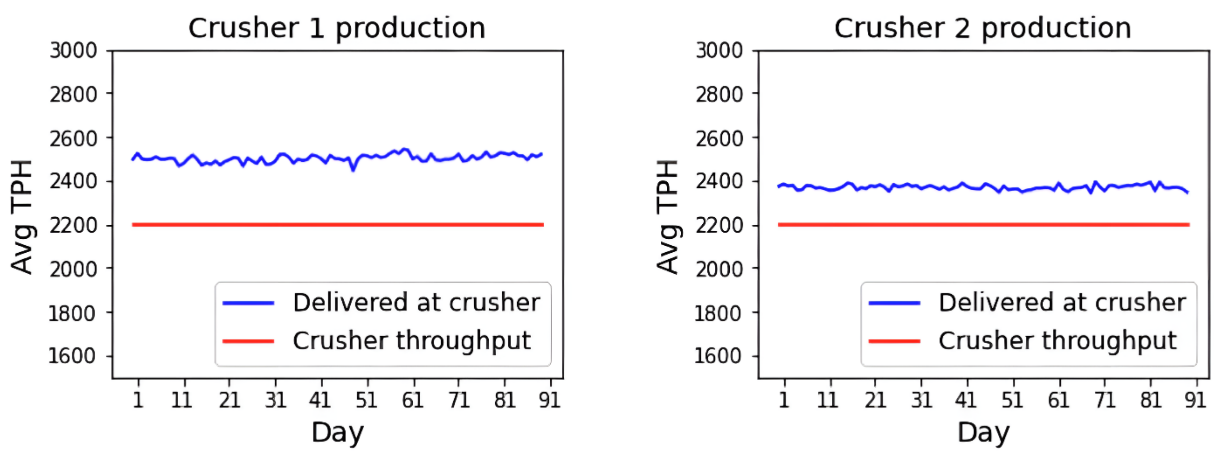


Figure 4-10. Daily average tonnes-per-hour of ore delivered at each crusher by the shovel allocation plan generated by the agent.

## 4.6 Conclusion

DRL algorithms are relatively new techniques combining machine learning advances, such as Neural Networks (NN), with decision-making optimization. It allows the use of simulators to provide samples, or experiences, of actions and their impact on the environment to learn optimal decision-making strategies. This in turn allows to incorporate multiple sources of uncertainties into the dynamics of the environment simulator. In this paper, a DRL algorithmic framework was developed for generating a shovel to mining faces allocation plan for truck and shovel open-pit mining operations over a 3-month period. A DES model of the open-pit production environment was developed. The model takes into account uncertainties that may arise during the truck and shovel production cycle. A DQL algorithm that incorporates recent developments in the DRL field, was proposed. The algorithm trains a shovel allocation agent that plays the simulation repeatedly, learning from the experiences. The goal is to develop a shovel allocation plan that meets ore and waste stripping production targets. The proposed framework was evaluated through a case study on an iron ore open-pit operation. The DES model was created by analyzing the equipment dispatch records to establish statistical distributions of the equipment performance. Information on the mine haul road network and mining faces was also utilizedin its development. The DRL agent was trained to provide a shovel allocation plan over a 3-month period. After about 550 independent simulations of the production operations, the agent learned a shovel allocation plan that consistently achieved the production targets by the end of the training process.

The main limitation of the framework is that the decision-making strategy learned directly depends on how detailed the environment is, and how representative of the real-world operation it is. This is also a common problem of other optimization frameworks, as every model requires a set of assumptions. Future research in this direction should include truck fleet selection optimization for each shovel allocation decision and letting the agent decide the destination of the mining faces to incorporate mine planning decisions into the optimization framework. Moreover, the DRL framework has many hyperparameters that could be difficult to understand

and choose without prior knowledge of AI and RL. This can result in a significant amount of time required to fine-tune the algorithms. Although the field of DRL is relatively new, it has proven to be effective to build decision-making systems in challenging and uncertain environments, and the applications to mine planning and scheduling are promising.

# CHAPTER 5

# SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

*Chapter 5 encompasses the summary and conclusion, highlighting the significant contributions of this research and providing recommendations for future endeavors.*

## 5.1 Summary of Research

The open-pit production environment is a highly dynamic system where performance is driven by complex and uncertain interactions between different types of equipment as they complete their operating cycles. A key step in the open-pit tactical planning stage is allocating shovel to different mining faces for extraction and dispatching hauling equipment in operations. The complex details and high uncertainty of the open-pit environment make it challenging to account for every operational detail within a mathematical optimization model, which is the preferred method to propose solutions for these problems (Blom et al., 2019; Moradi Afrapoli & Askari-Nasab, 2019). The use of simulation models in mining operations has been increasingly adopted as a tool to evaluate the viability and robustness of mine plans and other strategic decisions under different sources of uncertainty. However, simulation models have not been fully incorporated in the optimization process for tactical open-pit mine planning decision-making, where it is currently used only as a post-optimization evaluation tool.

Artificial Intelligence (AI) methods such as Deep Reinforcement Learning (DRL) have been successful at learning optimal sequential decision-making strategies in highly dynamic and uncertain engineering systems, such as production plants, transportation logistics, and natural resource extraction. The DRL framework uses a simulation model of the system to be optimized, a digital twin, to let a neural network-based agent interact with and receive feedback to train and learn an optimal, stochastic, decision-making policy. This research proposed a framework to use a Discrete Event Simulation (DES) model of the open-pit production system as a virtual environment, that integrates uncertainties across the entire loading and hauling operating cycles, to let the DRL agent interact with and learn optimal decision-making strategy to achieve some set goals. Two main applications were presented using this framework for the dynamic dispatching of a truck fleet (Chapter 3) and for allocating loaders to mining faces for extraction planning (Chapter 4).

The scope and assumptions of the DES model are described below.

- Only shovel and truck operations are considered; no ancillary equipment or operations such as drilling and blasting are considered. This simplifies the environment but keeps the level of detail required for truck fleet dispatching and shovel allocation decisions for the short-term plan.
- The mining faces, aggregation of individual blocks from the resource model, are considered as a whole, and as shovels dig, the loaded bucket material has the same characteristics as the whole mining cut, such as density and grade. No finer divisions are considered.
- Shovel's bucket cycle time, truck spotting times, dumping time, and velocities are probability distributions.
- The mine road network is modelled as connected edges along which trucks travel, where each edge has a given gradient. The mine road network is assumed to be split into edges, or individual road segments, of approximately equal grade between intersections or destination nodes, holding either a loading or a dumping area.
- Processing plant components are not modelled, and trucks dump into a hopper that feeds the processing plant, which is considered the end point of the truck operating cycle and used to record production statistics. Hoppers have a given capacity to hold material delivered from the mine, and a fixed output flow. If the hopper does not have enough

capacity for the upcoming truck payload, this has to wait until there is enough room to dump.

The scope and assumptions of the truck fleet dispatching DRL framework are described below.

- In the DES model, truck movement is simulated across each individual road segment. Each road segment holds a queue of trucks to account for potential bunching of trucks.
- The agent is tasked with achieving loader and destinations, both ore processing streams and waste stripping, shifting production targets and keeping the ore feed quality within a given range.
- The agent can decide to which ore processing destination to dump its payload. There are no fixed ore processing streams for the material coming from each ore mining face. Material from a given ore mining face can be dumped into any ore processing stream by any truck cycle.
- The shovel movement schedule is assumed to be known beforehand. During the simulation, once a shovel has depleted a mining face, it takes some time to move to its next assignment, during which it is marked as unavailable for new truck assignments until it reaches its new mining area.
- Equipment failures and breakdowns are not explicitly incorporated in the decision-making process, as rerouting or reallocations. However, the agent's state observation uses current assignments to suggest allocation decisions and a mask to avoid incompatible loaders. Therefore, by indicating a shovel as unavailable or removing a truck from the system, the agent can suggest actions with the reduced fleet.

The scope and assumptions of the DRL shovel allocation framework are described below.

- The agent is tasked with achieving shift production targets over a production quarter, three-month horizon. Targets as expressed as tonnes delivered by the truck fleet to each destination by the end of each shift.
- The mining faces have a fixed destination policy.
- Each shovel has a fixed truck fleet throughout the proposed time horizon.
- Truck movement is not simulated at each road segment but rather by calculating the time required considering the entire path between its source and destination. The collection of road segments to travel is retrieved, and a total travel time is calculated by sampling the velocity distributions and correcting for the road segment gradients. This allows faster simulation runs to model the three months at each iteration.

## 5.2 Conclusions

The key conclusions from this study are summarized below.

- The literature review conducted in this research identified the emergence of Machine Learning (ML) and AI based methods as hot research topics in the area of open-pit mine planning. This coincides with the emergence and rapid adoption of AI techniques in other engineering fields and poses an opportunity to adapt these to help solve core problems in open-pit mine planning and operations.
- There is a gap in integrating equipment uncertainty in the optimization process for short-term open-pit mine planning applications. DES models are commonly used to evaluate the performance of a mining production system under equipment activity uncertainties;

however, it is not directly used in the optimization process. Instead, the DES model is used after a schedule or equipment plan has been obtained to evaluate its performance under uncertainty.

- A DRL approach is proposed to fully integrate operational uncertainties, from loading and hauling equipment activities and their impact on the system's productivity, in the scheduling and equipment allocation problems in open-pit mining. The DRL uses a DES model of the open-pit truck and haulage activities to interact with and learn a robust equipment scheduling and allocation strategy.
- The proposed DRL framework for dynamic truck dispatching uses a DES model of the loading and hauling activities, considering the truck fleet movement along a shared road network, to learn a robust truck dispatching policy that achieves shift production goals, equipment productivity goals, and ore processing stream quality goals.
- The dynamic truck dispatching agent was verified in a case study where it controlled the truck assignments for a month of production in the DES simulated environment, which exceeded the training episode lengths of one production shift. The verification showed that the agent learned the dynamics of the open-pit environment and provided consistent behaviour that achieved all competing goals across multiple realizations.
- The proposed DRL framework for shovel allocation interacts with a DES model of the loading and hauling activities in an open-pit mining environment to learn a robust shovel allocation strategy that achieves the daily production goals over a production quarter, three-month time horizon.

## 5.3 Contributions of the Research

The main scientific contribution of this research to the current state-of-the-art is the development and verification of a practical framework for uncertainty-based tactical open-pit planning based on DRL. This research contributes to creating new knowledge, understanding, and innovative technologies required to bridge the gap for adopting data-driven and AI applications in the mining industry.

The proposed formulation and methodology developed offer the following improvements over previous research in the context of open-pit planning:

- Integrate equipment activities uncertainties in the optimization process.
- Capability of considering many mining production scenarios during the RL training stage, which allowed the learning of a robust short-term planning policy.
- Model-free approach, which is entirely data-driven but guided by engineering constraints, that allows taking full advantage of mining databases and records.

The main industrial contribution of this research includes developing and testing a discrete event simulation model of open-pit production sytems integrated with popular AI software packages to facilitate the testing of innovative applications. Moreover, the practical framework developed by this research contributes to transferring knowledge to end-users and practitioners in the field of open-pit short-term planning.

## 5.4 Recommendations for Future Research

In this thesis, two main contributions were made to the current state-of-the-art literature in open-pit short-term planning by considering the adoption of AI-based, DRL, methods for dynamic truck dispatching and shovel allocation planning. This section offers suggestions for future research on each application individually, which other researchers can use to enhance this work.

### 5.4.1 Dynamic Truck Dispatching

Considering the system's assumptions described in Chapter 3, some recommendations for future research are below.

- Explore limitations in the generalization of the system and period of time for the model's validity. The performance of the dynamic truck dispatching model was verified for one month of production, which is significantly larger than the periods reported in other DRL applications in the literature, ranging from one shift to a maximum of one week. This is a key limitation as the system may require retraining past this verification time period. Further work should focus on enhancing the training process to improve the system's validity for longer periods of time and minimize retraining needs.
- Incorporate economic and other variables in the optimization process. Currently, the system is only tasked with achieving mining extraction rates, ore processing rates, and ore quality targets. However, it would be interesting to incorporate other variables such as operating costs, fuel consumption or greenhouse gas emissions to task the agent to learn a policy to minimize these as well.
- Incorporate relocation or another type of reactive decision in the face of equipment breakdowns. The current framework does not explicitly handle or adapt to the breakdown of trucks in the system. Currently, the system would observe this via its state description; however, it has no action type to react to such situations.

- Incorporate geological and mineral grade uncertainty in the optimization process. The proposed system assumes a deterministic ore grade at the bench faces; however, uncertainty in the quality grade of the ore being mined significantly impacts the performance of mining systems.

### 5.4.2 Shovel Allocation Planning

Considering the system's assumptions as described in Chapter 4, below are some recommendations for future research.

- Include truck fleet size decisions. Currently, the shovel allocation method proposed here estimates the truck fleet size using a deterministic calculation based on the location of the mining face the shovel is assigned to and its dumping destination. The system should decide along with the next shovel allocation and the truck fleet size required for this next route to optimize this decision under the different sources of uncertainties. This would allow for an integrated shovel and truck fleet allocation system that is much more useful for decision-makers.
- Optimize material destination decisions. Currently, the shovel allocation system assumes each mining face has a specified material destination facility, either a waste dump or an ore processing stream. However, optimizing the material destination decision from each mining face would also be beneficial. This could be accomplished by coupling a mathematical programming-based model that takes the shovel allocation decisions to find an optimal material allocation within the training loop of the DRL framework.
- Incorporate ancillary operations. Currently, no ancillary operations, such as drilling and blasting, are considered in the optimization framework. Considering this and developing

an integrated system for open-pit production scheduling, including drilling and blasting development activities, would be beneficial.

- Explore the possibility of developing a general shovel allocation model that can be used in different mines. This would require redesigning the agent's state and reward function representations and widening the training conditions to encourage generalization.
- Incorporate geological and mineral grade uncertainty in the optimization process. The proposed system assumes a deterministic ore grade at the bench faces; however, uncertainty in the quality grade of the ore being mined significantly impacts the performance of mining systems.

# BIBLIOGRAPHY

Ahmadi, M. R., & Bazzazi, A. A. (2019). Cutoff grades optimization in open pit mines using meta-heuristic algorithms. *Resources Policy*, *60*, 72–82. https://doi.org/10.1016/j.resourpol.2018.12.001

Ahmadi, M. R., & Shahabi, R. S. (2018). Cutoff grade optimization in open pit mines using genetic algorithm. *Resources Policy*, *55*, 184–191. https://doi.org/10.1016/j.resourpol.2017.11.016

Akbari, M., Nguyen, T., & Do, A. (2021). A systematic review of machine learning in logistics and supply chain management : current trends and future directions. *Benchmarking: An International Journal*, *28*(10), 2977–3005. https://doi.org/10.1108/BIJ-10-2020-0514

Alarie, S., & Gamache, M. (2002). Overview of solution strategies used in truck dispatching systems for open pit mines. *International Journal of Surface Mining, Reclamation and Environment*, *16*(1), 59–76. https://doi.org/10.1076/ijsm.16.1.59.3408

Albor Consuegra, F. R., & Dimitrakopoulos, R. (2009). Stochastic mine design optimisation based on simulated annealing: pit limits, production schedules, multiple orebody scenarios and sensitivity analysis. *Mining Technology*, *118*(2), 79–90. https://doi.org/10.1179/037178409x12541250836860

Alcaraz, J. J., Losilla, F., & Caballero-Arnaldos, L. (2022). Online model-based reinforcement learning for decision-making in long distance routes. *Transportation Research Part E: Logistics and Transportation Review*, *164*(June), 102790. https://doi.org/10.1016/j.tre.2022.102790

Ali, D., & Frimpong, S. (2021). DeepHaul: a deep learning and reinforcement learning-based smart automation framework for dump trucks. *Progress in Artificial Intelligence*, *10*(2), 157–180. https://doi.org/10.1007/s13748-021-00233-7

Alipour, A., Khodaiari, A. A., Jafari, A., & Tavakkoli-Moghaddam, R. (2018). Uncertain production scheduling optimization in open-pit mines and its ellipsoidal robust counterpart. *International Journal of Management Science and Engineering Management*, 1–9. https://doi.org/10.1080/17509653.2017.1410454

Alipour, A., Khodaiari, A. A., Jafari, A., & Tavakkoli-Moghaddam, R. (2019). Production scheduling of open-pit mines using genetic algorithm: a case study. *International Journal of Management Science and Engineering Management*, *15*(3), 176–183. https://doi.org/10.1080/17509653.2019.1683090

Alzubaidi, L., Zhang, J., Humaidi, A. J., Dujaili, A. Al, Duan, Y., Shamma, O. Al, Santamaría, J., Fadhel, M. A., Amidie, M. Al, & Farhan, L. (2021). Review of deep learning : concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(53). https://doi.org/10.1186/s40537-021-00444-8

Askari-Nasab, H., & Awuah-Offei, K. (2009). Open pit optimisation using discounted economic block values. *Mining Technology*, *118*(1), 1–12. https://doi.org/10.1179/037178409x12450752943243

Askari-Nasab, H., Frimpong, S., & Szymanski, J. (2007). Modelling open pit dynamics using discrete simulation. *International Journal of Mining, Reclamation and Environment*, *21*(1), 35–49. https://doi.org/10.1080/17480930600720206

Ataei, M., & Osanloo, M. (2004). Using a Combination of Genetic Algorithm and the Grid Search Method to Determine Optimum Cutoff Grades of Multiple Metal Deposits. *International Journal of Surface Mining, Reclamation and Environment*, *18*(1), 60–78. https://doi.org/10.1076/ijsm.18.1.60.23543

Avalos, S., & Ortiz, J. M. (2023). Multivariate Geostatistical Simulation and Deep Q-Learning to Optimize Mining Decisions. *Mathematical Geosciences*. https://doi.org/10.1007/s11004-023-10049-x

Awuah-Offei, K., Temeng, V. A., & Al-Hassan, S. (2003). Predicting equipment requirements using SIMAN simulation - a case study. *Mining Technology*, *112*(3), 180–184. https://doi.org/10.1179/037178403225003609

Azhari, F., Sennersten, C. C., Lindley, C. A., & Sellers, E. (2023). Deep learning implementations in mining applications: a compact critical review. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-023-10500-9

Azimi, Y., Osanloo, M., & Esfahanipour, A. (2011). Optimisation of Mining Policy Under Different Economical Conditions Using a Combination of Non-Linear Programming and Genetic Algorithm. In E. Y. Baffi (Ed.), *35th Application of Computers and Operations Research in the Minerals Industry Symposium APCOM 2011* (pp. 501–516).

Bahrpeyma, F., & Reichelt, D. (2022). A review of the applications of multi-agent reinforcement learning in smart factories. *Frontiers in Robotics and AI*, *9*(December), 1–21. https://doi.org/10.3389/frobt.2022.1027340

Bastos, G. S., Souza, L. E., Ramos, F. T., & Ribeiro, H. C. (2011). A Single-dependent Agent Approach for Stochastic Time-Dependent Truck Dispatching in Open-Pit Mining. *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 1057–1062. https://doi.org/10.1109/ITSC.2011.6082902

Beretta, F., Rodrigues, A. L., Peroni, R. L., & Costa, J. F. C. L. (2019). Automated lithological classification using UAV and machine learning on an open cast mine. *Applied Earth Science*, *128*(3), 79–88. https://doi.org/10.1080/25726838.2019.1578031

Bertolini, M., Mezzogori, D., Neroni, M., & Zammori, F. (2021). Machine Learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications*, *175*, 114820. https://doi.org/10.1016/j.eswa.2021.114820

Blom, M., Pearce, A. R., & Stuckey, P. J. (2016). A Decomposition-Based Algorithm for the Scheduling of Open-Pit Networks Over Multiple Time Periods. *Management Science*, *62*(10), 3059–3084.

Blom, M., Pearce, A. R., & Stuckey, P. J. (2019). Short-term planning for open pit mines: a review. *International Journal of Mining, Reclamation and Environment*, *33*(5), 318–339. https://doi.org/10.1080/17480930.2018.1448248

Blom, M., Pearce, A. R., Stuckey, P. J., Blom, M., Pearce, A. R., Short-term, P. J. S., Blom, M., Pearce, A. R., & Stuckey, P. J. (2016). Short-term scheduling of an open-pit mine with multiple objectives multiple objectives. *Engineering Optimization*, *49*(5), 777–795. https://doi.org/10.1080/0305215X.2016.1218002

Bonates, E., & Lizotte, Y. (1988). A combined approach to solve truck dispatching problems. In C. Fystas & S. Singhal (Eds.), *Computer Applications in the mineral industry* (pp. 403–412).

Both, C., & Dimitrakopoulos, R. (2020). Joint stochastic short-term production scheduling and fleet management optimization for mining complexes. *Optimization and Engineering*, *21*(4), 1717–1743. https://doi.org/10.1007/s11081-020-09495-x

Chaowasakoo, P., Seppälä, H., Koivo, H., & Zhou, Q. (2017a). Digitalization of mine operations: Scenarios to benefit in real-time truck dispatching. *International Journal of Mining Science and Technology*, *27*(2), 229–236. https://doi.org/10.1016/j.ijmst.2017.01.007

Chaowasakoo, P., Seppälä, H., Koivo, H., & Zhou, Q. (2017b). Improving fleet management in mines : The benefit of heterogeneous match factor. *European Journal of Operational Research*, *261*(3), 1052–1065. https://doi.org/10.1016/j.ejor.2017.02.039

Choi, Y., Nguyen, H., Bui, X.-N., Nguyen-Thoi, T., & Park, S. (2020). Estimating Ore Production in Open-pit Mines Using Various Machine Learning Algorithms Based on a Truck-Haulage System and Support of Internet of Things. *Natural Resources Research*, *30*(2), 1141–1173. https://doi.org/10.1007/s11053-020-09766-5

Curry, J. A., Ismay, M. J. L., & Jameson, G. J. (2014). Mine operating costs and the potential impacts of energy and grinding. In *Minerals Engineering* (Vol. 56, pp. 70–80). Elsevier Ltd. https://doi.org/10.1016/j.mineng.2013.10.020

Danish, A. A. K., Khan, A., Muhammad, K., Ahmad, W., & Salman, S. (2021). A simulated annealing based approach for open pit mine production scheduling with stockpiling option. *Resources Policy*, *71*, 102016. https://doi.org/10.1016/j.resourpol.2021.102016

de Carvalho, J. P., & Dimitrakopoulos, R. (2021). Integrating production planning with truck-dispatching decisions through reinforcement learning while managing uncertainty. *Minerals*, *11*(6). https://doi.org/10.3390/min11060587

Dendle, N., Isokangas, E., & Corry, P. (2022). Efficient simulation for an open-pit mine. *Simulation Modelling Practice and Theory*, *117*(August 2021), 102473. https://doi.org/10.1016/j.simpat.2021.102473

Dirkx, R., & Dimitrakopoulos, R. (2018). Optimizing Infill Drilling Decisions Using Multi-Armed Bandits: Application in a Long-Term, Multi-Element Stockpile. *Math Geosci*, *50*(1), 35–52. https://doi.org/10.1007/s11004-017-9695-9

Dulac-Arnold, G., Mankowitz, D., & Hester, T. (2019). Challenges of Real-World Reinforcement Learning. In N. Lawrence & M. Reid (Eds.), *Proceedings of the 36th International Conference on Machine Learning*. Proceedings of Machine Learning Research.

Eivazy, H., & Askari-Nasab, H. (2012). A mixed integer linear programming model for short-term open pit mine production scheduling. *Mining Technology*, *121*(2), 97–108. https://doi.org/10.1179/1743286312Y.0000000006

Ferland, J., Amaya, J., & Suzy Djuimo, M. (2007). Application of a Particle Swarm Algorithm to the Capacitated Open Pit Mining Problem. In S. C. Mukhopadhyay & G. S. Gupta (Eds.), *Autonomous Robot and Agents. Studies in Computational Intelligence.* (Vol. 76). Springer. https://doi.org/https://doi.org/10.1007/978-3-540-73424-6_15

Fortunato, M., Gheshlaghi, A., Piot, B., Menick, J., Hessel, M., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pierquin, O., Blundell, C., & Legg, S. (2018). Noisy Networks for Exploration. *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*.

Franco-Sepulveda, G., Jaramillo, G., & Del Rio, J. C. (2018). Use of Genetic Algorithms for Optimization of Open-Pit Mining Operations with Geological and Market Uncertainty. In E. Widzyk-Capehart, A. Hekmat, & R. Singhal (Eds.), *27th International Symposium on Mine Planning and Equipment Selection - MPES 2018* (pp. 107–119). Springet.

Gilani, S.-O., & Sattarvand, J. (2016). Integrating geological uncertainty in long-term open pit mine production planning by ant colony optimization. *Computers & Geosciences*, *87*, 31–40. https://doi.org/10.1016/j.cageo.2015.11.008

Gilani, S.-O., Sattarvand, J., Hajihassani, M., & Abdullah, S. S. (2020). A stochastic particle swarm based model for long term production planning of open pit mines considering the geological uncertainty. *Resources Policy*, *68*, 101738. https://doi.org/10.1016/j.resourpol.2020.101738

Goodfellow, R., & Dimitrakopoulos, R. (2017). Simultaneous Stochastic Optimization of Mining Complexes and Mineral Value Chains. *Math Geosci*, *49*(3), 341–360. https://doi.org/10.1007/s11004-017-9680-3

Goris Cervantes, E., Upadhyay, S., & Askari-Nasab, H. (2019). Improvements to production planning in oil sands mining through analysis and simulation of truck cycle times. *CIM Journal*, *10*(1), 39–52.

Guo, H., Nguyen, H., Vu, D.-A., & Bui, X.-N. (2019). Forecasting mining capital cost for open-pit mining projects based on artificial neural network approach. *Resources Policy*, 101474. https://doi.org/10.1016/j.resourpol.2019.101474

Gu, Q., Li, X., Chen, L., & Lu, C. (2020). Layout optimization of crushing station in open-pit mine based on two-stage fusion particle swarm algorithm. *Engineering Optimization*, 1–24. https://doi.org/10.1080/0305215x.2020.1817430

Gurgur, C. Z. (2011). Optimisation of a real-time multi-period truck dispatching system in mining operations. *International Journal of Applied Decision Sciences*, *4*(1), 57–79. https://doi.org/10.1504/IJADS.2011.038091

Hazrathosseini, A., & Moradi Afrapoli, A. (2023). The advent of digital twins in surface mining: Its time has finally arrived. *Resources Policy*, *80*(August 2022), 103155. https://doi.org/10.1016/j.resourpol.2022.103155

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2018). Rainbow: Combining Improvements in Deep Reinforcement Learning. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence2*.

Hildebrandt, F. D., Thomas, B. W., & Ulmer, M. W. (2023). Opportunities for reinforcement learning in stochastic dynamic vehicle routing. *Computers and Operations Research*, *150*(October 2022), 106071. https://doi.org/10.1016/j.cor.2022.106071

Huo, D., Sari, Y. A., Kealey, R., & Zhang, Q. (2023). Reinforcement Learning-Based Fleet Dispatching for Greenhouse Gas Emission Reduction in Open-Pit Mining Operations. *Resources, Conservation and Recycling*, *188*(March 2022), 106664. https://doi.org/10.1016/j.resconrec.2022.106664

Hustrulid, W., Kuchta, M., & Martin, R. (2013). *Open Pit Mine Planning & Design* (3rd Editio). CRC Press.

Icarte, G., Berrios, P., Castillo, R., & Herzog, O. (2020). A Multiagent System for Truck Dispatching in Open-Pit Mines. In M. Freitag, H. D. Haasis, H. Kotzab, & J. Pannek

(Eds.), *International Conference on Dynamics in Logistics* (pp. 363–373). Springer. https://doi.org/https://doi.org/10.1007/978-3-030-44783-0_35

Icarte, G., J., D. P., & Herzog, O. (2021). A Dynamic Scheduling Multiagent System for Truck Dispatching in Open-Pit Mines. In A. P. Rocha, L. Steels, & J. van den Herik (Eds.), *International Conference on Agents and Artificial Intelligence* (pp. 132–148). Springer. https://doi.org/https://doi.org/10.1007/978-3-030-71158-0_6

Jiao, Y., Tang, X., Qin, Z. (Tony), Li, S., Zhang, F., Zhu, H., & Ye, J. (2021). Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *130*(July), 103289. https://doi.org/10.1016/j.trc.2021.103289

Kavvadias, S., Drosatos, G., & Kaldoudi, E. (2020). Supporting topic modeling and trends analysis in biomedical literature. *Journal of Biomedical Informatics*, *110*(September), 103574. https://doi.org/10.1016/j.jbi.2020.103574

Khan, A., & Niemann-Delius, C. (2015). Application of Particle Swarm Optimization to the Open Pit Mine Scheduling Problem. In C. Niemann-Delius (Ed.), *12th International Synposium Continuous Surface Mining*. Springer. https://doi.org/https://doi.org/10.1007/978-3-319-12301-1_19

Kozan, E., & Liu, S. Q. (2015). A new open-pit multi-stage mine production timetabling model for drilling , blasting and excavating operations. *Mining Technology*, *125*(1), 47–53. https://doi.org/10.1179/1743286315Y.0000000031

Kumar, A., & Dimitrakopoulos, R. (2019). Application of simultaneous stochastic optimization with geometallurgical decisions at a copper–gold mining complex. *Mining Technology*, *128*(2), 88–105. https://doi.org/10.1080/25726668.2019.1575053

Kumar, A., & Dimitrakopoulos, R. (2021). Production scheduling in industrial mining complexes with incoming new information using tree search and deep reinforcement learning. *Applied Soft Computing*, *110*, 15. https://doi.org/10.1016/j.asoc.2021.107644

Kumar, A., Dimitrakopoulos, R., & Maulen, M. (2020). Adaptive self-learning mechanisms for updating short-term production decisions in an industrial mining complex. *Journal* *of Intelligent Manufacturing*, *31*(7), 1795–1811. https://doi.org/10.1007/s10845-020-01562-5

Kumral, M., & Dowd, P. A. (2005). A simulated annealing approach to mine production scheduling. *Journal of the Operational Research Society*, *56*(8), 922–930. https://doi.org/10.1057/palgrave.jors.2601902

Ladosz, P., Weng, L., Kim, M., & Oh, H. (2022). Exploration in deep reinforcement learning: A survey. *Information Fusion*, *85*(March), 1–22. https://doi.org/10.1016/j.inffus.2022.03.003

Lamghari, A., & Dimitrakopoulos, R. (2012). A diversified Tabu search approach for the open-pit mine production scheduling problem with metal uncertainty. *European Journal of Operational Research*, *222*(3), 642–652. https://doi.org/10.1016/j.ejor.2012.05.029

Lamghari, A., & Dimitrakopoulos, R. (2016). Progressive hedging applied as a metaheuristic to schedule production in open-pit mines accounting for reserve uncertainty. *2European Journal of Operational Research*, *253*(3), 843–855.

Lamghari, A., & Dimitrakopoulos, R. (2020). Hyper-heuristic approaches for strategic mine planning under uncertainty. *Computers & Operations Research*, *115*, 104590. https://doi.org/10.1016/j.cor.2018.11.010

Lamghari, A., Dimitrakopoulos, R., & Ferland, J. A. (2014). A variable neighbourhood descent algorithm for the open-pit mine production scheduling problem with metal uncertainty. *Journal of the Operational Research Society*, *65*(9), 1305–1314. https://doi.org/10.1057/jors.2013.81

Lamghari, A., Dimitrakopoulos, R., & Ferland, J. A. (2015). A hybrid method based on linear programming and variable neighborhood descent for scheduling production in open-pit mines. *Journal of Global Optimization*, *63*(3), 555–582. https://doi.org/10.1007/s10898-014-0185-z

Lawal, A. I., & Kwon, S. (2021). Application of artificial intelligence to rock mechanics: An overview. *Journal of Rock Mechanics and Geotechnical Engineering*, *13*(1), 248–266. https://doi.org/10.1016/j.jrmge.2020.05.010

Leite, A., & Dimitrakopoulos, R. (2007). A stochastic optimization model for open pit mine planning: Application and risk analysis at a copper deposit. *Transactions of Mining and Metallurgy: Mining Technology*, *116*(3).

Levinson, Z., & Dimitrakopoulos, R. (2020). Simultaneous stochastic optimisation of an open-pit gold mining complex with waste management. *International Journal of Mining, Reclamation and Environment*, *34*(6), 415–429. https://doi.org/10.1080/17480930.2019.1621441

Levinson, Z., Dimitrakopoulos, R., & Keutchayan, J. (2023). Simultaneous stochastic optimization of an open-pit mining complex with preconcentration using reinforcement learning. *Applied Soft Computing*, *138*, 110180. https://doi.org/10.1016/j.asoc.2023.110180

L'Heureux, G., Gamache, M., & Soumis, F. (2013). Mixed integer programming model for short term planning in open-pit mines. *Mining Technology*, *122*(2), 101–109. https://doi.org/10.1179/1743286313Y.0000000037

Li, C., Bai, L., Yao, L., Waller, S. T., & Liu, W. (2023). A bibliometric analysis and review on reinforcement learning for transportation applications. *Transportmetrica B : Transport Dynamics*, *11*(1), 1095–1135. https://doi.org/10.1080/21680566.2023.2179461

Li, S., Sari, Y. A., & Kumral, M. (2020). Optimization of Mining–Mineral Processing Integration Using Unsupervised Machine Learning Algorithms. *Natural Resources Research*, *29*(5), 3035–3046. https://doi.org/10.1007/s11053-020-09628-0

Lotfian, R., Gholamnejad, J., & Mirzaeian Lardkeyvan, Y. (2020). Effective solution of the long-term open pit production planning problem using block clustering. *Engineering Optimization*, *53*(7), 1119–1134. https://doi.org/10.1080/0305215x.2020.1771703

Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers and Operations Research*, *134*(March), 105400. https://doi.org/10.1016/j.cor.2021.105400

McCoy, J. T., & Auret, L. (2019). Machine learning applications in minerals processing: A review. *Minerals Engineering*, *132*, 95–109. https://doi.org/10.1016/j.mineng.2018.12.004

Mena, R., Zio, E., Kristjanpoller, F., & Arata, A. (2013). Availability-based simulation and optimization modeling framework for open-pit mine truck allocation under dynamic constraints. *International Journal of Mining Science and Technology*, *23*(1), 113–119. https://doi.org/10.1016/j.ijmst.2013.01.017

Mirzaei-nasirabad, H., Mohtasham, M., Askari-nasab, H., & Alizadeh, B. (2023). An optimization model for the real-time truck dispatching problem in open-pit mining operations. *Optimization and Engineering*. https://doi.org/10.1007/s11081-022-09780-x

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Mohammadi, S., Kakaie, R., Ataei, M., & Pourzamani, E. (2017). Determination of the optimum cut-off grades and production scheduling in multi-product open pit mines using imperialist competitive algorithm (ICA). *Resources Policy*, *51*, 39–48. https://doi.org/10.1016/j.resourpol.2016.11.005

Mohtasham, M., Mirzaei-nasirabad, H., & Askari-nasab, H. (2021). A multi-objective model for fleet allocation schedule in open-pit mines considering the impact of prioritising objectives on transportation system performance. *International Journal of Mining, Reclamation and Environment*, *35*(10), 709–727. https://doi.org/10.1080/17480930.2021.1949861

Mohtasham, M., Mirzaei-Nasirabad, H., Askari-Nasab, H., & Alizadeh, B. (2022). Multi-stage optimization framework for the real-time truck decision problem in open-pit mines: a case study on Sungun copper mine. *International Journal of Mining, Reclamation and Environment*, *36*(7), 461–491. https://doi.org/10.1080/17480930.2022.2067709

Montiel, L., & Dimitrakopoulos, R. (2013). Stochastic Mine Production Scheduling with Multiple Processes: Application at Escondida Norte, Chile. *Journal of Mining Science*, *49*(4), 583–597.

Montiel, L., & Dimitrakopoulos, R. (2015). Optimizing mining complexes with multiple processing and transportation alternatives: An uncertainty-based approach. *European Journal of Operational Research*, *247*(1), 166–178. https://doi.org/10.1016/j.ejor.2015.05.002

Moosavi, E. (2020). *Modeling Long-Term Production Scheduling Problem and Its Solution Using a Bat Meta-heuristic Method*. 111–119. https://doi.org/10.1007/978-3-030-33954-8_13

Moosavi, E., Gholamnejad, J., Ataee-pour, M., & Khorram, E. (2014). A Hybrid Augmented Lagrangian Multiplier Method for the Open Pit Mines Long-Term Production Scheduling Problem Optimization. *Journal of Mining Science*, *50*(6), 1047–1060.

Moradi Afrapoli, A., & Askari-Nasab, H. (2019). Mining fleet management systems: a review of models and algorithms. *International Journal of Mining, Reclamation and Environment*, *33*(1), 42–60. https://doi.org/10.1080/17480930.2017.1336607

Moradi Afrapoli, A., & Askari-Nasab, H. (2020). A stochastic integrated simulation and mixed integer linear programming optimisation framework for truck dispatching problem in surface mines. *International Journal of Mining and Mineral Engineering*, *11*(4), 257–284. https://doi.org/https://doi.org/10.1504/IJMME.2020.111929

Moradi Afrapoli, A., Tabesh, M., & Askari-Nasab, H. (2018). A stochastic hybrid simulation-optimization approach towards haul fleet sizing in surface mines. *Mining Technology*, *128*(1), 9–20. https://doi.org/10.1080/25726668.2018.1473314

Moradi Afrapoli, A., Tabesh, M., & Askari-Nasab, H. (2019). A multiple objective transportation problem approach to dynamic truck dispatching in surface mines. *European Journal of Operational Research*, *276*(1), 331–342. https://doi.org/10.1016/j.ejor.2019.01.008

Mousavi, A., Kozan, E., & Liu, S. Q. (2016). Comparative analysis of three metaheuristics for short-term open pit block sequencing. *Journal of Heuristics*, *22*(3), 301–329. https://doi.org/10.1007/s10732-016-9311-z

Naeem, M., Tahir, S., Rizvi, H., & Coronato, A. (2020). A Gentle Introduction to Reinforcement Learning and Its Application in Different Fields. *IEEE Access*, *8*, 209320–209344. https://doi.org/10.1109/ACCESS.2020.3038605

Nazari, M., Oroojlooy, A., Snyder, L., & Takac, M. (2018). Reinforcement Learning for Solving the Vehicle Routing Problem. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*. Advances in Neural Information Processing Systems.

Newman, A., Rubio, E., Caro, R., Weintraub, A., & Eurek, K. (2010). A review of operations research in mine planning. *Interfaces*, *40*(3), 222–245.

Nourali, H., & Osanloo, M. (2018). A regression-tree-based model for mining capital cost estimation. *International Journal of Mining, Reclamation and Environment*, *34*(2), 88–100. https://doi.org/10.1080/17480930.2018.1510300

Nourali, H., & Osanloo, M. (2019). Mining capital cost estimation using Support Vector Regression (SVR). *Resources Policy*, *62*, 527–540. https://doi.org/10.1016/j.resourpol.2018.10.008

Obando-Ceron, J. S., & Castro, P. S. (2021). Revisiting Rainbow: Promoting more Insightful and Inclusive Deep Reinforcement Learning Research. *Proceedings of the 38th International Conference on Machine Learning, PMLR*, 1373–1383. http://arxiv.org/abs/2011.14826

Olson, J. P., Vohnout, S. I., & White, J. (1993). On improving truck/shovel productivity in open pit mines. *CIM Bulletin*, *86*(973), 43–49.

Osaba, E., Carballedo, R., Diaz, F., Onieva, E., Masegosa, A. D., & Perallos, A. (2018). Good practice proposal for the implementation, presentation, and comparison of metaheuristics for solving routing problems. *Neurocomputing*, *271*, 2–8. https://doi.org/10.1016/j.neucom.2016.11.098

Ozdemir, B., & Kumral, M. (2018). Appraising production targets through agent-based Petri net simulation of material handling systems in open pit mines. *Simulation Modelling Practice and Theory*, *87*, 138–154. https://doi.org/10.1016/j.simpat.2018.06.008

Ozdemir, B., & Kumral, M. (2019). Simulation-based optimization of truck-shovel material handling systems in multi-pit surface mines. *Simulation Modelling Practice and Theory*, *95*, 36–48. https://doi.org/10.1016/j.simpat.2019.04.006

Paduraru, C., & Dimitrakopoulos, R. (2017). Adaptive policies for short-term material flow optimization in a mining complex. *Mining Technology*, *127*(1), 56–63. https://doi.org/10.1080/14749009.2017.1341142

Paduraru, C., & Dimitrakopoulos, R. (2019). Responding to new information in a mining complex: fast mechanisms using machine learning. *Mining Technology*, *128*(3), 129–142. https://doi.org/10.1080/25726668.2019.1577596

Paithankar, A., & Chatterjee, S. (2019). Open pit mine production schedule optimization using a hybrid of maximum-flow and genetic algorithms. *Applied Soft Computing*, *81*, 105507. https://doi.org/10.1016/j.asoc.2019.105507

Panzer, M., & Bender, B. (2021). Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research*. https://doi.org/10.1080/00207543.2021.1973138

Panzer, M., Bender, B., & Gronau, N. (2021). Deep Reinforcement Learning in Production Planning and Control: A Systematic Literature Review. In D. Herberger & M. Hubner (Eds.), *Proceedings of the 2nd Conference on Production Systems and Logistics: CPSL 2021* (pp. 535–545). Leibniz Universitat Hannover.

Pendharkar, P., & Rodger, A. (2000). Nonlinear programming and genetic search application for production scheduling in coal mines. *Annals of Operations Research*, *95*, 251–267.

Pu, Y., Apel, D. B., Szmigiel, A., & Chen, J. (2019). Image Recognition of Coal and Coal Gangue Using a Convolutional Neural Network and Transfer Learning. *Energies*, *12*(9), 1735. https://doi.org/10.3390/en12091735

Raj, M. G., Vardhan, H., & Rao, Y. V. (2009). Production optimisation using simulation models in mines: A critical review. *International Journal of Operational Research*, *6*(3), 330–359. https://doi.org/10.1504/IJOR.2009.026937

Rezazadeh Azar, E., & McCabe, B. Y. (2013). A Visual Sensing Approach to Estimate Material Hauling Cycles in Heavy Construction and Surface Mining Jobsites. *30th International Symposium on Automation and Robotics in Construction*, 774–782.

Riff, M., Moossen, M., & Bonnaire, X. (2005). Miner Ants Colony: A New Approach to Solve a Mine Planning Problem. In F. A., J. Kok, J. Pena, A. Siebes, & A. Feelders (Eds.), *International Symposium on Intelligent Data Analysis 2005: Advances in Intelligent Data*

*Analysis VI* (Vol. 3646). Springer. https://doi.org/https://doi.org/10.1007/11552253_30

Ristovski, K., Gupta, C., Harada, K., & Tang, H. K. (2017). Dispatch with confidence: Integration of machine learning, optimization and simulation for open pit mines. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *Part F1296*, 1981–1989. https://doi.org/10.1145/3097983.3098178

Saliba, Z., & Dimitrakopoulos, R. (2019). Simultaneous stochastic optimization of an open pit gold mining complex with supply and market uncertainty. *Mining Technology*, *128*(4), 216–229. https://doi.org/10.1080/25726668.2019.1626169

Saliba, Z., & Dimitrakopoulos, R. (2020). An application of simultaneous stochastic optimisation of an open-pit mining complex with tailings management. *International Journal of Mining, Reclamation and Environment*, *34*(8), 592–607. https://doi.org/10.1080/17480930.2019.1688954

Samantha, B., Vaibhav Raj, K., Battacherjee, A., & Chakravarty, D. (2013). Uncertainty-based mine production scheduling for ore grade control using conditional simulation and multiobjective genetic algorithms. *Mining Engineering*, *65*(4), 66–76.

Senécal, R., & Dimitrakopoulos, R. (2019). 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*, *34*(7), 459–475. https://doi.org/10.1080/17480930.2019.1595902

Shishvan, M. S., & Benndorf, J. (2018). Simulation-based optimization approach for material dispatching in continuous mining systems. *European Journal of Operational Research*, *275*(3), 1108–1125. https://doi.org/10.1016/j.ejor.2018.12.015

Shishvan, M. S., & Benndorf, J. (2019). Simulation-based optimization approach for material dispatching in continuous mining systems. *European Journal of Operational Research*, *275*(3), 1108–1125. https://doi.org/10.1016/j.ejor.2018.12.015

Shishvan, M. S., & Sattarvand, J. (2015). Long term production planning of open pit mines by ant colony optimization. *European Journal of Operational Research*, *240*(3), 825–836. https://doi.org/10.1016/j.ejor.2014.07.040

Siami-Irdemoosa, E., & Dindarloo, S. R. (2015). Prediction of fuel consumption of mining dump trucks: A neural networks approach. *Applied Energy*, *151*, 77–84. https://doi.org/10.1016/j.apenergy.2015.04.064

Soofastaei, A., Aminossadati, S. M., Arefi, M. M., & Kizil, M. S. (2016). Development of a multi-layer perceptron artificial neural network model to determine haul trucks energy consumption. *International Journal of Mining Science and Technology*, *26*(2), 285–293. https://doi.org/10.1016/j.ijmst.2015.12.015

Soofastaei, A., Aminossadati, S. M., Kizil, M. S., & Knights, P. (2016). A discrete-event model to simulate the effect of truck bunching due to payload variance on cycle time, hauled mine materials and fuel consumption. *International Journal of Mining Science and Technology*, *26*(5), 745–752. https://doi.org/10.1016/j.ijmst.2016.05.047

Soumis, F., Ethier, J., & Elbrond, J. (1989). Truck dispatching in an open pit mine. *International Journal of Surface Mining, Reclamation and the Environment*, *3*(2), 115–119.

Souza, M. J. F., Coelho, I. M., Ribas, S., Santos, H. G., & Merschmann, L. H. C. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, *207*(2), 1041–1051. https://doi.org/10.1016/j.ejor.2010.05.031

Sutton, R., & Barto, A. (2018). *Reinforcement Learning: An Introduction* (2nd editio). The MIT Press.

Sutton, R. S. (1988). Learning to Predict by the Methods of Temporal Differences. *Machine Learning*, *3*(1), 9–44. https://doi.org/10.1023/A:1022633531479

Tabesh, M., & Askari-Nasab, H. (2013). Automatic Creation of Mining Polygons Using Hierarchical Clustering Techniques. *Journal of Mining Science*, *49*(3), 426–440.

Tabesh, M., & Askari-Nasab, H. (2019). Clustering mining blocks in presence of geological uncertainty. *Mining Technology*, *128*(3), 162–176. https://doi.org/10.1080/25726668.2019.1596425

Ta, C. H., Kresta, J. V., Forbes, J. F., & Marquez, H. J. (2005). A stochastic optimization approach to mine truck allocation. *International Journal of Surface Mining, Reclamation and Environment*, *19*(3), 162–175. https://doi.org/10.1080/13895260500128914

Tan, Y., Chimbat, U., Miwa, K., & Takakuwa, S. (2012). Operations Modeling and Analysis of Open pit Copper Mining Using GPS Tracking Data. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, & A. Uhrmacher (Eds.), *2012 Winter Simulation Conference*. INFORMS.

Temeng, V. A., Otuonye, F. O., Jr, J. O. F., Temeng, V. A., Otuonye, F. O., Jr, J. O. F., Temeng, V. A., Otuonye, F. O., & Frendewey, J. O. (1997). Real-time truck dispatching using a transportation algorithm Real-time truck dispatching using a transportation algorithm. *International Journal of Surface Mining, Reclamation and Environment*, *11*, 203–207. https://doi.org/10.1080/09208119708944093

Temeng, V., Otuonye, F., & Frendewey, J. (1998). A Nonpreemptive Goal Programming Approach to Truck Dispatching in Open Pit Mines. *Mineral Resources Engineering*, *07*(2).

Tolouei, K., Moosavi, E., Bangian Tabrizi, A. H., & Afzal, P. (2020). Application of an improved Lagrangian relaxation approach in the constrained long-term production scheduling problem under grade uncertainty. *Engineering Optimization*, *53*(5), 735–753. https://doi.org/10.1080/0305215x.2020.1746295

Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *British Journal of Management*, *14*, 207–222.

Tsitsiklis, J. N., & Van Roy, B. (1997). An Analysis of Temporal-Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control*, *42*(5), 674–690.

Ugurlu, O. F., Ozdemir, B., & Kumral, M. (2020). Drilling Performance Management Through Reliability-Based Optimization. *Journal of Failure Analysis and Prevention*, *20*, 23–29. https://doi.org/10.1007/978-3-030-48021-9_3

Upadhyay, S. P., & Askari-Nasab, H. (2017). Dynamic shovel allocation approach to short-term production planning in open-pit mines. *International Journal of Mining, Reclamation and Environment*, *33*(1), 1–20. https://doi.org/10.1080/17480930.2017.1315524

Upadhyay, S. P., & Askari-Nasab, H. (2018a). Simulation and optimization approach for uncertainty-based short-term planning in open pit mines. *International Journal of*

*Mining Science and Technology*, *28*(2), 153–166. https://doi.org/10.1016/j.ijmst.2017.12.003

Upadhyay, S. P., & Askari-Nasab, H. (2018b). Simulation and optimization approach for uncertainty-based short-term planning in open pit mines. *International Journal of Mining Science and Technology*, *28*(2), 153–166. https://doi.org/10.1016/j.ijmst.2017.12.003

Upadhyay, S. P., Tabesh, M., Badiozamani, M., & Askari-Nasab, H. (2019). A Simulation Model for Estimation of Mine Haulage Fleet Productivity. In E. Topal (Ed.), *28th International Symposium on Mine Planning and Equipment Selection - MPES 2019.* (pp. 42–50). Springer. https://doi.org/https://doi.org/10.1007/978-3-030-33954-8_5

Upadhyay, S., Tabesh, M., Badiozamani, M., Moradi Afrapoli, A., & Askari-Nasab, H. (2021). A simulation-based algorithm for solving surface mines' equipment selection and sizing problem under uncertainty. *CIM Journal*, *12*(1), 36–46.

van Hasselt, H., Guez, A., & Silver, D. (2016). Deep Reinforcement Learning with Double Q-Learning. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-16)*, 2094–20100. https://doi.org/https://doi.org/10.1609/aaai.v30i1.10295

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lactot, M., & de Freites, N. (2016). Dueling Network Architectures for Deep Reinforcement Learning. *Proceedings of the 33rd International Conference on Machine Learning2*.

White, J. W., & Olson, J. P. (1986). Computer-based dispatching in mines with concurrent operating objectives. *Mining Engineering*, *38*(11), 1045–1054.

Whittle, D. (2011). Open-Pit Planning and Design. In P. Darling (Ed.), *SME Mining Engineering Handbook* (3rd Editio, pp. 877–901). Society for Mining, Metallurgy and Exploration.

Williams, J., Singh, J., Kumral, M., & Ramirez Ruiseco, J. (2021). Exploring Deep Learning for Dig-Limit Optimization in Open-Pit Mines. *Natural Resources Research*, *30*(3), 2085–2101. https://doi.org/10.1007/s11053-021-09864-y

Wong, A., Bäck, T., Kononova, A. V., & Plaat, A. (2022). Deep multiagent reinforcement learning: challenges and directions. In *Artificial Intelligence Review* (Issue 0123456789). Springer Netherlands. https://doi.org/10.1007/s10462-022-10299-x

Xiao, Y., & Watson, M. (2017). Guidance on Conducting a Systematic Literature Review. *Journal of Planning Education and Research*, *39*(1), 93–112. https://doi.org/10.1177/0739456x17723971

Yaghini, A., Hall, R. A., & Apel, D. (2020). Modeling the influence of electric shovel operator performance on mine productivity. *CIM Journal*, *11*(1), 58–68. https://doi.org/10.1080/19236026.2020.1734405

Yao, Z., Huang, Q., Ji, Z., Li, X., & Bi, Q. (2021). Deep learning-based prediction of piled-up status and payload distribution of bulk material. *Automation in Construction*, *121*, 103424. https://doi.org/10.1016/j.autcon.2020.103424

Yeganejou, M., Badiozamani, M., Moradi-Afrapoli, A., & Askari-Nasab, H. (2021). Integration of simulation and dispatch modelling to predict fleet productivity: an open-pit mining case. *Mining Technology*, 1–13. https://doi.org/10.1080/25726668.2021.2001255

Zhang, C., Odonkor, P., Zheng, S., Khorasgani, H., Serita, S., & Gupta, C. (2020). Dynamic Dispatching for Large-Scale Heterogeneous Fleet via Multi-agent Deep Reinforcement Learning. *2020 IEEE International Conference on Big Data (Big Data)*, 1436–1441.

Zhang, H., Nguyen, H., Bui, X.-N., Nguyen-Thoi, T., Bui, T.-T., Nguyen, N., Vu, D.-A., Mahesh, V., & Moayedi, H. (2020). Developing a novel artificial intelligence model to estimate the capital cost of mining projects using deep neural network-based ant colony optimization algorithm. *Resources Policy*, *66*, 101604. https://doi.org/10.1016/j.resourpol.2020.101604

Zhang, J., He, T., Sra, S., & Jadbabaie, A. (2020). Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity. *Eighth International Conference on Learning Representations ICLR 2020*.