Interpretable Deep Reinforcement Learning for Dynamic Truck Dispatching in Open-Pit Mining

Francisco Rosales
Universidad ESAN
frosales@esan.edu.pe

Angelo Diaz
IMCA
angelo.diaz@imca.edu.pe

***Abstract*—Dynamic resource allocation under uncertainty remains a central challenge in operations research, particularly in capital-intensive industries such as open-pit mining, where inefficient dispatching decisions can lead to substantial productivity losses and increased operational costs. This paper presents a Deep Reinforcement Learning (DRL) framework for the dynamic truck assignment problem in open-pit mining, with a dual emphasis on performance optimization and policy interpretability. The problem is modeled as a Markov Decision Process and implemented within a discrete-event simulation environment that captures the stochastic behavior of truck–shovel interactions. A Deep Q-Learning approach with neural network function approximation is employed to learn adaptive dispatching policies directly from simulated experience. To promote transparency and industrial applicability, Explainable Artificial Intelligence techniques based on Shapley values are applied to interpret and validate the learned decision strategies. Experimental results demonstrate that the proposed framework substantially outperforms heuristic dispatching methods while providing interpretable insights that support trust and deployment in real mining operations.**

***Index Terms*—Deep Q-Learning, Explainable AI, Truck Dispatching, Markov Decision Process, Open-Pit Mining, Discrete-Event Simulation, Interpretable Reinforcement Learning**

I. Introduction

Efficient truck dispatching is a critical operational problem in open-pit mining, where resource allocation decisions must be made under uncertainty and tight operational constraints. Poor dispatching decisions can lead to congestion, underutilization of equipment, and significant productivity losses. This paper proposes a Deep Reinforcement Learning (DRL) framework for dynamic truck dispatching that simultaneously optimizes operational performance and enhances decision interpretability.

The truck–shovel assignment problem is formulated as a Markov Decision Process (MDP) and embedded within a discrete-event simulation environment that captures the stochastic and asynchronous nature of mining operations. A Deep Q-Learning (DQL) agent learns dispatching policies directly from simulated experience, without requiring an explicit model of system dynamics.

The application of reinforcement learning to mining systems builds upon a substantial body of prior research. Foundational work by Bellman [1] and Watkins and Dayan [2] established the theoretical basis for dynamic programming and Q-learning. More recent advances in Deep Q-Networks [3] have enabled reinforcement learning methods to scale to complex, high-dimensional environments. In mining applications, Noriega et al. [4] and Huo et al. [5] demonstrated the effectiveness of DRL for haulage optimization, while de Carvalho and Dimitrakopoulos [6] and Chiarot et al. [7] explored reinforcement learning under operational constraints.

Despite promising performance gains, most existing studies focus primarily on aggregate productivity metrics and provide limited insight into the rationale behind learned decisions. In industrial mining environments—characterized by high capital costs, safety considerations, and strong operational accountability—this lack of interpretability presents a significant barrier to adoption. Dispatching decisions must not only be effective but also explainable to engineers, operators, and managers.

This study explicitly addresses this gap by integrating Explainable Artificial Intelligence (XAI) techniques into the reinforcement learning pipeline. By leveraging SHAP (SHapley Additive exPlanations), we provide transparent, post-hoc explanations of dispatching decisions, highlighting how system-level and agent-level features influence policy behavior. This interpretability enables validation against domain knowledge, increases trust in the learned policy, and supports its use as a practical decision-support tool in real mining operations.

The remainder of the paper is organized as follows. Section II presents the operational framework and mining environment. Section III details the MDP formulation and Deep Q-Learning implementation. Section IV describes the discrete-event simulation environment. Section V presents experimental results and interpretability analysis. Section VI concludes the paper and outlines future research directions.

II. Operational Framework

This section formalizes the truck dispatching problem as a sequential decision-making task under uncertainty. The environment, decision variables, and reward structure are defined within a reinforcement learning framework to ensure alignment between learned policies and operational objectives.

A. Loading and Haulage Processes

The loading and haulage cycle consists of four stages: (i) loading at the shovel, (ii) hauling to the destination, (iii) unloading material, and (iv) returning for the next loading operation. The dispatching decision—selecting which shovel an available truck should be assigned to after unloading—constitutes the core optimization problem.

For modeling clarity, trucks and shovels are assumed to be homogeneous, with identical load capacities and normally distributed loading and unloading times. While simplified, this assumption allows controlled analysis of learning behavior and isolates the effects of dispatching logic from equipment heterogeneity.

B. Open-Pit Mining Environment

The environment is modeled as a two-dimensional spatial system with fixed shovel and dump locations. The simulation includes three shovels, four dumps, and ten trucks. Each shovel follows predefined material routing rules: ore is transported to crushers, waste to waste dumps, and low-grade ore to stockpiles. When multiple crushers are available, ore is routed to the least utilized facility to promote balanced downstream processing.

C. Truck Dispatching Variables

The environment state is represented using feature vectors that encode both global system conditions and agent-specific information.

State Representation: Global features include queue lengths at each shovel and dump, as well as the normalized remaining simulation time. Agent-level features capture each truck's current location (encoded as a one-hot vector), remaining travel time, and residual operation time. This representation provides sufficient information to capture congestion, spatial distribution, and temporal efficiency.

Action Space: At each decision epoch, the agent assigns an available truck to one of the eligible shovels that has not yet reached its extraction threshold. This formulation reflects operational constraints while allowing dynamic adaptation to evolving system conditions.

Reward Function: The reward function is designed to balance productivity maximization with congestion mitigation:

$$R = \begin{cases} -\text{Queue time}, & \mathcal{W}, \\ \text{Low-grade ore} - \text{Trucks in queues}, & \mathcal{S}, \\ \text{Pure ore} - \text{Queue time}, & \mathcal{C}, \end{cases} \tag{1}$$

where $\mathcal{W}$, $\mathcal{S}$, and $\mathcal{C}$ denote waste dumps, stockpiles, and crushers, respectively.

This structure reflects operational priorities commonly observed in mining practice. Ore deliveries to crushers are directly rewarded due to their high economic value, while queue times are penalized to discourage congestion. Low-grade ore deliveries are incentivized but adjusted by queue penalties to prevent excessive buildup at stockpiles. Waste handling is treated as necessary but non-value-adding, and is therefore penalized primarily through waiting time. While the reward terms are not explicitly weighted, their relative influence emerges naturally from operational frequencies and system dynamics.

Although alternative reward formulations are possible, empirical observations during training indicated that the learned policy is robust to moderate variations in reward scaling, consistently exhibiting congestion-aware and productivity-oriented behaviors. A systematic sensitivity analysis of reward design constitutes an important direction for future research.

III. Deep Reinforcement Learning

Truck dispatching is modeled as a stochastic sequential decision-making problem solved using Deep Q-Learning. The MDP is defined by the tuple $\langle S, A, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\gamma \in [0,1]$ is the discount factor.

A. Deep Q-Learning Implementation

The action-value function $q(s,a;w)$ is approximated using a neural network with parameters w, which are updated using temporal-difference learning:

$$w_{t+1} = w_t + \alpha \Big(r + \gamma \max_{a'} q(s',a';w_t) - q(s,a;w_t) \Big) \nabla_w q(s,a;w_t). \tag{2}$$

This model-free approach is particularly well suited to mining operations, where explicit transition models are difficult to obtain due to stochastic loading times, equipment interactions, and dynamic traffic conditions.

IV. Open-Pit Mine Simulation

The simulation environment provides a virtual testbed for evaluating reinforcement learning–based dispatching strategies. It integrates discrete-event modeling of truck–shovel interactions with stochastic ore type discovery in order to realistically capture the operational dynamics of open-pit mining systems.

A. Discrete-Event Simulation Framework

The mining operation is modeled as a discrete-event system in which each event represents a key transition in the truck loading and haulage process. Events are processed chronologically using a priority queue, enabling accurate representation of the stochastic and dynamic behavior of the system.

Each event is defined by a tuple (`time`, `truck`, `event_type`, `position`, `index`), which specifies the simulation time, the truck involved, the event type (ARRIVAL or DEPARTURE), the facility category (shovel or dump), and the corresponding facility index. This event-driven formulation ensures deterministic state transitions while maintaining computational efficiency and scalability.

B. Stochastic Modeling of Ore Type Discovery

Geological uncertainty is incorporated through probabilistic modeling of ore type discovery at each loading event. Each shovel operates according to predefined material composition distributions, as summarized in Table I.

TABLE I: Material type probabilities by shovel

Shovel	Ore	Waste	Low-grade
Shovel 1	0.50	0.30	0.20
Shovel 2	0.40	0.40	0.20
Shovel 3	0.35	0.45	0.20

For shovel i, the probability of extracting material type k is defined as $p_{i,k} = n_{i,k} / \sum_{k'} n_{i,k'}$, where $n_{i,k}$ denotes the expected number of extractions of material k according to the mine plan. This formulation preserves overall production targets while introducing realistic local variability in material composition.

C. Simulation Assumptions

To ensure learning stability and analytical tractability, the simulation relies on several simplifying assumptions, summarized in Table II.

TABLE II: Key simulation assumptions

Category	Assumption
Equipment	Homogeneous trucks and shovels with identical capacities and travel speeds
Spatial	Fixed two-dimensional layout with predefined deterministic haulage routes
Operations	No equipment failures or maintenance-related interruptions
Information	Perfect state observability at all decision epochs

These assumptions allow the analysis to focus on the core dispatching dynamics while isolating the effects of reinforcement learning. Extensions incorporating equipment heterogeneity, stochastic failures, and partial observability are identified as important directions for future work.

V. Evaluation and Results

The proposed framework was evaluated through simulation experiments assessing convergence behavior, policy performance, and interpretability.

A. Policy Performance Comparison

Table III compares the RL-based policy against benchmark dispatching strategies.

TABLE III: Average performance across different dispatching policies

Policy	Ave Ore	Ave Waste	Ave Stockpile
Random	117.197	63.632	90.339
Shortest Queue	93.938	61.000	101.698
RL-Based	**154.000**	**40.000**	**91.000**

The RL-based policy significantly outperforms both benchmarks, achieving higher ore production while reducing waste handling and maintaining balanced stockpile utilization. Although the experimental setup considers a fixed-scale scenario with homogeneous equipment, the results demonstrate the agent's ability to learn non-myopic dispatching behavior that accounts for system-wide interactions. Evaluating robustness under varying fleet sizes, shovel configurations, and demand patterns represents a natural extension of this work.

B. Interpretable Model Analysis Using SHAP

To interpret the learned dispatching policy, we employ SHAP (SHapley Additive exPlanations), an explainability method grounded in cooperative game theory. SHAP attributes a contribution value to each input feature by computing its marginal impact on the model's output across all possible feature coalitions. This provides a consistent and locally accurate explanation of individual decisions.

In the context of reinforcement learning, SHAP is applied post hoc to the trained Q-network to explain why a particular action was selected in a given state. For each dispatching decision, SHAP values quantify how features such as queue lengths, truck locations, and remaining travel times influence the estimated action values.

Beyond individual instances, we analyzed SHAP values across multiple decision points and simulation episodes, observing consistent patterns. Queue lengths at shovels and crushers systematically exert strong influence on action selection, indicating congestion-aware behavior. Spatial distribution of trucks contributes to anticipatory routing decisions, while remaining travel times affect prioritization under time pressure. These aggregated observations suggest that the learned policy consistently follows rational operational principles, rather than relying on isolated or incidental behaviors.

VI. Conclusions and Future Work

This paper demonstrates that Deep Reinforcement Learning, integrated with discrete-event simulation, can learn effective and interpretable dispatching policies for open-pit mining operations. The proposed framework outperforms heuristic baselines while providing transparent insights into decision-making through SHAP-based explanations.

While the experimental evaluation focuses on a fixed-scale, homogeneous setting, the results highlight the potential of DRL as a scalable and trustworthy decision-support tool. Future work will extend the framework to heterogeneous fleets, stochastic equipment failures, alternative reward formulations, and multi-agent learning architectures. Further systematic analysis of interpretability metrics over long horizons will also strengthen the integration of explainable reinforcement learning in industrial applications.

References

[1] R. Bellman, "A markovian decision process," *Indiana University Mathematics Journal*, vol. 6, no. 4, pp. 679–684, 1957.

[2] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[3] A. Zai and B. Brown, *Deep Reinforcement Learning in Action*. Manning Publications, 2020.

[4] R. Noriega, Y. Pourrahimian, and H. Askari-Nasab, "Deep reinforcement learning based real-time open-pit mining truck dispatching system," *Computers & Operations Research*, vol. 173, p. 106815, 2025.

[5] D. Huo, Y. A. Sari, and Q. Zhang, "Smart dispatching for low-carbon mining fleet: A deep reinforcement learning approach," *Journal of Cleaner Production*, vol. 435, p. 140459, 2024.

[6] J. P. de Carvalho and R. Dimitrakopoulos, "Integrating production planning with truck-dispatching decisions through reinforcement learning while managing uncertainty," *Minerals*, vol. 11, no. 6, p. 587, 2021.

[7] T. V. Chiarot Villegas, S. F. Segura Altamirano, D. M. Castro Cárdenas, A. M. Sifuentes Montes, L. I. Chaman Cabrera, A. S. Aliaga Zegarra, C. L. Oblitas Vera, and J. C. Albán Palacios, "Improving productivity in mining operations: a deep reinforcement learning model for effective material supply and equipment management," *Neural Computing and Applications*, vol. 36, no. 9, pp. 4523–4535, 2024.