Deep Reinforcement Learning based Truck Dispatcher for Open-pit Mines

Vaibhav Mukundan, Apurva Narayan

Western University, Canada
vmukund@uwo.ca, apurva.narayan@uwo.ca

Abstract

Truck dispatching has been a critical component of open-pit mining systems, directly influencing productivity and operational efficiency. Effective dispatch orders can directly reduce cycle times and improve throughput. This paper introduces a novel reinforcement learning-based truck dispatching system designed to address scheduling challenges in a dynamic environment. By employing a DDQN model, the approach aims to accomplish the objectives. A centralized agent monitors the entire fleet, leveraging a detailed state representation of trucks, shovels, and destinations. The model is deployed in a simulated environment created based on real-world data to assess it's performance.

Introduction

One of the primary methods for mineral extraction, open-pit mining, accounts for a significant portion of global mineral production involving large-scale operations requiring extensive use of trucks, shovels, and processing plants to transport vast quantities of material. Material transportation alone constitutes more than 50% to 60% of total operational costs, emphasizing the importance of efficient truck dispatching systems (Moradi Afrapoli, Tabesh, and Askari-Nasab 2017; Mirzaei-Nasirabad et al. 2023; Yao et al. 2023). Inefficiencies in scheduling dispatch orders result in significant productivity losses and cost overruns, especially, while considering the dynamics of the environment. Hence, developing advanced dispatching techniques is crucial to address these challenges and achieve optimized operations.

With research being conducted over the years, various methods have been proposed to handle the truck dispatch challenge. Stochastic models have been one of the most commonly used approaches for handling uncertainties in terms of travel times, queue lengths, and equipment availability for improved decision-making processes. For instance, Mirzaei-Nasirabad et al. was able to minimize the fleet waiting times while ensuring deviations from production requirements were minimal. Another study published by Wang et al. introduced a multi-objective programming model for reducing queue times and optimizing production flows based on real-time adjustments.

Recently, there has been a huge shift towards Reinforcement Learning models as an alternative to stochastic models, considering its ability to adapt to dynamic environments. To minimize operational delays and improve fleet productivity under uncertain and variable mining conditions, Noriega and Pourrahimian developed an RL-based framework for dynamically optimizing truck-shovel assignments. Huo, Sari, and Zhang proposed a DDQN framework based on real-time operational conditions to optimize truck dispatching, thereby significantly reducing fuel consumption, and ensuring the greenhouse gas (GHG) reduction targets are met.

This paper proposes a centralized reinforcement learning-based truck dispatching system for open-pit mines. Unlike traditional multi-agent frameworks, the proposed model employs a single agent utilizing a Double Deep Q-Network (DDQN) (Van Hasselt, Guez, and Silver 2016) to optimize truck-shovel-destination assignments dynamically. The action space consists of discrete assignments each represented as a combination of truck, shovel, and destination. The state space encodes operational parameters for trucks, shovels, and destinations. A simulation environment mimics the real-world scenario and provides real-time updates on the state variables, enabling the model to iteratively select optimal actions, execute them in the simulated environment, and receive feedback in the form of rewards. This interaction trains the model to minimize overall cycle times and adapt dynamically to operational uncertainties.

Related Works

While many kinds of research have been conducted throughout history for solving the truck dispatch problem, there were 2 common objectives: (1) Increase productivity and (2) Reduce cost overruns. Since the late 1970s, many mining companies have adopted various mathematical optimization techniques to solve the problem (Munirathinarn and Yingling 1994; White and Olson 1987). Generally, the problem is divided into multiple stages where the goal of the initial stage will be to determine the target material flow rate for each path for each scenario to maximize productivity followed by dynamically assigning trucks in the next stage to meet the objective (White and Olson 1987).

One common approach to solving the problem is creating a multi-objective optimization framework in which the model uses a simulation to replicate dynamic mining operations and various optimization algorithms to balance objectives (Kazemi Ashtiani et al. 2023). Another approach would be utilizing a multi-agent system to create various autonomous agents that represent various trucks to make decentralized decisions based on real-time data to handle various uncertainties (Noriega and Pourrahimian 2018).

However, one of the biggest limitations of mathematical optimization frameworks is scalability when dealing with more variables. This shifted the research towards Reinforcement Learning models. Huo et al. proposed a Q-learning-based framework for optimizing fuel consumption where each truck is considered a separate agent. Another approach was proposed by Noriega, Pourrahimian, and Askari-Nasab where a centralized Double Deep Q-Learning (DDQN) framework was combined with a discrete event simulation (DES) environment to capture the stochastic dynamics of truck-shovel cycles for meeting ore and waste quantity targets. In another study, Zhang et al. developed a multi-agent Deep Q-Learning approach using experience-sharing and memory-tailoring techniques for optimizing dispatch decisions by minimizing queuing and starvation times leading to significant improvements in terms of productivity when compared with traditional methods.

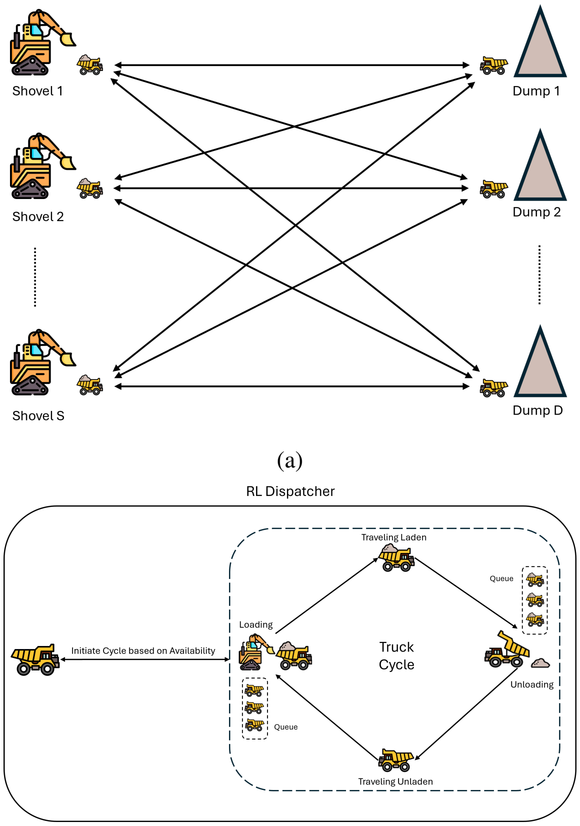
Methodology

The final stage of open-pit mining involves truck dispatching decisions to meet production goals and maintain operational efficiency. A dispatch plan involves assigning trucks to transport materials between shovels and dumps based on factors like shift timing, resource availability, material quality, etc. Consider an open-pit mine having S shovels, D dumps, and T trucks. Each shovel $s_j \in \{s_1, s_2, \ldots, s_S\}$ is responsible for loading the materials onto the truck which gets unloaded at the dump $d_k \in \{d_1, d_2, \ldots, d_D\}$. Trucks $t_i \in \{t_1, t_2, \ldots, t_T\}$ cycle between shovels and dumps, performing various operations as shown in Fig 1a. The dispatcher must consider various factors like truck travel times, queue times at shovels and dumps, and other unpredictable cycle delays making the decision-making process extremely difficult.

In this paper, we employ a centralized Reinforcement Learning framework called Deep Reinforced Agent for Open-Pit Mining (DRAOM) that utilizes the DDQN algorithm (Van Hasselt, Guez, and Silver 2016) to make dispatch decisions. In this, the RL agent interacts with the DES environment created using SimPy (Matsuyama, Klausmann, and Team 2024) which simulates the open-pit mining operation by capturing various real-world elements such as truck movements, shovel operations, and queue dynamics. As shown in fig 1b, the RL Dispatcher monitors the current system state by capturing various activities and assigns optimal tasks to each available truck. The feedback received in the form of rewards based on the activities helps the model's training process improve its overall decision-making ability.

Agent

Instead of considering each truck as a separate agent, we have a centralized dispatcher who acts as the agent, monitors various environmental activities, and assigns the right task to the truck. The agent can make better-informed decisions by having a holistic view of the environment.



(a)

(b)

Figure 1: (a) Environment showcasing the decisions made by the dispatcher. Once the truck has completed loading from the shovel or unloading at the dump, the dispatcher must provide the next location for the truck to carry out its task. (b) Various actions are involved in a truck cycle. A RL Dispatcher decides the actions for each truck based on the current state.

Action Space

The action space is a set of all possible assignments of trucks to shovels and dumps. Each action is represented by a tuple (t_i, s_j, d_k). Considering the fact that only certain resources will be available for a given shift, a masking mechanism is implemented that filters out illegal actions, allowing the agent to choose only the legal ones. This ensures that while the size of the action space remains fixed, the model can dynamically adjust itself based on the resource constraints.

State Space

The state space captures the current status of the system by incorporating all the features of truck, shovel, and dump. The state space is a concatenated array consisting of the following components:

1. **Truck Features:** For each truck t_i, the state includes:
   - σ_{t_i}: Operational status of the truck.
   - l_{t_i}: Loading duration.
   - d_{t_i}: Dumping duration.
   - t_{t_i}: Time spent in laden travel.
   - r_{t_i}: Time spent in unladen travel.
   - c_{t_i}: Payload capacity of the truck.
2. **Shovel Features:** For each shovel s_j, the state includes:
   - a_{s_j}: Availability of the shovel.
   - q_{s_j}: Number of trucks queued at the shovel.
   - g_{s_j}: Ore grade at the shovel.
   - sr_{s_j}: Stripping ratio at the shovel.
3. **Dump Features:** For each dump d_k, the state includes:
   - a_{d_k}: Availability of the dump.
   - q_{d_k}: Number of trucks queued at the dump.

Similar to the action space, all the unavailable resources are marked as null indicating that the current resource is unavailable. This ensures that the model does not consider these resources in its calculation so that they don't affect the predictions for the next state.

Reward Function

The reward function is a critical component that guides the agent towards making optimal decisions. The reward function is given as follows:

$$R = R_{\text{cycle}} + R_{\text{queue}} + R_{\text{site}} + R_{\text{invalid}} + R_{\text{completion}}, \quad (1)$$

R_{cycle} is the difference in total cycle time across all trucks between the current (t) and next (t+1) state. R_{cycle} is given by:

$$R_{\text{cycle}} = \frac{1}{1000} \cdot \left(\sum_{i=1}^{T} (l_{t_i} + d_{t_i} + t_{t_i} + r_{t_i}) - \sum_{i=1}^{T} (l_{t_i}^{'} + d_{t_i}^{'} + t_{t_i}^{'} + r_{t_i}^{'}) \right), \quad (2)$$

where l_{t_i}, d_{t_i}, t_{t_i}, and r_{t_i} are the loading, dumping, laden travel, and unladen travel times for truck t_i, and $l_{t_i}^{'}$, $d_{t_i}^{'}$, $t_{t_i}^{'}$, $r_{t_i}^{'}$ are the corresponding times in the next state.

R_{queue} adds a penalty based on the queues at shovels and dumps.

$$R_{\text{queue}} = -2 \cdot \left(\sum_{j=1}^{S} q_{s_j}^{'} + \sum_{k=1}^{D} q_{d_k}^{'} \right), \quad (3)$$

where $q_{s_j}^{'}$ and $q_{d_k}^{'}$ are the queue lengths at shovel s_j and dump d_k in the next state.

R_{site} calculates the reward based on the ore grade and stripping ratio at all the shovels where the trucks are currently queued.

$$R_{\text{site}} = \sum_{j=1}^{S} (q_{s_j}^{'} > 0) \cdot \left(\frac{g_{s_j}^{'}}{100} - \frac{sr_{s_j}^{'}}{400} \right), \quad (4)$$

where $g_{s_j}^{'}$ is the ore grade, and $sr_{s_j}^{'}$ is the stripping ratio at shovel s_j in the next state.

$R_{invalid}$ adds a penalty when the truck is assigned to perform an unauthorized action while $R_{completiion}$ rewards the model when the task is completed.

Model Training

The ability of the DDQN model to mitigate the overestimation bias of traditional Q-Learning by decoupling the action selection and evaluation processes makes it suitable for our problem. At each step, the agent chooses an action a_t using epsilon-greedy strategy (Sutton and Barto 2018) where either the random action is chosen based on probability ϵ or the best action maximizing the Q-value is chosen with probability $1 - \epsilon$. After observing the next state s_{t+1} and reward r_t, the Q-value is updated as follows:

$$Q_{\text{new}}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot \Big[r_t + \gamma \cdot \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t) \Big], \quad (5)$$

where α is the learning rate and γ is the discount factor.

During the training process, the experiences are sampled from the relay buffer and calculated using the target network to calculate the target Q-values. The loss is then calculated by applying CrossEntropyLoss between the predicted and target Q-values. The agent then updates the primary network through backpropagation and performs a soft update of the target network to stabilize learning. The Hyperparameters used for training the model are listed in table 1.

Hyperparameter	**Value**
Epsilon Decay (ϵ_{decay})	0.99
Discount Factor (γ)	0.99
Learning Rate (α)	1×10^{-6}
Soft Update Factor (τ)	0.001
Batch Size	64
Replay Buffer Size	10,000

Table 1: DDQN model Hyperparameters

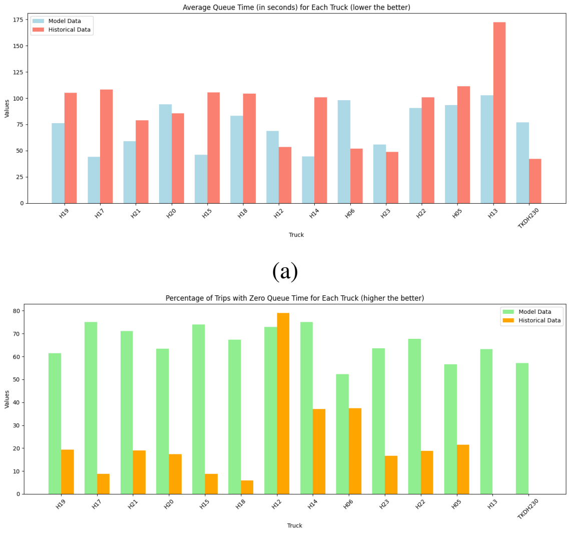
Once in every 20 episodes, certain resources are made unavailable thereby ensuring that the model also learns how to make decisions with the available resources.

Results

The model was tested in a simulation environment having 14 trucks, 6 shovels, and 11 dumps across four 6-hour shifts with varying resource availability. The results were compared with those generated from a heuristic model originally used to make decisions on the site.

Queue Time Evaluation

The RL model has showcased its ability to make optimized decisions by reducing the average queue time for various trucks. Fig 2a demonstrates that the RL model has reduced the overall queue time by an average of 12% with significant performance improvements for some trucks, while others show marginal improvements. On the other hand, all trucks achieve around 65% of trips with zero-queue delays as indicated in fig 2b which is a significant improvement over the traditional method.



(a)

(b)

Figure 2: (a) Average queue time (in seconds) for each truck. (b) Percentage of trips with zero queue time for each truck.

Average Cycle Time Evaluation

The time a truck takes to complete a cycle is defined as the cumulative sum of queue time, loading time, dumping time, and the time it takes to travel from one place to another place. Although there is a general reduction in the average cycle time for each truck as shown in fig 3, some trucks spend more cycle time on average than their traditional counterparts. This may occur because the RL dispatcher chooses locations far apart to reduce the time wasted in queues.

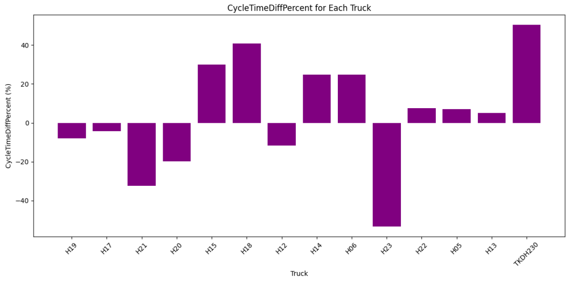


Figure 3: Percentage difference in cycle time for each truck.

Conclusion

In this paper, we developed an RL model combined with a DES environment to simulate an open-pit mine. While the model outperformed the heuristic model by reducing the average queue time and cycle time of a truck resulting in an overall increase in productivity, the model was trained based on assuming values for ore grades, stripping ratios for shovels, and fixed payload for each truck thereby not completely capturing the real-world variability. Future work will focus on enhancing the model by integrating production rates, and variable payloads for trucks, and utilizing actual ore grade and stripping ratio values to demonstrate more accurate and realistic performances.

References

Huo, D.; Sari, Y. A.; Kealey, R.; and Zhang, Q. 2023. Reinforcement Learning-Based Fleet Dispatching for Greenhouse Gas Emission Reduction in Open-Pit Mining Operations. *Resources, Conservation & Recycling*, 188: 106664.

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

Kazemi Ashtiani, M.; Afrapoli, A. M.; Doucette, J.; and Askari-Nasab, H. 2023. Optimizing Green Truck Dispatch in an Open Pit Mine through a Multi-Objective Simulation and Optimization Framework with Fuel Consumption Consideration. *MOL Report Eleven*, 104: 1–8.

Matsuyama, O.; Klausmann, S.; and Team, S. D. 2024. SimPy: Discrete Event Simulation for Python. https://simpy.readthedocs.io/en/latest/. Accessed: 2024-11-13.

Mirzaei-Nasirabad, H.; Mohtasham, M.; Askari-Nasab, H.; and Alizadeh, B. 2023. An optimization model for the real-time truck dispatching problem in open-pit mining operations. *Optimization and Engineering*, 24: 2449–2473.

Moradi Afrapoli, A.; Tabesh, M.; and Askari-Nasab, H. 2017. An Investigation into Dispatch Optimizers using Truck-Shovel Simulation and a New Multi-Objective Truck Dispatching Technique. *MOL Report Eight*, 201: 1–10.

Munirathinarn, M.; and Yingling, J. C. 1994. A review of computer-based truck dispatching strategies for surface mining operations. *International Journal of Surface Mining Reclamation and Environment*, 8: 1–15.

Noriega, R.; and Pourrahimian, Y. 2018. A Multiagent System for Truck Dispatching in Open-pit Mines. In *Proceedings of the 27th International Conference on Autonomous Agents and Multiagent Systems*, 189–196.

Noriega, R.; and Pourrahimian, Y. 2023. Truck Fleet Dispatching in Open Pit Mines Using Artificial Intelligence Methods. *MOL Report Eleven*, 203: 1–10.

Noriega, R.; Pourrahimian, Y.; and Askari-Nasab, H. 2025. Deep Reinforcement Learning Based Real-Time Open-Pit Mining Truck Dispatching System. *Computers and Operations Research*, 173: 106815.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2nd edition.

Van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep Reinforcement Learning with Double Q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1): 2094–2100.

Wang, X.; Dai, Q.; Bian, Y.; Xie, G.; Xu, B.; and Yang, Z. 2023. Real-time truck dispatching in open-pit mines. *International Journal of Mining, Reclamation and Environment*, 37(7): 504–523.

White, J. W.; and Olson, J. P. 1987. Computer-based dispatching in mines with concurrent operating objectives. In *Proceedings of the 18th International Symposium on the Application of Computers and Operations Research in the Mineral Industry*, 118–127.

Yao, J.; Wang, Z.; Chen, H.; Hou, W.; Zhang, X.; Li, X.; and Yuan, W. 2023. Open-Pit Mine Truck Dispatching System Based on Dynamic Ore Blending Decisions. *Sustainability*, 15(4): 3399.

Zhang, C.; Odonkor, P.; Zheng, S.; Khorasgani, H.; Serita, S.; and Gupta, C. 2020. Dynamic Dispatching for Large-Scale Heterogeneous Fleet via Multi-agent Deep Reinforcement Learning. In *arXiv preprint arXiv:2008.10713*.