Optimization-Based Dispatching Policies for Open-Pit Mining

Amanda Smith*[1], Jeff Linderoth†[1,2], and Jim Luedtke‡[1,2]

[1] *Department of Industrial and Systems Engineering, University of Wisconsin-Madison*
[2] *Wisconsin Institute for Discovery, Uniuversity of Wisconsin-Madison*

February, 2020

Abstract

We propose, implement, and test two approaches for dispatching trucks in an open-pit mining operation. The first approach relies on a nonlinear optimization model that incorporates queueing effects to set target average flow rates between mine locations. The second approach is based on a time-discretized mixed integer programming (MIP) model. The MIP model is difficult to solve exactly in real-time operations, so we present a heuristic that quickly produces high-quality feasible solutions. We give computational results demonstrating the effectiveness of the proposed heuristics and several key model components. We test the dispatching policies and model features by building a discrete-event simulation of an open-pit mine. We present a full computational study of the two policies in which we perform output analysis on key metrics of the open-pit mine simulation. Results indicate that the MIP-based dispatching policy consistently outperforms the more commonly-used average flow rate dispatching policy on open-pit mines in a wide variety of operations settings.

1 Introduction

We propose two policies to make dispatching decisions for open-pit mining operations, and via a discrete-event simulation, we demonstrate how the policies perform on open-pit mines with different characteristics. The first policy relies on matching immediate dispatching decisions to target rates of trucks moving between pairs of locations and is similar to other dispatching methods proposed in the literature. For the second policy, we present a mixed-integer programming (MIP) model of an open-pit mine truck dispatching problem that introduces new approaches for modeling the effects of queueing and blending.

1.1 Truck Dispatching in Open-Pit Mines

An open-pit mine is a large area in which trucks move ore between different locations. The mine consists of mining sites and processing sites. Ore is extracted from the mining sites and placed in

*`amanda.smith@wisc.edu`
†`linderoth@wisc.edu`
‡`jim.luedtke@wisc.edu`

large trucks that carry the ore to the processors. The ore extracted from different sites may have different quality levels. When trucks arrive at the processors, they dump the ore into short-term stockpiles from which ore is fed into the processor at an (ideally) fixed rate. If the size of the short-term stockpiles falls below a given threshold, the processors are slowed to avoid complete shutdown. Once processed, the ore is taken outside the mine to be sold or converted into other products. The ore loaded into the processor should ideally meet minimum quality standards, but keeping the processors fed and extracting a sufficient amount of ore from each mining site are higher priorities.

To make routing decisions during the daily operations of the mine, a dispatcher typically has access to the mine's processing, mining, and quality targets and knows the current state of the system. When a truck completes a task, it must receive a new destination from the dispatcher. Each dispatching decision should make progress in meeting the mine's processing, mining, and ore quality goals. These goals tend to compete, forcing decision makers to find an acceptable balance between them.

The dispatching problem has a number of complicating components. The open-pit mine is a dynamic stochastic environment, so dispatching decisions can quickly become obsolete. Decisions must be made in real time to keep the trucks active. There is limited loading and unloading capacity at each site, so trucks often enter a queue before they can be loaded or unloaded. Because different mining sites have different ore qualities, decision makers must account for the effects of blending ore from different sites. Finally, the dispatching problem has three potentially competing objectives: meeting processing rate targets, meeting mining extraction targets, and meeting ore quality targets, and it is difficult to create a dispatching rule that optimizes all three objectives concurrently. The combination of all these factors makes the dispatching problem difficult to model and challenging to solve in practice.

1.2 Literature Survey

A number of different rules have been used for truck dispatching in mining operations. One method is to solve a medium-term "rate" problem that assigns target flow rates of trucks between pairs of locations. The method then uses an assignment problem based on the the target rates to make dispatching decisions. The first part of this approach, the rate-setting problem, has been explored by a number of authors. One approach is to use simulation or queueing theory to produce schedules (Kappas and Yegulalp, 1991; Oraee and Asi, 2004; Mena et al, 2013). Other authors use optimization models and techniques to solve the rate problem, including mixed-integer programming (MIP) models (Goodman and Sarin, 1988; Naoum and Haidar, 2000; Burt, 2008; Faraji, 2013; Smith and Wicks, 2014), stochastic optimization models (Ta et al, 2005; Topal and Ramazan, 2012), or rate-setting linear programming (LP) models (Ercelebi and Bascetin, 2009; Choudhary, 2015).

Given target flow rates, a number of different approaches exist to use these rates to implement a real-time dispatching policy. Early work (Elbrond and Soumis, 1987a; Soumis et al, 1989; White et al, 1993) considered integrating the rate-setting problem with the immediate dispatching decision by using LP and assignment problems for the real-time decisions. The work (Temeng, 1998) models dispatching with goal constraints to enforce ore quality and demonstrates significant improvement in ore quality compared to traditional LP models. The authors of (Ahangaran et al, 2012) build a integer programming model with an objective to minimize the cost of loading and transportation.

The model does not account for the timing of decisions; a decision is made at a fixed point in time without regard to the future state of the system. The constraints in the model include hard bound constraints on the acceptable ore quality and ore volume, and an algorithm for solving the model is presented.

Heuristic dispatching rules are commonly used to determine where a truck should go next. These rules are often embedded in a simulation in order to demonstrate how the current decision will affect the future state of the system. The work presented in (Subtil et al, 2011) finds the optimal number of trucks within the mine and uses a simulation-based heuristic that combines simulation and a multi-criterion LP model for dispatching. A heuristic known as the 1-truck-for-N-shovels strategy compares the benefit and cost of sending an available truck to any of the N mining sites but ignores the potential effects of the given truck's interactions with other trucks, such as queueing (Chatterjee and Brake, 1981; Tu and Hucka, 1985; Lizotte and Bonates, 1987; Sadler, 1988; Li, 1990; Bonates, 1992; Forsman et al, 1993; Panagiotou and Michalakopoulos, 1995; Ataeepour and Baafi, 1999). A related strategy known as the M-trucks-for-1-shovel strategy ranks mine sites by how much additional material needs to be removed to meet the target and finds the next M trucks to be dispatched. This strategy ignores many of the interaction effects between system components over time (White et al, 1982; Arnold and White, 1983). Finally, it is possible to combine these two strategies into an M-trucks-for-N-shovels strategy to better account for the interconnected nature of the mining system. The works (Hauck, 1973, 1979; Munirathinam and Yingling, 1994) suggest single-stage systems to determine a sequence of dispatching decisions by solving a series of assignment problems. The works (Elbrond and Soumis, 1987b; Temeng et al, 1997) extend this to a multi-stage system, but the models do not allow assigning multiple trucks to a single location and cost calculations do not account for future queueing effects. The important practical problem of dispatching trucks for mining operations has received significant attention from the research community, but there is more that could be done.

1.3 Contributions

To make real-time dispatching decision in an open-pit mine, we propose and evaluate two novel dispatching policies. Our main contributions are a detailed, multi-objective nonlinear model for setting target average flow rates and a discrete-time multi-objective MIP model of mining operations. The solution of the average flow-rate problem is provided as input to a greedy target-matching policy that dispatches trucks so as to match the target flow rates as closely as possible. The solution of the MIP-model is used to create an ordered list of next dispatch locations for each site that can be used as a dispatching policy.

One limitation of existing target-based average flow rate dispatching methods in the literature is that they do not explicitly account for queueing effects of multiple trucks at the same location. Our nonlinear flow-rate model includes novel constraints to capture the effects of (M/G/1) queueing at mine locations. The queueing constraints are shown to be representable as rotated second-order cone constraints, so the overall average-flow rate problem remains tractable. To make dispatch decisions, we propose to match available trucks to the sequence of two locations farthest from meeting their target flow rates. This greedy policy, like the heuristics described in literature, is myopic, only considering the next decision to be made with little regard to how it will affect the future system.

The time-discretized MIP model we build accounts for many of the challenges the dispatching problem poses. Specifically, a drawback of the existing dispatch strategies is that the future effects of queueing at different locations in the mine is ignored. A contribution of our approach is the incorporation of a family of constraints that model the effects of queueing by approximating the maximum number of trucks that can be loaded or unloaded at each site within different windows of time. In addition, the nonconvex nature of the standard blending model poses a computational challenge when attempting to control ore quality at the processing sites. Our MIP model uses a family of linear constraints, implemented in a moving-window fashion, to approximate the nonconvex blending conditions and control ore quality at the processing sites. We describe a problem-specific relaxation induced neighborhood search (RINS) heuristic algorithm and demonstrate the algorithm can obtain high-quality solutions in at most 90 seconds, which is rapid enough for real-time operations.

To test the impact of our novel model characteristics and to compare the dispatching policies, we build a discrete-event simulation of an open-pit mining operation. A primary contribution of our work is a demonstration that the MIP-based dispatching model consistently outperforms the more commonly-used average flow-rate dispatching approach in a wide variety of operational settings.

We first describe the multi-objective nonlinear flow-rate model and a two-phase dispatching policy in Section 2. In Section 3, we describe the multi-objective discrete-time MIP model, the associated dispatching policy, and a problem-specific relaxation-induced neighborhood search (RINS) heuristic. Section 4 contains the results of a computational study that demonstrates that the RINS heuristic can provide high quality solutions within operational time constraints. Finally, in Section 5, we describe comprehensive computational studies based on a discrete-event simulation of open-pit mining operations. The studies demonstrate the positive impact of many of the novel model features we introduce. We also compare the performqance of teh two dispatching policies on mines with different characteristics, such as number of trucks, ore qualities, and travel times.

2 Average Flow-Rate-Based Dispatching Policy

In the truck dispatching problem, each time a truck finishes being loaded or unloaded, some form of dispatching function must be used to acquire the truck's next destination. In this section, we describe a dispatching policy that can be used to determine the next destination for any truck in the open-pit mine. This policy is an example of the two-phase policy procedure typical in the literature in which an average rate model is solved, yielding target flow rates. Next, when a dispatching decision is to be made, a heuristic is used to match the dispatching decisions to the target flow rates. We propose a version of this two-phase approach involving a nonlinear (second-order cone-constrained) first-phase problem that incorporates queueing effects. For the second phase, we implement a greedy target-matching heuristic that matches available trucks to dispatching locations to minimize the deviation from target flow rates.

2.1 Average Flow Rate Model

In this section, we describe an average flow-rate model that is solved at regular intervals or any time a significant disruption occurs in the system (such as an equipment failure). Given target processing and mining rates and target ore quality, solving the model yields the average flow rates of trucks to each location that meet the targets as closely as possible, subject to physical constraints. Decision variables in the flow-rate model determine the violation of loading volume, unloading volume, and ore quality targets, as well as the rate of trucks traveling between pairs of locations in the mine. Constraints in this model measure violation of the target parameters, balance flow into and out of locations in the mine, bound the total trucks in use at any given time, and track the average queue lengths using a nonlinear M/G/1 queueing formula. The objective is to meet the processing rate, mining rate, and ore quality targets, with the greatest emphasis given to meeting the processing rate targets.

2.1.1 Sets and Parameters

The input data for the flow-rate model describe how the system should be operating in steady state. The mine consists of a set of locations, N which is partitioned into mining sites M, processing sites P, dump sites D, and long-term stockpile sites S. The set M is partitioned into sites where ore is extracted (ore mines) G and sites where overburden is extracted (waste mines) B. Material taken from ore mines is of high-enough quality to be processed or taken to a long-term stockpile. In our model, we treat long-term stockpiles as storage sites for material from ore mines, so trucks at ore mine sites can be dispatched to either a processing site or a long-term stockpile. Material taken from waste mines must be taken to a dump site. There is also a set of truck sizes K, where a truck of type $k \in K$ can hold α_k tons of ore. There are Γ_k trucks of type $k \in K$.

Trucks travel between pairs of locations, and each route has a parameter representing travel time, t_{ij} for $i \in N$, $j \in N$. Each site $i \in N$ has an upper ($\overline{a}_i$) and lower ($\underline{a}_i$) bound on the tons of ore that can be either extracted (at a mine) or processed (at a processing, dump, or stockpile site) each hour. There is also a set of attributes, R, that are tracked to measure ore quality. There are upper ($\overline{Q}_{jr}$) and lower ($\underline{Q}_{jr}$) bounds on attribute percentages for all processing sites $j \in P$ and for each attribute $r \in R$. Each ore mine $i \in G$ has parameters Q_{ir} representing the average attribute percentage for each $r \in R$. Each site $i \in N$ can serve trucks with a mean service time of $1/\mu_i$ hours/truck and a standard deviation of σ_i hours/truck. We assume a general distribution for the service times. These data are summarized in Table 1.

2.1.2 Decision Variables and Constraints

For each site $i \in N$, variables v_i^ℓ measure the rate (in tons/hour) by which the lower bound $\underline{a}_i$ is violated. Similarly, v_i^u measures the tons/hour by which the upper bound $\overline{a}_i$ is violated. The variables f_{ijk} represent the trucks/hour of type $k \in K$ traveling from $i \in N$ to $j \in N$, so the value $\sum_{k \in K} \alpha_k f_{ijk}$ gives the total tons of material traveling from i to j in each hour. The full set of

Parameter	Description	Units
α_k	Size of truck type $k \in K$	Tons/truck
Γ_k	# of trucks of type k	Trucks
t_{ij}	Time required to travel from $i \in N$ to $j \in N$	Hours
$\overline{a}_i$	Maximum production rate at site $i \in N$	Tons/hour
$\underline{a}_i$	Minimum production rate at site $i \in N$	Tons/hour
$\overline{Q}_{jr}$	Upper bound on grade-element ratio of attribute $r \in R$ at production site $j \in P$	(%)
$\underline{Q}_{jr}$	Lower bound on grade-element ratio of attribute $r \in R$ at production site $j \in P$	(%)
Q_{ir}	grade-element ratio of attribute $r \in R$ at location $i \in G$	(%)
$1/\mu_i$	Mean service time at $i \in M \cup P$	Hours/truck
σ_i	Standard deviation of service time at site $i \in M \cup P$	Hours/truck

Table 1: A summary of all parameters and data used in the flow rate model.

constraints measuring rate bound violations is

$$v_i^\ell \geq \underline{a}_i - \sum_{j \in P \cup S} \sum_{k \in K} \alpha_k f_{ijk} \qquad \forall i \in G \tag{1}$$

$$v_i^u \geq \sum_{j \in P \cup S} \sum_{k \in K} \alpha_k f_{ijk} - \overline{a}_i \qquad \forall i \in G \tag{2}$$

$$v_i^\ell \geq \underline{a}_i - \sum_{j \in D} \sum_{k \in K} \alpha_k f_{ijk} \qquad \forall i \in B \tag{3}$$

$$v_i^u \geq \sum_{j \in D} \sum_{k \in K} \alpha_k f_{ijk} - \overline{a}_i \qquad \forall i \in B \tag{4}$$

$$v_j^\ell \geq \underline{a}_j - \sum_{i \in M} \sum_{k \in K} \alpha_k f_{jik} \qquad \forall j \in P \cup S \cup D \tag{5}$$

$$v_j^u \geq \sum_{i \in M} \sum_{k \in K} \alpha_k f_{jik} - \overline{a}_j \qquad \forall j \in P \cup S \cup D. \tag{6}$$

Similarly, we measure the amount by which the lower and upper bounds on each attribute $r \in R$ are not met at each processing site $j \in P$. To do so, we introduce variables h_{jr}^ℓ and h_{jr}^u to represent the violation of the lower and upper bounds, respectively. The bound violation is calculated as the non-negative difference between the lower or upper bound on ore quality and the actual quality of the ore brought from all ore mining sites G:

$$h_{jr}^\ell \geq -\left(\sum_{i \in G} \sum_{k \in K} \alpha_k (Q_{ir} - \underline{Q}_{jr}) f_{ijk} \right) \qquad \forall j \in J, \forall r \in R \tag{7}$$

$$h_{jr}^u \geq \sum_{i \in G} \sum_{k \in K} \alpha_k (Q_{ir} - \overline{Q}_{jr}) f_{ijk} \qquad \forall j \in J, \forall r \in R. \tag{8}$$

We balance the flow rate of trucks into each location $i \in N$ with the flow rate of trucks out of that location:

$$\sum_{j\in N\setminus M} f_{jik} = \sum_{j\in P\cup S} f_{ijk} \qquad \forall i \in G, \forall k \in K \tag{9}$$

$$\sum_{j\in N\setminus M} f_{jik} = \sum_{j\in D} f_{ijk} \qquad \forall i \in B, \forall k \in K \tag{10}$$

$$\sum_{j\in B} f_{jik} = \sum_{j\in M} f_{ijk} \qquad \forall i \in D, \forall k \in K \tag{11}$$

$$\sum_{j\in G} f_{jik} = \sum_{j\in M} f_{ijk} \qquad \forall i \in P, \forall i \in S, \forall k \in K. \tag{12}$$

To incorporate the effects of queueing at different sites, we introduce decision variables λ_i that represent the arrival rates at each location $i \in N$, written as a function of truck flow rates:

$$\lambda_i = \sum_{j\in P\cup D\cup S} \sum_{k\in K} f_{jik} \qquad \forall i \in M \tag{13}$$

$$\lambda_j = \sum_{i\in G} \sum_{k\in K} f_{ijk} \qquad \forall j \in P \cup S \tag{14}$$

$$\lambda_j = \sum_{i\in B} \sum_{k\in K} f_{ijk} \qquad \forall j \in D. \tag{15}$$

Specifically, the arrival rate at $i \in M$ is the total flow in trucks/hour coming from a processing site, dump site, or long-term stockpile. The arrival rate at $j \in P \cup S$ is the total flow in trucks/hour coming from an ore mine, and the arrival rate at $j \in D$ is the total flow in trucks/hour coming from a waste mine.

We choose to model queueing in an open-pit mine as an M/G/1 queue. We introduce the variable $L_i, i \in N$ to represent the average length of the queue of trucks at location $i \in N$, and we bound L_i from below by the formula for the average number of customers in the system for an M/G/1 queue:

$$L_i \geq \frac{\lambda_i}{\mu_i} + \frac{\lambda_i^2(1/\mu_i^2 + \sigma_i^2)}{2(1-\lambda_i/\mu_i)} \qquad \forall i \in N.$$

The use of the M/G/1 queueing model assume that for each $i \in N$ service times follow a general distribution with mean service time $1/\mu_i$ and standard deviation σ_i, and arrival rates follow a Poisson distribution with the parameter λ_i. This is of course an approximation, since arrivals at sites are dependent on the arrivals and departures of other sites in the network. Although not necessarily representative of the true arrival rates, this approximation improves on the typical assumption that there is no queueing of trucks and does so in a computationally tractable way. We can exploit the convexity of the M/G/1 queueing constraint by reformulating it as a second-order cone constraint, which can be handled efficiently with commercial solvers. We rewrite the queueing constraint by introducing vectors of auxiliary variables y and z. The constraints

$$y_i = L_i - \frac{\lambda_i}{\mu_i} \qquad \forall i \in N \tag{16}$$

$$z_i = 2\left(1 - \frac{\lambda_i}{\mu_i}\right) \qquad \forall i \in N \tag{17}$$

$$y_i z_i \geq (1/\mu_i^2 + \sigma_i^2)\lambda_i^2 \qquad \forall i \in N \tag{18}$$

$$y_i, z_i \geq 0 \qquad \forall i \in N \tag{19}$$

are a reformulation of the constraint describing the average length of the M/G/1 queue. Given the average number of the trucks in the queue L_i, we can constrain the total number of trucks in the system at any given time:

$$\sum_{i \in N} L_i + \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} t_{ij} f_{ijk} \leq \sum_{k \in K} \Gamma_k. \tag{20}$$

Inequality (20) states that the average number of trucks waiting or being served plus the average number of trucks traveling cannot exceed the total number of trucks.

We also enforce non-negativity on all decision variables:

$$v_i^\ell, v_i^u, \lambda_i, L_i \geq 0 \qquad \forall i \in N \tag{21}$$

$$f_{ijk} \geq 0 \qquad \forall i \in N, \forall j \in N, \forall k \in K \tag{22}$$

$$h_{jr}^\ell, h_{jr}^u \geq 0 \qquad \forall j \in P, \forall r \in R. \tag{23}$$

All decision variables given in this section are summarized in Table 2. To simplify notation, let X^N be the set of solutions which satisfy all constraints:

$$X^N := \{y = (f, v^\ell, v^u, \lambda, h^\ell, h^u, y, z, L) : (1) - (23)\}.$$

Then a solution $x \in X^N$ is a feasible solution to the average flow-rate model.

2.1.3 Objective

We use a three-phase hierarchical objective structure to model the relative importance of each objective in this inherently multi-objective problem. The hierarchy prioritizes meeting processing rates, followed by meeting the mine extraction targets, and finally ore quality targets if possible. The most important goal is to keep the processor supplied with ore at all times, so we first solve the model with constraints minimizing deviation from processing volume targets and the objective

$$\min_{x \in X^N} \sum_{i \in P} (v_i^\ell + v_i^u).$$

Let $\hat{v}_j^\ell$ and $\hat{v}_j^u$ be the values of the processing rate variables in a solution to this problem. In the next level of the hierarchy, we bound the variables v_j^ℓ and v_j^u for $j \in P$ and minimize the maximum deviation from the mining rate targets:

$$\min_{x \in X^N} \max_{i \in M} \{v_i^\ell, v_i^u\}$$
$$\text{s.t. } v_j^\ell \leq \hat{v}_j^\ell, v_j^u \leq \hat{v}_j^\ell \qquad \forall j \in J.$$

Variable	Description	Units
f_{ijk}	Flow rate of trucks of type $k \in K$ from $i \in N$ to $j \in N$	Trucks/hour
v_i^ℓ	Violation of minimum loading/unloading rate bound at site $i \in N$	Tons/hour
v_i^u	Violation of maximum loading/unloading rate bound at site $i \in N$	Tons/hour
λ_i	Arrival rate of trucks at location $i \in N$	Trucks/hour
h_{jr}^ℓ	Violation of minimum grade for attribute $r \in R$ at processing site $j \in P$	Tons
h_{jr}^u	Violation of maximum grade for attribute $r \in R$ at processing site $j \in P$	Tons
L_i	Average number of trucks waiting in queue or being loaded or unloaded at location $i \in N$	Trucks

Table 2: A summary of primary decision variables used in the flow rate model

If any of the processing rate violation variables are positive, we terminate with the solution at the second level of the hierarchy. Otherwise, after solving the second-level problem, we store the values $(\hat{v}_i^\ell, \hat{v}_i^u)$ for all mining sites $i \in M$ and introduce bounds on the mining target violations for each mining site based on the solution. The bounds on mining targets are set to be slightly larger than their optimal values in the second-level problem to give some additional flexibility:

$$v_i^\ell \leq \hat{v}_i^\ell + 0.01\underline{a}_i \quad \forall i \in M \tag{24}$$

$$v_i^u \leq \hat{v}_i^u + 0.01\overline{a}_i \quad \forall i \in M. \tag{25}$$

We fix the processing rate violation variables to 0 and solve the final phase:

$$\begin{aligned} \min_{x \in X^N} \; & \sum_{j \in P} \sum_{r \in R} (h_{jr}^\ell + h_{jr}^u) \\ \text{s.t.} \; & v_j^\ell = 0, v_j^u = 0 \qquad \forall j \in P \\ & v_i^\ell, v_i^u \;\; \text{satisfy } (24-25) \; \forall i \in M. \end{aligned}$$

In implementation, the average flow-rate model is solved regular intervals to find target flow rates for each truck type between each pair of locations. We denote the target rates for each truck type $k \in K$ coming from solution of this model by $\hat{f}_{ijk}$ for an origin-destination pair (i, j).

2.2 Target-Matching Heuristic

Having found a solution to the average flow model with flow values $\hat{f}$, we need a policy for making real-time dispatching decisions to match the target rates. Let T be the length of the planning horizon, and denote the current time t, for $t \in [0, T]$. We denote the *actual* total flow of trucks (either in simulation or an actual implementation) that have traveled between two locations by time t by r_{ijk} for each origin-destination pair $(i, j) \in N \times N$ and each truck type $k \in K$.

Each time a dispatching decision must be made, we check the list of possible pairs of trips for the available truck starting at the truck's location, $i \in N$. For each possible pair of destinations $(j, \ell) \in N \times N$, we compute the difference between the target flow on that round trip at the current time t, $(\hat{f}_{ijk} + \hat{f}_{j\ell k})(t/T)$, and the actual flow $(r_{ijk} + r_{j\ell k})$. With $i \in N$ and $k \in K$ fixed, let

$$(j^*, \ell^*) \in \arg \max_{(j,\ell) \in (N \times N)} \{(\hat{f}_{ijk}(t/T) - r_{ijk}) + (\hat{f}_{j\ell k}(t/T) - r_{j\ell k})\}.$$

We choose to dispatch the truck to the location j^*. Finally, we update actual flow for that route: $r_{ij^*k} \leftarrow r_{ij^*k} + 1$. If two or more locations satisfy the argmax requirement, we break the tie arbitrarily. Although computationally simple, the greedy heuristic is myopic in that only a single truck's next round-trip is considered. Since no other trucks are considered, we cannot take advantage of combining strategies for different trucks.

3 Discrete-Time MIP Dispatching Policy

In this section, we propose a discrete-time MIP model that can be used to define an alternative dispatching policy. Given target processing volumes, mining volumes, and ore qualities, solving the model yields decisions to meet the targets as closely as possible, while obeying physical constraints on the system. The horizon should be long enough that the current decisions take into account the future state of the system, but short enough that a solution to the MIP instance can be obtained quickly. Decision variables in the dispatching model determine the violation of extraction, processing rate, and ore quality targets; the number of trucks queued, loaded and unloaded, and available at each site within the mine; the number of trucks traveling between pairs of locations in the mine; and the volume of ore in the short-term stockpiles. Constraints in the dispatching model measure violation of the target parameters, balance flow into and out of locations in the mine, bound the total trucks that can be loaded (unloaded) at each site in the mine in a given window of time, and track the short-term stockpile level over time. Decision variables associated with numbers of trucks must assume integer values. The objective is to minimize the violation of all three targets: processing rate, extraction, and ore quality, with the greatest emphasis given to meeting the processing rate targets.

3.1 Parameters

We define the dispatching model for a certain length of time $T^{\max}$, using a set of discrete time periods $T := \{1, 2, \ldots, T^{\max}\}$. We extend the horizon to include some past time periods for ease of notation and denote the extended horizon by $T^H = \{-\max_{ij} \tau_{ij}, \ldots, 0, 1, T^{\max}\}$, where τ_{ij} represents the number of time periods required to travel between location $i \in N$ and location $j \in N$.

The input data for the dispatching problem describe the current state of the system. Trucks travel between pairs of locations, and each route has a parameter representing travel time in number of periods, τ_{ij} for $i \in N$, $j \in N$. The average number of trucks that can be loaded or unloaded in each period is $\bar{p}_i$ for all $i \in N$. Note that we do not assume $\bar{p}_i$ is an integer for any $i \in N$. Each ore mine $i \in G$ has parameters Q^M_{ir} representing the attribute percentage for each attribute $r \in R$. Although the true ore quality is uncertain, it is reasonable to assume this value can be estimated for short-term planning horizons based on recent observations of ore quality. We will treat Q^M_{ir} as a random variable in simulation studies of mine operations in Section 5.1. We also are given total extraction targets Θ^M_i for the planning horizon (in tons) for each mining site $i \in M$. Each processing site $j \in P$ has a target processing rate Θ^P_j, which (ideally) is a constant rate of depletion, in tons/period, unless the short-term stockpile becomes empty, in which case the processor must stop. The short-term stockpiles for each processing site $j \in P$ have an initial amount of ore, I_{j0} that has an initial attribute percentage, Q^I_{jr0}, for each attribute $r \in R$. There are ore quality targets Q^P_{jr} for each processing site $j \in P$ and for each attribute $r \in R$. The number of trucks of type $k \in K$ en route from $i \in N$ to $j \in N$ that began their trip at period $t \in T^H \setminus T$ are given by $\hat{x}_{ijkt}$. Trucks of type $k \in K$ available for dispatching at site $i \in N$ at the start of the planning horizon are denoted by $\hat{g}_{ik}$. Finally, trucks that are in queue at period 0 at mining site $i \in M$ are given by $\hat{q}^M_{ik}$ and trucks in queue at processing site $j \in P$ that have arrived from mine $i \in M$ are given by $\hat{q}^P_{ijk}$. These data are summarized in Table 3.

3.2 Decision Variables and Constraints

The dispatching problem includes decision variables to describe the new state of the system and variables indicating where trucks should be sent in each period. In particular, x_{ijkt} is the number of trucks sent from $i \in N$ to $j \in N$ of type $k \in K$ in period $t \in T$. The number of trucks loaded at $i \in M$ of type $k \in K$ in period $t \in T$ is represented by the variable y^M_{ikt}. The number of trucks of type $k \in K$ from mine $i \in G$ unloaded at $j \in P$ in period $t \in T$ is y^P_{ijkt}.

We balance queues of trucks both before and after each truck is loaded or unloaded at a site. In particular, at every mine $i \in M$, for each type of truck $k \in K$, we have the structure depicted in Figure 1. All trucks that arrive enter a queue that they leave as soon as the loading site becomes available. After the truck is loaded, it becomes available for dispatching. The truck may wait before it departs to another destination in the mine, so we also keep track of how many trucks are available at each site.

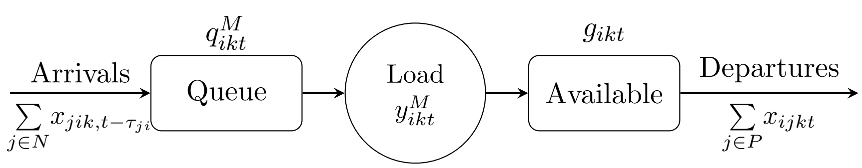


Figure 1: Example of the workflow at a mining site

Processing site queues have a similar structure, but there are separate queue and unloading variables for each possible origin (ore mining site), since each ore mining site could have different ore qualities and this quality must be specifically accounted for when computing the quality of material being processed.

The variables q^P_{ijkt} measure the number of trucks of type $k \in K$ queued at each site $j \in P$ from mining site $i \in M$ at time t. The following constraints are used to track the number of trucks queued at a processing site:

$$q^P_{ijkt} = q^P_{ijk,t-1} - y^P_{ijkt} + x_{ijk,t-\tau_{ij}} \qquad \forall k \in K, \forall i \in G, \forall j \in P, \forall t \in T^H. \tag{26}$$

Parameter	Description	Units
T	The set of time periods in the planning horizon ($t \in \{1, ..., T^{\max}\}$)	–
T^H	The set of time periods in the planning horizon augmented with earlier periods to account for trucks being loaded, unloaded or en route at $t = 0$ ($t \in \{-\max_{ij} \tau_{ij}, ..., T^{\max}\}$)	–
α_k	Size of truck type $k \in K$	Tons/truck
$\bar{p}_i$	Maximum number of trucks that can be processed per period at site $i \in N$ (may be fractional)	Trucks/period
Θ_i^M	Amount that should be extracted from mine $i \in M$ during the planning horizon	Tons
Θ_j^P	Amount that should be processed by processing site $j \in P$ in each period	Tons
I_{j0}	Amount in short-term stockpile $j \in P$ at period 0	Tons
C_j	Maximum capacity of short-term stockpile $j \in P$	Tons
τ_{ij}	Number of periods needed to travel from $i \in N$ to $j \in N$	Periods
Q_{ir}^M	Ore attribute $r \in R$ percentage at mine $i \in G$	(%)
Q_{jr}^P	Target ore attribute $r \in R$ percentage of the ore at processing site $j \in P$	(%)
Q_{jr0}^I	Ore attribute $r \in R$ percentage of the ore in short-term stockpile $j \in P$ at period 0	(%)
$\hat{x}_{ijkt}$	# of trucks of type k sent from $i \in N$ to $j \in N$ at period $t \in T^H \setminus T$	Trucks
$\hat{g}_{kj}$	# of trucks of type $k \in K$ available at $j \in N$ at $t = 0$	Trucks
$\hat{q}_{ik}^M$	# of trucks of type $k \in K$ waiting in a queue at mine $i \in M$ at period $t = 0$	Trucks
$\hat{q}_{ijk}^P$	# of trucks of type $k \in K$ waiting in a queue at $j \in P$ holding load from mine $i \in G$ at period $t = 0$	Trucks

Table 3: A summary of parameters used in the discrete-time MIP dispatching model

Trucks waiting in a queue cannot be dispatched until after they are unloaded. Thus, the constraints require that for each processing site and for each mine, the number of trucks in queue at time t, q^P_{ijkt}, is the number of trucks in queue at time $t-1$ ($q^P_{ijk,t-1}$) minus the number of trucks unloaded at time t (y^P_{ijkt}), plus the number of trucks arriving at time t ($\sum_{i\in M} x_{ijk,t-\tau_{ij}}$). The subscript $t-\tau_{ij}$ on the flow variables accounts for travel time; if a truck leaves location i at time t, it will not arrive at its destination j until time $t+\tau_{ij}$.

For the mining sites, we do not need a separate queue variable for each truck's origin. We use q^M_{ikt} to track the number of trucks queued at mining site i of type k in time period t. The queue balance constraints at the mining sites are then:

$$q^M_{ikt} = q^M_{ik,t-1} - y^M_{ikt} + \sum_{j\in N\setminus M} x_{jik,t-\tau_{ji}} \qquad \forall k\in K, \forall i\in M, \forall t\in T^H. \tag{27}$$

We initialize the model to fix the initial queue lengths to their current value.

$$q^P_{ijk,0} = \hat{q}^P_{ijkt} \qquad \forall i\in M, \forall j\in P, \forall k\in K \tag{28}$$

$$q^M_{ik,0} = \hat{q}^M_{ik} \qquad \forall i\in M, \forall k\in K. \tag{29}$$

We assume the unloading times at dump sites and long-term stockpiles are relatively small, so that queueing does not need to be modeled at these sites. If this were not true, the model could easily be modified to treat these sites the same as processing sites.

When a truck has finished being unloaded or loaded, it is ready to be dispatched to a new location. However, if there is no need for immediate dispatching, our model allows trucks to idle at a location until a later time. We use variables g_{ikt} to keep track of the number of trucks of type $k\in K$ at site $i\in N$ in period $t\in T$ that have completed their loading or unloading but have not yet been dispatched to a new location. The constraints to balance the number of available trucks at each site depend on the type of site:

$$g_{jkt} = g_{jk,t-1} - \sum_{i\in M} x_{jikt} + \sum_{i\in G} y^P_{ijkt} \qquad \forall k\in K, \forall j\in P, \forall t\in T^H \tag{30}$$

$$g_{jkt} = g_{jk,t-1} - \sum_{i\in M} x_{jikt} + \sum_{i\in B} x_{ijk,t-\tau_{ij}} \; \forall k\in K, \forall j\in D, \forall t\in T^H \tag{31}$$

$$g_{jkt} = g_{jk,t-1} - \sum_{i\in M} x_{jikt} + \sum_{i\in G} x_{ijk,t-\tau_{ij}} \; \forall k\in K, \forall j\in S, \forall t\in T^H \tag{32}$$

$$g_{ikt} = g_{ik,t-1} - \sum_{j\in P\cup S} x_{ijkt} + y^M_{ikt} \qquad \forall k\in K, \forall i\in G, \forall t\in T^H \tag{33}$$

$$g_{ikt} = g_{ik,t-1} - \sum_{j\in D} x_{ijkt} + y^M_{ikt} \qquad \forall k\in K, \forall i\in B, \forall t\in T^H. \tag{34}$$

At each processing site, the number of trucks available for dispatching at time t is the number of trucks available at time $t-1$ ($g_{jk,t-1}$), minus the number of trucks that are sent back to the mines at t ($\sum_{i\in M} x_{jikt}$), plus the number of trucks that are unloaded at time t ($\sum_{i\in G} y^P_{ijkt}$). Similar constraints exist for each of the different sites in the mine. We initialize the model with parameters indicating the current availability of trucks.

$$g_{ik,0} = \hat{g}_{ik} \qquad \forall i\in N, \forall k\in K. \tag{35}$$

We track the amount of ore in the short-term stockpile at each processing site using variables I_{jt} for the tons of ore at processing site $j \in P$ at the end of time period $t \in T$. We monitor the amount of ore by keeping track of the ore added each period and the amount of ore removed (processed) in each period, which should be the parameter Θ_j^P. We introduce variables γ_{jt} that measure by how much we violate the processing rate target at each site $j \in P$ and each time period $t \in T$. We model the amount of ore in the short-term stockpiles with the following set of constraints:

$$I_{jt} = I_{j,t-1} - (\Theta_j^P - \gamma_{jt}) + \sum_{k \in K, i \in G} \alpha_k y_{ijkt}^P \qquad \forall t \in T, \forall j \in P \tag{36}$$

$$I_{jt} \leq C_j \qquad \forall t \in T, j \in P \tag{37}$$

$$I_{jT^{\max}} \geq I_{j0} \qquad \forall j \in P \tag{38}$$

$$I_{jt} \geq 0.3 C_j \qquad \forall j \in P, t \geq \lfloor T^{\max}/2 \rfloor \tag{39}$$

Constraints (36) say the amount in the short-term stockpile at time t must equal the amount in the short-term stockpile at time $t-1$ minus the amount removed plus the amount added. The desired rate of removal of material from the short-term stockpile is the parameter Θ^P. Constraints (37) enforce the short-term stockpile capacity restrictions, and constraints (38) attempt to alleviate end-of-horizon effects by requiring the stockpile level at $T^{\max}$ to be at least as large as the initial short-term stockpile level. Recall that the processors must slow down if the inventory in its associated short-term stockpile is too low. Constraints (39) require that after the halfway point of the planning horizon, the ore in each short-term stockpile is at least the slow-down threshold, typically set to 30% of the stockpile capacity. Note that it might not be physically possible to meet these constraints in all cases, but we model the inventory and processing constraints in such a way that the model is always feasible. Specifically, we can always increase γ_{jt} enough to meet the inventory requirements in any period. Of course, if $\Theta_j^P - \gamma_{jt} < 0$ we claim to have negative production, which is not physically possible. However, because we minimize $\sum_{j \in P, t \in [0, T^{\max}]} \gamma_{jt}$, anytime it is feasible to meet the processing rate targets and the inventory constraints, these variables will be 0. Thus, we model the processors slowing down as not meeting the target processing rate.

We fix each flow variable x_{ijkt} for $t \in T^H \setminus T$ to the number of trucks en route at that same time, $\hat{x}_{ijkt}$,

$$x_{ijkt} = \hat{x}_{ijkt} \qquad \forall i \in N, \forall j \in N, \forall k \in K, \forall t \in T^H \setminus T.$$

We also require that all decision variables are non-negative, and that truck availability, loading/unloading, and flow variables are integral:

$$h_{it}^U, \gamma_{it}, I_{jt}, q_{ikt}^M, q_{ijkt}^P \geq 0 \quad \forall i, j \in N, \forall k \in K, \forall t \in T, \forall r \in R \tag{40}$$

$$x_{ijkt}, g_{kjt}, y_{ijkt}^P, y_{ikt}^M \in \mathbb{Z}_+ \qquad \forall i, j \in N, \forall k \in K, \forall t \in T. \tag{41}$$

One of the difficulties of modeling the dispatching problem is incorporating the limited loading capacity at the mining sites and unloading capacity at the processing sites. We approximate the effect of the limited capacity by rounding up the maximum number of trucks that can be loaded (unloaded) at each site to the nearest integer:

$$\sum_{i\in G}\sum_{k\in K} y^P_{ijkt} \leq \lceil \bar{p}_j \rceil \quad \forall j \in P, \forall t \in T \tag{42}$$

$$\sum_{i\in M}\sum_{k\in K} x_{jikt} \leq \lceil \bar{p}_j \rceil \ \forall j \in D, \forall t \in T \tag{43}$$

$$\sum_{i\in M}\sum_{k\in K} x_{jikt} \leq \lceil \bar{p}_j \rceil \ \ \forall j \in S, \forall t \in T \tag{44}$$

$$\sum_{k\in K} y^M_{ikt} \leq \lceil \bar{p}_i \rceil \qquad \forall i \in M, \forall t \in T. \tag{45}$$

We note that we could choose to round down instead, but this underestimates the capacity and if $\bar{p}_i < 1$ for some site $i \in N$ no trucks could ever be loaded or unloaded at that site. However, rounding up to the nearest integer in each period overestimates the rate at which trucks can be loaded or unloaded. We mitigate this overestimation by also implementing loading capacity constraints over windows of time of length d_i, where $d_i \in \mathbb{Z}$ is chosen to minimize the rounding error of $\lceil d_i\bar{p}_i \rceil$ for each site $i \in N$. In particular, we choose d_i as follows:

$$d_i \in \arg\min \left\{ \frac{\lceil d\bar{p}_i \rceil - d\bar{p}_i}{d\bar{p}_i} : d \in \{1, \ldots, d^{\max}\} \right\}$$

for a fixed value $d^{\max}$. In our test instances, we choose $d^{\max} = 10$. Keeping the window length d_i short ensures that the overlapping window constraints we describe next for modeling loading and unloading capacity are not overly dense.

Once we have a value of d_j for each site, we build a family of constraints that approximate capacity at different sites in the mine:

$$\sum_{s=t}^{t+d_j-1} \sum_{i\in G}\sum_{k\in K} y^P_{ijks} \leq \lceil d_j\bar{p}_j \rceil \quad \forall j \in P, \forall t \in \{1,..,T^{\max} - d_j + 1 : t > d_j\} \tag{46}$$

$$\sum_{s=t}^{t+d_j-1} \sum_{i\in M}\sum_{k\in K} x_{jiks} \leq \lceil d_j\bar{p}_j \rceil \forall j \in D, \forall t \in \{1,...,T^{\max} - d_j + 1 : t > d_j\} \tag{47}$$

$$\sum_{s=t}^{t+d_j-1} \sum_{i\in M}\sum_{k\in K} x_{jiks} \leq \lceil d_j\bar{p}_j \rceil \ \forall j \in S, \forall t \in \{1,...,T^{\max} - d_j + 1 : t > d_j\} \tag{48}$$

$$\sum_{s=t}^{t+d_i-1} \sum_{k\in K} y^M_{iks} \leq \lceil d_i\bar{p}_i \rceil \qquad \forall i \in M, \forall t \in \{1,...,T^{\max} - d_i + 1 : t > d_i\}. \tag{49}$$

For $t = d_i, \cdots, T^{\max} - d_i + 1$, we require that the total number of trucks processed at site $i \in N$ in the time interval $[t, t + d_i - 1]$ is no more than $d_i\bar{p}_i$, rounded up to the nearest integer. Some slight modifications need to be made to these constraints at the beginning of the planning horizon:

$$\sum_{s=1}^{t}\sum_{i\in G}\sum_{k\in K} y^P_{ijks} \le \lceil t\bar{p}_j \rceil \quad \forall j \in P, \forall t \in T : t \le d_j \tag{50}$$

$$\sum_{s=1}^{t}\sum_{i\in M}\sum_{k\in K} x_{jiks} \le \lceil t\bar{p}_j \rceil \; \forall j \in D, \forall t \in T : t \le d_j \tag{51}$$

$$\sum_{s=1}^{t}\sum_{i\in M}\sum_{k\in K} x_{jiks} \le \lceil t\bar{p}_j \rceil \quad \forall j \in S, \forall t \in T : t \le d_j \tag{52}$$

$$\sum_{s=1}^{t}\sum_{k\in K} y^M_{iks} \le \lceil t\bar{p}_i \rceil \qquad \forall i \in M, \forall t \in T : t \le d_i. \tag{53}$$

For $t = 1, \ldots, d_i$, we require that the total number of trucks processed in the time interval $[1, t]$ is no more than $t\bar{p}_i$, rounded up to the nearest integer. By enforcing all constraints (42) - (53), the model effectively approximates and constrains loading and unloading rate limits at all sites. We show the impact of including these constraints in a computational study in Section 5.3.2.

To simplify notation, let X be the set of solutions which satisfy all constraints:

$$X^M := \{v = (x,\ y^P,\ y^M\ q^M,\ q^P,\ g,\ \gamma,\ I) : (26) - (53)\}.$$

Then a solution $v \in X^M$ is a feasible solution to the dispatching problem. The decision variables of the model are summarized in Table 4.

Variable	Description	Units
x_{ijkt}	# of trucks of type k that leave $i \in N$ traveling to $j \in N$ at time t	Trucks
y^P_{ijkt}	# of type $k \in K$ trucks holding load from $i \in G$ unloaded at $j \in P$ at time $t \in T$	Trucks
y^M_{ikt}	# of type $k \in K$ trucks loaded at $i \in M$ at time $t \in T$	Trucks
q^P_{ijkt}	# of trucks of type $k \in K$ waiting in a queue at $j \in P$ holding load from $i \in G$ at time $t \in T$	Trucks
q^M_{ikt}	# of trucks of type $k \in K$ waiting in a queue at mine $i \in M$ at time $t \in T$	Trucks
g_{kjt}	# of trucks of type $k \in K$ available at $j \in N$ at $t \in T$	Trucks
I_{jt}	Amount in short-term stockpile $j \in P$ at $t \in T$	Tons
γ_{jt}	Lower violation of the amount extracted from short-term stockpile (and processed) at $j \in P$ at $t \in T$	Tons
w^U_i	Upper violation of the extraction targets for $i \in M$	Tons
w^L_i	Lower violation of the extraction targets for $i \in M$	Tons
$b^U_{jr\ell}$	Upper violation of the ore attribute percentage $r \in R$ at $j \in P$ during W_ℓ	Tons
$b^L_{jr\ell}$	Lower violation of the ore attribute percentage $r \in R$ at $j \in P$ during W_ℓ	Tons

Table 4: A summary of all decision variables used in MIP model

3.3 Objective

Like in the nonlinear average flow-rate model of Section 2.1, we use a three-phase hierarchical objective structure to model the relative importance of each objective. This hierarchy prioritizes meeting processing rates, followed by mining targets, and finally ore quality targets if possible.

We first minimize the deviation from the processing rate target. Specifically, we solve the problem:

$$\min_{v \in X^M} \sum_{j \in P} \sum_{t \in T} \gamma_{jt}. \tag{P1}$$

Ideally, we achieve an objective value of 0 (no processing rate target violations). In our application, not meeting the processing targets is considered an emergency state, so in this situation the volume of ore processed matters much more than any other performance metric. We record the value of the processing target violation variables ($\hat{\gamma}$) and proceed to the next phase.

In the second phase, we minimize the deviation from the mining extraction targets. We know there exists a feasible solution to P1 with $\gamma \leq \hat{\gamma}$, so we introduce upper bounds on these variables. We also need to track the progress in meeting mining targets. We introduce variables w_i^U and w_i^L to measure the upper and lower violation of the mining extraction targets over the full planning horizon for each mine $i \in M$. We measure these violations with the following constraints:

$$w_i^U \geq \sum_{t \in T, k \in K} \alpha_k y_{ikt}^M - \Theta_i^M \quad \forall i \in M \tag{54}$$

$$w_i^L \geq \Theta_i^M - \sum_{t \in T, k \in K} \alpha_k y_{ikt}^M \quad \forall i \in M \tag{55}$$

$$w_i^U, w_i^L \geq 0 \qquad \forall i \in M. \tag{56}$$

Constraints (54) - (56) measure the difference between the extraction amount ($\sum_{t \in T, k \in K} \alpha_k y_{ik}^M$) from a mine site $i \in M$ during the time horizon and the target extraction (Θ_i^M). The difference is assigned to either the w^L or the w^U variables, depending on whether the extraction is below or above the target (respectively). We are concerned with the size of the greatest target violation rather than the sum total of all mines' target violations, so we introduce another variable z which is at least the maximum violation and minimize z in the objective function.

The full model for the second phase is then

$$\begin{aligned}
\min\ & z && \qquad \text{(P2)}\\
\text{s.t. } & (x,\ y^P,\ y^M,\ q^M,\ q^P,\ g,\ \gamma,\ I) \in X^M && \\
& (y^M,\ w^U,\ w^L) && \text{satisfy } (54) - (56)\\
& \gamma_{jt} \leq \hat{\gamma}_{jt} && \forall j \in P, t \in T\\
& z \geq w_i^U,\ z \geq w_i^L,\ z \geq 0 && \forall i \in M.
\end{aligned}$$

If the processing target violation variables γ were nonzero in after the first phase, meaning that the operations are in an emergency state, we terminate with the best solution to (P2). Otherwise, we record the actual mining target violation values for each site ($\hat{w}^U, \hat{w}^L$) after solving P2 and progress to the final phase.

In the final phase, we bound the mining target violation variables by the value $\hat{w}^U$ (or $\hat{w}^L$) plus a fraction (typically 1%) of the target extraction for each site. This choice allows some flexibility around the P2 solution values. The objective in the final phase is to minimize the violation of the ore quality targets. The bounds on mining targets are as follows:

$$w_i^U \leq \hat{w}_i^U + 0.01\Theta_i^M \quad \forall i \in M \tag{57}$$

$$w_i^L \leq \hat{w}_i^L + 0.01\Theta_i^M \quad \forall i \in M. \tag{58}$$

In the final phase, we measure the deviation from the quality targets. The goal is to monitor the quality of the ore in the short-term stockpile over the horizon, but since the quality changes over time, the standard blending formula results in nonconvex constraints. Specifically, the attribute calculations in each period are

$$\hat{C}_{jrt} = \hat{C}_{jr,t-1} + \sum_{i \in G}(Q_{ir}^M \sum_{k \in K} \alpha_k y_{ijkt}^P) - \frac{\hat{C}_{jr,t-1}}{I_{j,t-1}}(\Theta_j^P - \gamma_{jt}), \tag{59}$$

where $\hat{C}_{jrt}$ is a new decision variable corresponding to the tons of attribute $r \in R$ unloaded at processing site $j \in P$ at time $t \in T$. Since $\hat{C}_{jrt}$, I_{jt}, and γ_{jt} are all decision variables, the final term of this equation is the product of variables divided by another variable, making the equation nonconvex. Note that even this equation is an approximation, as it assumes perfect mixing of material once it enters the short-term stockpile.

To avoid introducing nonconvexity into our model, we use a family of overlapping constraints, each spanning a window of time of fixed length (a parameter H) and measure the violation of the ore quality *added* to the short-term stockpile within that window, rather than the violation of ore quality *in* the short-term stockpile in a given period. Each subsequent constraint is written for a window of time that is F periods in the future. This structure is similar to the L-average model proposed in (Moreno et al, 2017). We set the window length H so that is possible to receive truckloads from mines with different qualities within the window. This setting of H ensures there is a chance that the total ore added to the short-term stockpile in each window can meet the ore quality target. Let W be the largest integer such that $(W-1)F + H \leq T^{\max}$.

It is common for there to be an envelope of acceptable deviation from the quality targets that would be penalized very little or not at all. There are numerous options for modeling this piecewise-linear function. In this model, we introduce small bounds on the quality violation variables within which quality target violation is not penalized. We use a parameter ϵ to allow a window of quality target deviation before b^L or b^U become positive. In our test instances, we set ϵ to 0.005. The constraints to track and measure ore quality deviation are as follows:

$$b^U_{jr\ell} \geq \sum_{\substack{\ell F+1 \leq t \leq \min\{\ell F+H, T^{\max}\} \\ k \in K \\ i \in G}} \Big(Q^M_{ir} - Q^P_{jr} - \epsilon\Big)\alpha_k y^P_{ijkt} \qquad \forall j \in P, r \in R, \ell \in [1, W] \tag{60}$$

$$b^L_{jr\ell} \geq \sum_{\substack{\ell F+1 \leq t \leq \min\{\ell F+H, T^{\max}\} \\ k \in K \\ i \in G}} \Big(Q^P_{jr} - \epsilon - Q^M_{ir}\Big)\alpha_k y^P_{ijkt} \qquad \forall j \in P, r \in R, \ell \in [1, W] \tag{61}$$

$$b^U_{jr,0} \geq \sum_{\substack{1 \leq t \leq H \\ k \in K \\ i \in G}} \Big(Q^M_{ir} - Q^P_{jr} - \epsilon\Big)\alpha_k y^P_{ijkt} + \Big(Q^I_{jr0} - Q^P_{jr} - \epsilon\Big) I_{j0} \qquad \forall j \in P, r \in R \tag{62}$$

$$b^L_{jr,0} \geq \sum_{\substack{1 \leq t \leq H \\ k \in K \\ i \in G}} \Big(Q^P_{jr} - \epsilon - Q^M_{ir}\Big)\alpha_k y^P_{ijkt} + \Big(Q^P_{jr} - \epsilon - Q^I_{jr0}\Big) I_{j0} \qquad \forall j \in P, r \in R \tag{63}$$

$$b^L_{jr\ell}, b^U_{jr\ell} \geq 0 \qquad \forall j \in P, r \in R, \ell \in [0, W]. \tag{64}$$

Constraints (60) - (64) measure the difference between the target ore quality and the actual ore quality that is received by the short-term stockpile over W different time horizons. The constraints assign values to the variables b^L and b^U based on lower and upper violation of the quality targets, respectively. In constraints (60) and (61), the value is calculated as the absolute value of the difference between the quality of the $\sum_{\ell F+1 \leq t \leq \ell F+H} \sum_{k \in K, i \in G} \alpha_k y^P_{ijkt}$ tons of ore added to short-term stockpile j during the window $[\ell F + 1, \min\{\ell F + H, T^{\max}\}]$ for $\ell \in [1, \cdots, W-1]$ and the target quality of that same volume of ore. Small modifications must be made for the first ($\ell = 0$) constraints for each $j \in J$ (constraints (62) and (63)). In this case, we add the absolute value of the difference between the target quality of the ore in the short-term stockpile ($Q^P_{jr} I_{j0}$) and the actual quality of the ore in the short-term stockpile at the beginning of the planning horizon ($Q^I_{jr0} I_{j0}$). In our implementation, we found that values of $H = 20$ periods and $F = 4$ periods were a good choice of these parameters. In the final phase, we fix the processing rate violations to be zero and minimize the deviation from ore quality targets:

$$\begin{aligned} \min \; & \sum_{j \in P} \sum_{r \in R} \sum_{\ell \in W} (b^L_{jr\ell} + b^U_{jr\ell}) \\ \text{s.t. } & (x,\ y^P,\ y^M,\ q^M,\ q^P,\ g,\ \gamma,\ I) \in X^M \\ & \gamma_{jt} \leq 0 && \forall j \in P, t \in T \\ & (w^U, w^L) && \text{satisfy } (57) - (58) \\ & (b^U, b^L, y^P) && \text{satisfy } (60) - (64). \end{aligned} \tag{P3}$$

3.4 A MIP-Based Dispatching Policy

We next describe how the solution of MIP model can be used to derive a dispatching policy. At regular intervals, we solve the MIP model using data describing the current state of the system. A solution to the dispatching problem includes the flow variable values, which contain all the dispatching decisions for the planning horizon specified in the model; specifically, x_{ijkt} indicates how many trucks of type $k \in K$ from location $i \in N$ are sent to $j \in N$ in time period t. For each starting location $i \in N$, and each truck type $k \in K$, we create a list of destinations to which the MIP solution proposes to send a truck of type k at each time period t in the planning horizon. Assuming the destinations are indexed by $\{1, 2, \ldots, |N|\}$, the dispatch list takes the form

$$\mathcal{L}'_{ikt} = (\underbrace{1, \ldots, 1}_{x_{i1kt}}, \underbrace{2, \ldots, 2}_{x_{i2kt}}, \ldots, \underbrace{|N|, \ldots, |N|}_{x_{i|N|kt}}).$$

Thus, each list contains exactly x_{ijkt} copies of each destination $j \in N$. Typically the list is very short, since in a given time period, very few dispatch decisions for the same location and truck type are made. This list is shuffled randomly. Using these shuffled lists, we create for each location $i \in N$ and truck type $k \in K$ an ordered list of the destinations to which trucks should be sent from i.

$$\mathcal{L}_{ik} = (\mathcal{L}'_{ik,0}, \mathcal{L}'_{ik,1}, \ldots, \mathcal{L}'_{ik,T^{\max}}).$$

We dispatch trucks of type $k \in K$ from location $i \in N$ according to this list. Specifically, an iterator at each location $i \in N$ for each truck type $k \in K$ increments every time a dispatching decision is made at that location and indicates which list index to reference for the next dispatching decision at i with truck type k.

A key parameter in the use of this policy is how often the MIP model should be re-solved. Certainly, any time a major event occurs in the system, such as an equipment failure, the data should be updated and the model should be solved again. In normal operation, however, the correct timing of the re-solves is less clear. One option is to update the data with the current system information and re-solve the model after a fixed amount of time has passed. In Section 5.2 we discuss this approach in more detail.

3.5 Relaxation Induced Neighborhood Search Heuristic

For the three-phase hierarchical model to be used to make dispatching decisions in real time, we must be able to solve instances of P1, P2, and P3 quickly. In practice, we have found P3 to be difficult to solve to optimality because for a planning horizon of only 20 minutes, the model can have over 10,000 integer variables and the ore quality constraints have a structure that complicates the solution of the model.

We observe that P1 solves to optimality quickly enough in our test instances. P2 often solves quickly, but occasionally takes significant time to prove a solution is optimal. Thus, introducing a time limit (such as 20 seconds) on P2 is one way to ensure a bound on the mining target violations is obtained quickly. Similarly, a simple heuristic for solving P3 is to fix a time limit and record the best solution found. However, the solution quality could be bad, depending on the problem size and the choice of time limit. Thus, we implement a solution methodology that can find a good feasible solution to P3 quickly based on the solution to the LP relaxation of P3.

We propose a heuristic based on the relaxation induced neighborhood search (RINS) algorithm described in (Danna et al, 2005). In the RINS algorithm, a neighborhood that is likely to contain a high-quality MIP solution is constructed based on a solution to the LP relaxation of the MIP model. Next, a MIP model is solved to find the best solution within that neighborhood. In our implementation, we create a neighborhood by fixing a subset of the integer variables in P3 to 0 to reduce the size of the variable space, creating a problem that is more easily solved, and explore the neighborhood by solving the restricted P3 MIP. Our implementation of the RINS heuristic leverages the fact that many of the integer variables take value 0 in both the optimal MIP solution and the solution to the LP relaxation of P3. Motivated by this observation, we first solve the LP relaxation of P3, then we fix all integer variables that have value 0 in both the solution of the LP-relaxation to P3 and the solution to P2. In this way, the solution to P2 remains feasible to this restricted form of P3. Finally, we re-introduce all the integrality constraints and solve the restricted P3. This problem typically solves very quickly in our experiments. In the next section, we will demonstrate the effectiveness of this problem-specific RINS heuristic via a set of computational experiments.

4 Computational Experiments: Heuristic Solution Quality

In this section, we report on an experiment that demonstrates the performance of the RINS heuristic from Section 3.5, and we use the solutions to measure and assess the quality of our MIP model's approximation of the nonconvex blending condition (59).

In our method, we allow P1 to solve to optimality and enforce a 20-second time limit on P2. If a solution is not found in the 20-second time limit, we allow Gurobi to continue searching for a feasible solution for up to 90 seconds. If t' is the number of seconds required to solve P2, then we set a time limit of $t^{\max} = 90 - t'$ seconds for solving P3. Thus, a time limit of 90 seconds is given to solving both P2 and P3. In all our experience, the problem P1 solves in a matter of seconds. We evaluate two heuristics for solving P3:

1. Use Gurobi to directly solve P3 with a time limit of $t^{\max}$ with the parameter `MIPFocus=1` to emphasize finding feasible solutions quickly;
2. Use the RINS heuristic described in Section 3.5 to solve P3 with a time limit of $t^{\max}$ and with `MIPFocus=1`.

The models and solution approaches are implemented in Python 2.7 and solved with Gurobi v7.0.2. All computational results in the paper were collected on an Intel Xeon X5650 server with 12 cores at 2.66Ghz and 128GB of RAM using 24 threads. With the exception of the `MIPFocus` parameter, all Gurobi parameters are left in their default state.

The mine used in this experiment has five ore sites, five waste sites, two processing sites, two stockpile sites, five dump sites, one attribute, and two truck types ($|G| = 5$, $|B| = 5$, $|P| = 2$, $|S| = 2$, $|D| = 5$, $|R| = 1$, and $|K| = 2$). We test planning horizon lengths of 20 minutes, 30 minutes, and 40 minutes using 30-second time periods, so $T^{\max} \in \{40, 60, 80\}$. In these instances, each of the 40 available trucks is randomly assigned one of the two possible sizes. All other parameters used to build the model are randomly generated according to the instance-generation description given in Appendix A. We generate 10 separate instances of the random input parameters for each planning horizon length, for a total of 30 distinct instances. For the ore quality constraints (Equations (62)-(64)), we choose the periods covered by each constraint (H) to be 20 periods (10 minutes) and the distance between successive constraints (F) to be 4 periods (2 minutes). For more information on these choices, the interested reader is directed to the thesis (Smith, 2019), which contains a wealth of experiments aimed at tuning model and algorithm parameters.

Total violation of quality targets (P3 objective)												
	20-min horizon				30-min horizon				40-min horizon			
	P2	OPT	90-SEC	RINS	P2	OPT	90-SEC	RINS	P2	OPT	90-SEC	RINS
1	265	0.7	1.1	7.6	559	5.0	55.5	14.7	754	25.9	-	17.4
2	429	5.4	5.4	19.5	968	5.0	7.0	2.6	2065	37.0	49.7	5.3
3	240	0.6	0.8	100.0	359	1.0	-	1.6	242	1.7	168.8	4.3
4	302	**155.1**	155.1	168.7	365	144.6	148.0	154.6	554	143.0	155.2	147.6
5	235	0.2	0.2	2.9	157	0.0	0.0	7.6	771	9.5	-	4.7
6	302	0.5	0.8	1.3	763	0.5	4.3	3.0	1164	1.5	2241.8	9.0
7	107	1.7	1.7	3.1	278	1.8	12.4	3.9	632	3.6	-	6.4
8	659	31.1	31.1	40.6	1525	31.3	35.0	47.0	2848	37.3	57.4	44.6
9	135	**42.7**	42.7	42.7	249	**28.3**	28.3	28.3	565	**28.0**	-	28.3
10	325	**86.4**	86.4	86.5	1684	80.3	-	72.0	2406	86.1	-	86.9

Table 5: Summary of the total quality target violation objective using different solution methods compared to not solving P3. Each method is tested on 3 different planning horizons. Numbers in **bold** font indicate which instances were solved to optimality in an hour.

For each planning horizon length, we observe the total quality target violation in the solution to P2, in the solution to P3 when solved as close to optimality as possible in one hour (OPT), in the solution to P3 using a time limit of $t^{\max}$ (90-SEC), and in the solution to P3 using a time limit of $t^{\max}$ with the RINS heuristic (RINS). The results of this experiment can be found in Table 5.

From the results of the experiment, we observe that the ore quality objective improves significantly when solving P3 rather than stopping after solving P2. In many cases, the quality of the solution found by Gurobi in 90 seconds and the RINS heuristic in 90 seconds are quite comparable and not substantially worse than the solution found after running Gurobi for one hour. However, when using the 90-SEC heuristic, no feasible solution was found in two of the 30-minute horizon instances and five of the 40-minute horizon instances. We were able to find a solution within 90 seconds for all instances using the RINS heuristic. Thus, we use the RINS heuristic with a time limit of 90 seconds for all remaining computations.

Recall that the quality violation measurements in P3 are only an approximation of the true ore quality because of the nonconvex nature of the blending model. Because this approximation does not accurately calculate the true quality violations, after solving P3 we use Equation (59) to calculate the *true* ore quality in the given solution. We demonstrate the solution quality of each instance by reporting for each instance the percentage of the time horizon in which the true ore quality at the processing sites was within 1% of the target. Recall that we set the window value $\epsilon = 0.5\%$ in equations (60)— (63), so this is twice the window size in which the model attempts to keep the quality. The results of this calculation are summarized in Table 6.

The results of the experiment indicate that solutions from the RINS heuristic do a good job of keeping the ore quality close to the target at the processing sites. The fifth instances is paradoxical, since for the solutions to the 30 and 40 minute horizon problems, the solution without considering ore-quality has a larger fraction of time in which the ore quality in within a 1% band of the target at the processors. In this instance, the solution of the P2 problem had significant violations at the end of the horizon, but the violation was small a large fraction of the time.

The results presented in this section allow us to draw some conclusions about our model. As

Total percentage of time ore-quality within 1% of target												
	20-min horizon				30-min horizon				40-min horizon			
	P2	OPT	90-SEC	RINS	P2	OPT	90-SEC	RINS	P2	OPT	90-SEC	RINS
1	70.0	97.5	96.2	95.0	96.2	100	100	100	96.2	100	-	100
2	100	100	100	100	100	100	100	100	100	100	100	100
3	100	100	100	100	100	100	-	100	98.8	100	98.8	100
4	100	100	100	100	100	100	100	100	100	100	100	100
5	73.8	76.2	85.0	85.0	96.2	97.5	93.8	88.8	86.2	87.5	-	82.5
6	58.8	77.5	90.0	100	65.0	93.8	70	92.5	72.5	91.2	76.2	93.8
7	75.0	93.8	88.8	96.2	98.8	100	88.8	97.5	100	92.5	-	100
8	57.5	100	95.0	100	46.2	100	100	100	30.0	100	100	100
9	100	100	100	100	100	100	100	100	100	100	-	100
10	100	100	97.5	100	78.8	100	-	100	100	100	-	100

Table 6: Total percentage of time the ore-quality was within 1% of the ore-quality targets at the processing sites.

expected, the P3 solution quality deteriorates somewhat when solved with the RINS heuristic, but overall the solutions are comparable to solving the instance as close to optimality as possible with a one-hour time limit. More significantly, these solutions are obtained in 90 seconds or less, making this model efficiently implementable. For horizons of 30 or 40 minutes, the solution obtained from RINS within 90 seconds is better than the solution obtained by simply enforcing a 90-second on the emphasize feasibility method, since for many instances a solution could not even be found in that time. Furthermore, we conclude that the inclusion of P3 is important for meeting the ore quality objective, in spite of the additional computational difficulty over P1 and P2. Results also demonstrate that the moving and overlapping window approximation to the nonconvex blending constraints does a good job in ensuring tight adherence to ore quality targets at the processors.

5 Computational Experiments: Model and Dispatch Policy Evaluation

We conduct a set of computational experiments to better understand how the models and policies described in Sections 2 and 3 perform. Key to our evaluation is a discrete-event simulation of an open-pit mining operation. In this section, we describe the simulation model used, we describe key performance metrics, we report on the use of the simulation to assess the importance of novel features in our dispatching models, and finally we compare the performance of the average flow-rate-based and MIP-based dispatching policies when varying the number of trucks, the travel times, and the ore qualities.

5.1 Simulation Model

Trucks are loaded at mining sites $i \in M$, and the time to load a truck is a random variable $\boldsymbol{p}_i$. Loading at each mine is modeled as a single-server system, and trucks waiting to be loaded queue in a first-in first-out (FIFO) basis. The amount of material that is loaded into a truck of type $k \in K$ is represented by a random variable $\boldsymbol{\alpha_k}$ tons that follows a Jonson's S_U distribution with parameters $(\lambda_k, \gamma_k, \delta_k, \xi_k)$. Similarly, the time to unload a truck at a processing site, long-term stockpile, or dump site i is is a random variable $\boldsymbol{p}_i$, and each of these sites is also treated as a single-server system with a FIFO queue. The loading and unloading rates at each location $i \in N$ follow a Jonson's S_U distribution with parameters $(\lambda_i, \gamma_i, \delta_i, \xi_i)$. When a truck finishes loading or unloading at a site, its next destination is determined by calling a dispatching function, which takes as an input the state of the system and returns the next destination of the truck. In our simulation, the dispatching will be performed either according to the average-flow rate-based target-matching heuristic described in Section 2.2 or the MIP-based dispatching policy of Section 3.4. The time it takes a truck to travel between two locations $i \in N$, $j \in N$ is given by a random variable $\boldsymbol{\tau}_{ij}$ that follows a Jonson's S_U distribution with parameters $(\lambda_{ij}, \gamma_{ij}, \delta_{ij}, \xi_{ij})$. The Jonson S_U distribution was used in our simulation since it is a flexible distribution that allowed us to accurately model distributions coming from empirical data from a corporate partner in the project.

Each processing site $j \in P$ maintains a short-term stockpile where trucks dump material taken from the mines. Ore is removed from the stockpile for processing. We assume that the material added to the stockpile is instantaneously perfectly blended with material in the stockpile at the time the unloading is finished. Each processing site $j \in P$ has a processing rate Θ_j^P. We assume that Θ_j^P is a constant rate of depletion, in tons/minute, unless the total tons of ore in the short-term stockpile plus any truckloads currently en route to the processor falls below 30% of the capacity of he short-term stockpile. If this threshold is reached, the processing rate slows to $0.6\Theta_j^P$. If the short-term stockpile is emptied, the processor must stop. From an operational standpoint, slowing down or stopping the processing of material are to be avoided. Each short-term stockpile j has an initial amount of ore I_{j0} with an initial quality of Q_{jr0}^I for each attribute $r \in R$ and a maximum capacity of C_j tons for $j \in P$. If a truck's load would exceed the stockpile capacity, it must wait at that site until there is room in the short-term stockpile for the load. Each mining site $i \in M$ has an extraction target during that time of Θ_i^M.

Finally, we are given ore quality targets (in percentage of attribute $r \in R$) Q_{jr}^P at all processing sites $j \in P$ and for all attributes $r \in R$. Each truckload removed from an ore mine $i \in G$ has associated percentages of ore attributes represented by a random variable, $\boldsymbol{Q_{ir}^M}$ for all $r \in R$. The ore quality between subsequent truckloads at a site is correlated, since product extracted for consecutive truckloads is likely to be taken from neighboring locations at the mine. To model this correlation, we model the mean value q_{ir} of $\boldsymbol{Q_{ir}^M}$ as a piecewise linear function of the tons removed from the mining site. Let Θ_i^M be the target tons extracted during the shift at mine $i \in G$, x_i be the tons extracted so far from site $i \in G$, and M_{ir} be a given average of attribute $r \in R$ at site $i \in G$. The expected value of $\boldsymbol{Q_{ir}^M}$ is of the form

$$q_{ir}(x_i) = \begin{cases} \frac{-4M_{ir}}{\Theta_i^M} x_i + 2M_{ir} & \text{if } x_i \in [0, 0.5\Theta_i^M] \\ \frac{4M_{ir}}{\Theta_i^M} x - 2M_{ir} & \text{if } x_i \in (0.5\Theta_i^M, \Theta_i^M]. \end{cases} \tag{65}$$

Note that the average value of $q_{ir}(x)$ in the range $[0, \Theta_i^M]$ is M_{ir} and that the ore quality decreases during the first half of the extraction, and increases in the second half. For each mine $i \in G$, with probability $\frac{1}{2}$ we reflect the function $q_{ir}(x)$ across the line $q_{ir}(x) = M_{ir}$, so that some mines' quality starts by increasing and then decreases. Each time ore is extracted, we use (65) to find the current expected value of the quality $q_{ir}(x)$. Then the true quality of the ore taken from the mine, $\boldsymbol{Q_{ir}^M}$, follows a Uniform$(0.9q_{ir}(x),\ 1.1q_{ir}(x))$ distribution.

All random variables are assumed to be independent of each other. The details of the distributional parameters used for the Jonson S_U distributions for the load sizes, loading/unloading times, and travel times are available on request from the authors. The simulation is implemented in Python 2.7 using SimPy 3.0.

5.2 Key Performance Indicators and Dispatch Policy Tuning

When evaluating the performance of a dispatching policy, using the simulation model, we estimate six key metrics:

1. The percentage of simulation time in which the processing sites need to be shut down;
2. The percentage of simulation time in which the processing sites need to be slowed down to 60% of their target rate;
3. The maximum percentage by which we exceed the target extraction amount at any site;
4. The maximum percentage by which we fall short of the target extraction amount at any site;
5. The percentage of total ore processed at a site that is within 0.5% of the target quality for the site; and
6. The percentage of total ore processed at a site that is within 1% of the target quality for the site.

Keeping the processors running at full speed is critically important to mining operations, so the first two metrics are most important.

Space considerations do not allow us to present the results of experiments used to tune dispatching-policy parameters. The interested reader is directed to the Ph.D. thesis of Smith (Smith, 2019) for a full computational study. In this paper, we merely report a summary of the results of the study and describe good choices for parameter values. A key parameter for the Two-Phase dispatching policy of Section 2 is the interval of (simulation) time between which the nonlinear average flow-rate model is solved to set target flow rates for the target-matching heuristic described in Section 2.2. Potential values tested for the resolve-frequency were 5, 10, 15, 30, 60, 120, and 240 minutes. The computational study revealed that a resolve frequency of 60 minutes for the target-matching policy was a good choice. In fact, in the simulation study, the 60 minute resolve interval was the only value of the parameter tested that did not lead to instances where processing sites needed to be shut or slowed down in the simulation.

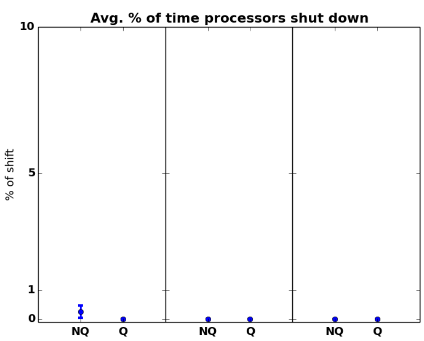
For the Discrete-Time MIP Dispatching Policy described in Section 3, there are two key parameters. The first is the re-solve interval—how much simulation time passes between successive re-solves of the discrete-time MIP dispatching model. In the computational study, values tested were 2, 5, 10, and 15 minutes. The second key parameter for the MIP model is the planning horizon. Because the MIP model is a time-discretized model with increments of 30 seconds, increasing horizon length corresponds to increasing model size (and difficulty). The values tested were 25, 30, and 35 minutes, and these values were tested in combination with the potential re-solve intervals. The result of the study was that a 25 minute planning horizon with a 5 minute re-solve frequency gave good performance for the MIP-based dispatching policy, and these are the values of the parameters we use in all remaining simulation studies.

5.3 Evaluation of Model Components

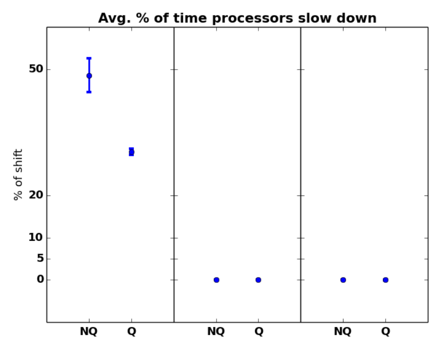
Using the simulation, we evaluate the impact of novel features of the nonlinear average flow-rate model and the discrete-time MIP dispatching model. Specifically, we test the effect of the queueing constraints (16)-(19) in the nonlinear flow-rate model and the moving window truck capacity constraints (46)-(53) and ore quality constraints (60)-(63) in the MIP dispatching model. We evaluate each policy on an instance with five good mines, four bad mines, two processing sites, two stockpile locations, two dump sites, one attribute, and one truck type: ($|G| = 5$, $|B| = 4$, $|P| = 2$, $|S| = 2$, $|D| = 2$, $|R| = 1$, $|K| = 1$). In our tests, we consider cases where the mine has 15, 25, or 35 trucks. The simulation is run over a 10 hours steady-state period of mining operations, and the target processing volume over the 10-hour simulation is 200,000 tons. All other parameters used to build an instance of the open-pit mine can be found in Appendix A. In all cases, we run 30 replications of the simulation to estimate the mean value of the six key metrics, and we will plot confidence intervals for these estimates in three panels, representing the evaluation of the instance for 15, 25, and 35, trucks, respectively.

5.3.1 Queuing constraints average flow-rate matching policy

Our first experiment demonstrates the utility of the queueing constraints in the nonlinear average flow rate model for the two-phase policy of Section 2.1. We run the simulation without consideration of queueing (NQ), and then we run the simulation by using rotated second-order cone constraints (16)-(19)) to approximate the effects of queueing (Q). Figure 2 displays the the shutdown and slowdown metrics for this study.



(a) 95% confidence intervals on fraction of time processors shut down for 30 replications.



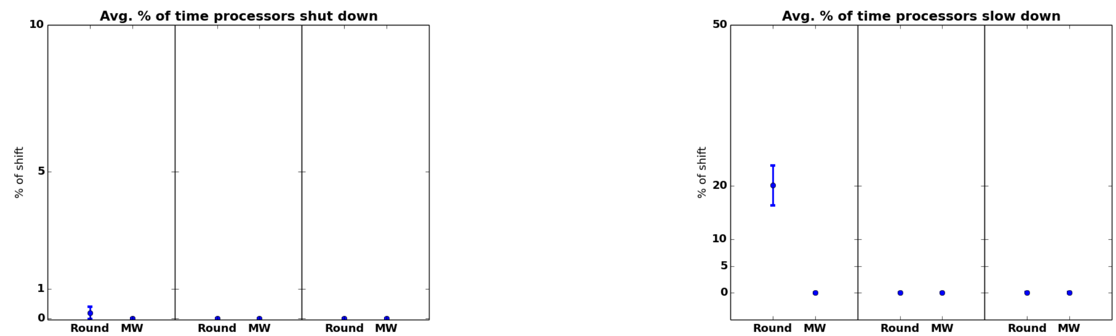
(b) 95% confidence intervals on fraction of time processors slowed down for 30 replications.

Figure 2: Comparing shutdown/slowdown metrics for nonlinear flow-rate model target-matching policy on processing metrics with queuing constraints (Q) and without queuing constraints (NQ). Three panels are for 15, 25, and 35 trucks, respectively.

We observe that the queueing constraints make no difference in our ability to meet processing targets for 25 and 35 trucks. When only 15 trucks are available, however, use of queueing constraints significantly improves the efficiency of the system. With the queueing constraints, neither of the processing sites is ever shut down and the percentage of time required to slow down reduced from nearly 50% to 30%. The mine target extraction amount and ore quality metrics are similar with and without the queueing constraints added. This experiment clearly demonstrates the positive impact of specifically considering queueing considerations in the average flow-rate model.

5.3.2 Capacity constraints in the MIP dispatching model

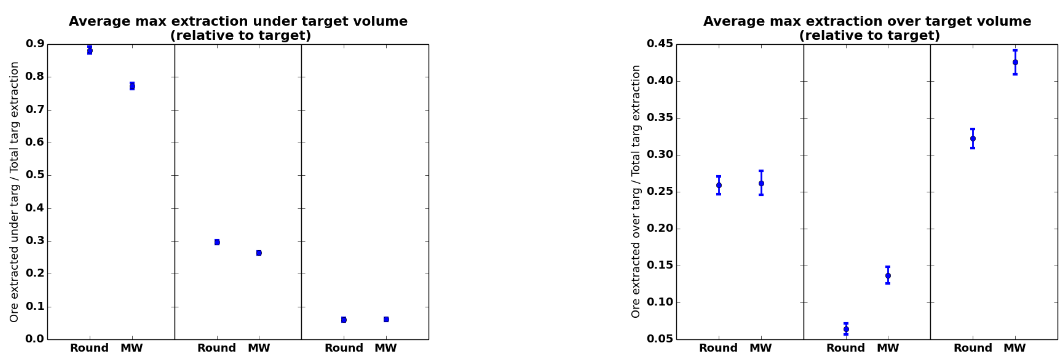
To evaluate the effectiveness of the moving window capacity constraints, we run the simulation first by simply rounding up the truck capacity in each time period at each location to the nearest integer (Round). We then run the simulation with moving window constraints (constraints (46)-(53)) that attempt to limit the "round-up error" (MW). Figures 3, 4, and 5 display the results of this study.



(a) 95% confidence intervals on fraction of time processors shut down for 30 replications.

(b) 95% confidence intervals on fraction of time processors slowed down for 30 replications.

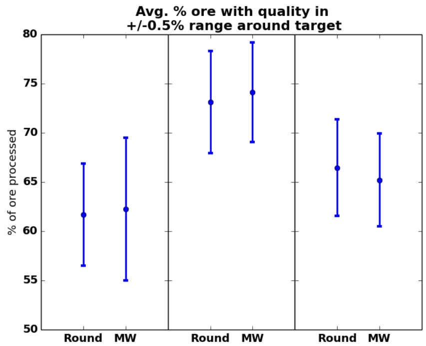
Figure 3: Comparing performance of discrete-time MIP policy on processing metrics with moving window capacity constraints (MW) and with only per-period capacity constraints (Round). Panels are for 15, 25, and 35 trucks, respectively.



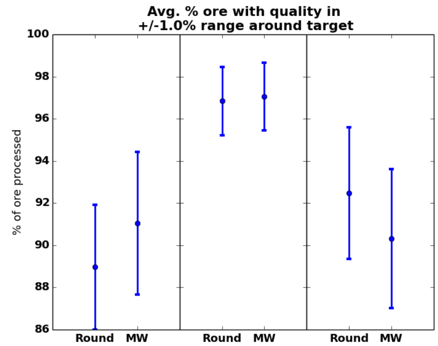
(a) 95% confidence intervals on maximum mining target shortfall across all mining sites (relative to target extraction) for 30 replications.

(b) 95% confidence intervals on maximum excess of mining targets across all mining sites (relative to target extraction) for 30 replications.

Figure 4: Comparing performance of discrete-time MIP policy on mining target metrics with moving window capacity constraints (MW) and with only per-period capacity constraints (Round). Panels are for 15, 25, and 35 trucks, respectively.



(a) 95% confidence intervals on percent of ore processed within ± 0.5% of ore quality target for 30 replications.



(b) 95% confidence intervals on percent of ore processed within ± 1.0% of ore quality target for 30 replications.

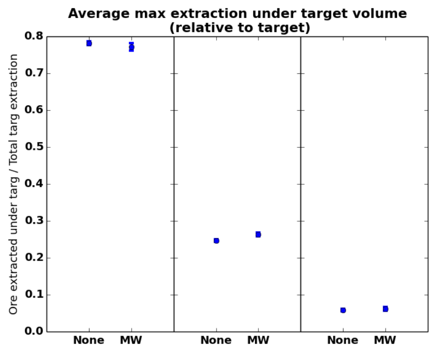
Figure 5: Comparing performance of discrete-time MIP policy on ore-quality metrics with moving window capacity constraints (MW) and with only per-period capacity constraints (Round). Panels are for 15, 25, and 35 trucks, respectively.

In Figure 3, we observe that the capacity constraints make no difference in our ability to meet processing targets for 25 and 35 trucks. When only 15 trucks are available, however, use of capacity constraints significantly improves the system processing performance. In Figures 4 and 5, we see that the two methods are nearly indistinguishable with respect to the mine extraction and ore quality metrics, although the MW variation does appear to over-mine slightly more than simply rounding up. We conclude that the moving window capacity constraints are critical to meeting the processing criteria when few trucks are available, but the use of the constraints has less impact when there sufficiently many trucks.

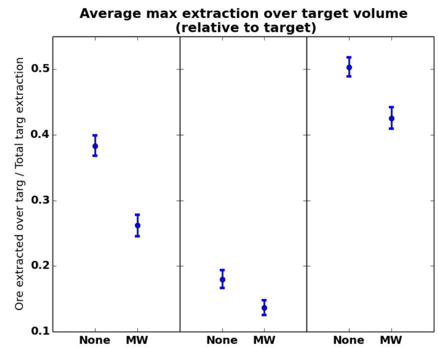
5.3.3 Ore quality constraints in the MIP dispatching model

To evaluate the effectiveness of the moving window ore quality constraints (constraints (60)-(64)), we run the simulation first by stopping after minimizing the mining target deviation (P2) rather than continuing to minimize the ore quality target violation (P3). This case is labeled (None) in the result panels. We then run the simulation using the moving window constraints that approximate ore quality violation (MW). Implementing the ore-quality minimization phase has no impact on the system's ability to meet processing targets, as neither processing site is slowed down nor shut down in any of the 30 simulation replicates. Figures 6 and 7 display the confidences intervals of estimates of the mine target and ore quality metrics.

Interestingly, although we would expect the MW variation to perform slightly worse on the mining metrics, in Figure 6, we see that the two variations are nearly indistinguishable in terms of under-mining, and the MW constraints in the simulation actually seem to reduce the over-mining metric. As expected, Figure 7 demonstrates that the ore quality values significantly improve when using the moving window ore quality constraints regardless of the number of trucks. We conclude that the moving window ore quality constraints are critical to meeting the ore quality targets and, surprisingly, might even improve the mining extraction target metrics.

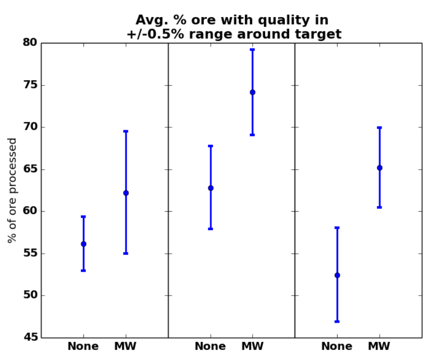


(a) 95% confidence intervals on maximum mining target shortfall across all mining sites (relative to target extraction) for 30 replications.

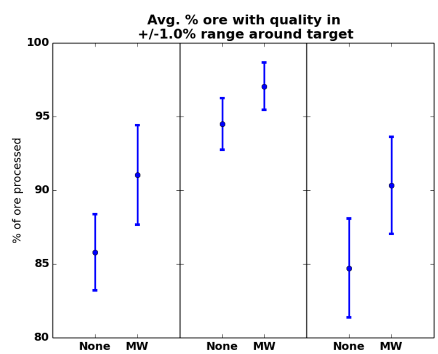


(b) 95% confidence intervals on maximum excess of mining targets across all mining sites (relative to target extraction) for 30 replications.

Figure 6: Comparing performance of discrete-time MIP policy on mining target metrics with moving window ore quality constraints (MW) and without the constraints (None). Panels are for 15, 25, and 35 trucks, respectively.



(a) 95% confidence intervals on percent of ore processed within $\pm$ 0.5% of ore quality target for 30 replications.



(b) 95% confidence intervals on percent of ore processed within $\pm$ 1.0% of ore quality target for 30 replications.

Figure 7: Comparing performance of discrete-time MIP policy on ore-quality metrics with moving window ore quality constraints (MW) and without the constraints (None). Panels are for 15, 25, and 35 trucks, respectively.

5.4 Evaluation of Policies

We conclude the computational experiments with a side-by-side comparison of the nonlinear average flow-rate model with target-matching policy described in Section 2 with the MIP-based dispatching policy of Section 3. We will compare the policies on mines with different characteristics, varying instance parameters that influence policy performance. Specifically, we vary the number of trucks in the mine, the variance of the means of the ore quality at mining sites, and the variance of the truck travel times. We test the policies on the same instance used to evaluate model components in Section 5.3 with $|G| = 5$, $|B| = 4$, $|P| = 2$, $|S| = 2$, $|D| = 2$, $|R| = 1$, and $|K| = 1$, For studies not involving the varying the number of trucks, there are 35 available trucks in the simulation.

5.4.1 Varying numbers of trucks

In our first experiment, we test the two dispatching policies on instances with 15, 25, 35, and 45 trucks. In none of the 30 replications for either policy or for any number of available trucks did a processing site have to shut down. However, with only 15 trucks, employing the average flow-rate model and target matching policy (G) resulted in significant require processor slowdown time, as seen in Figure 8.

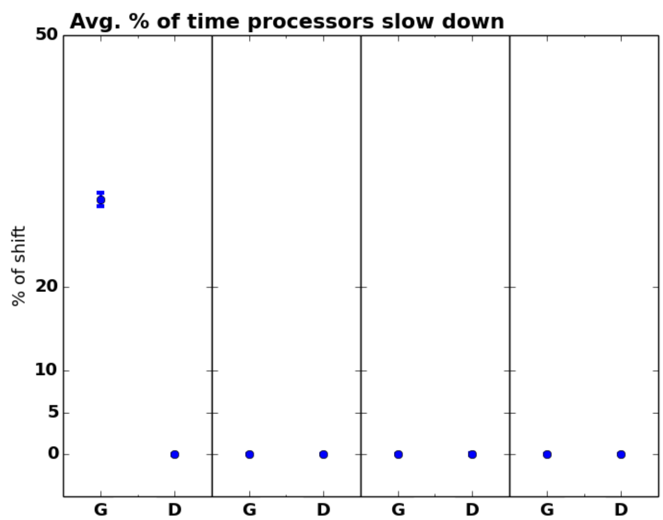
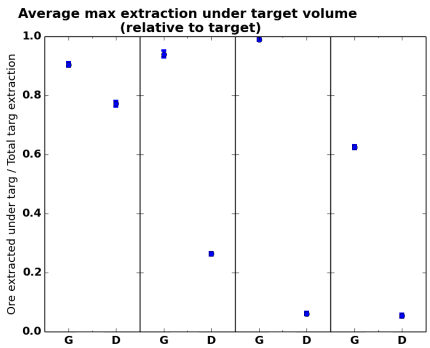


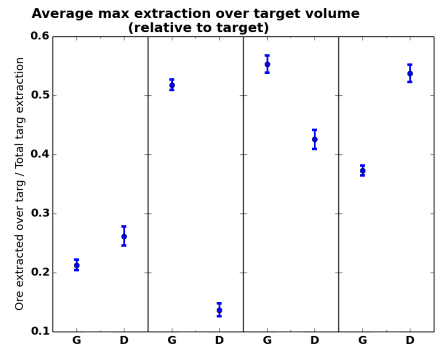
Figure 8: 95% confidence intervals on fraction of time processors slowed down for 30 replications for average flow-rate target-matching policy (G) and MIP-dispatching policy (D) for 15, 25, 35, and 45 trucks.

With respect to the mining target metrics, Figure (9) shows that that the discrete-time MIP policy under-mines significantly less than the average flow rate target-matching policy. In fact, the target-matching policy nearly *ignores* at least one mining site completely until there are at least 45 trucks available. The policies are difficult to distinguish on the over-mining metrics.

Finally, on the ore quality metrics displayed in Figures (9), we observe that the discrete-time MIP policy has the highest ore quality of the two policies for all instances. The target-matching policy gets closer to meeting ore quality targets with more trucks available, whereas the discrete-time MIP policy meets ore quality targets better when fewer trucks are available. In the 45-truck instance, the target-matching policy and discrete-time MIP policy have similar ore quality. Clearly in this study, the MIP-based dispatching model outperforms the policy based on matching the average flow rate.

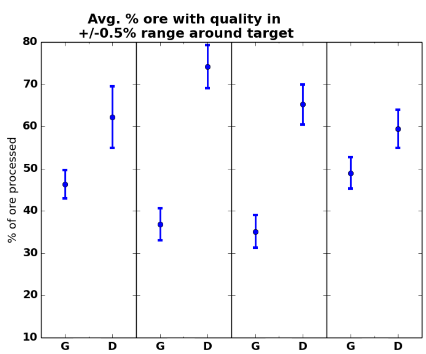


(a) 95% confidence intervals on maximum mining target shortfall across all mining sites (relative to target extraction) for 30 replications

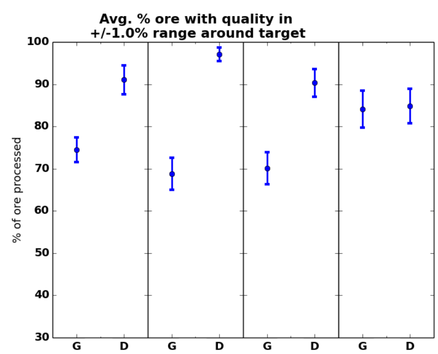


(b) 95% confidence intervals on maximum excess of mining targets across all mining sites (relative to target extraction) for 30 replications

Figure 9: Comparing average flow-rate target matching policy (G) and MIP-based dispatching policy (D) on mining extraction metrics with 15, 25, 35, and 45 trucks.



(a) 95% confidence intervals on percent of ore processed within $\pm$ 0.5% of ore quality target for 30 replications.



(b) 95% confidence intervals on percent of ore processed within $\pm$ 1.0% of ore quality target for 30 replications.

Figure 10: Comparing average flow-rate target matching policy (G) and MIP-based dispatching policy (D) on ore quality metrics with 15, 25, 35, and 45 trucks.

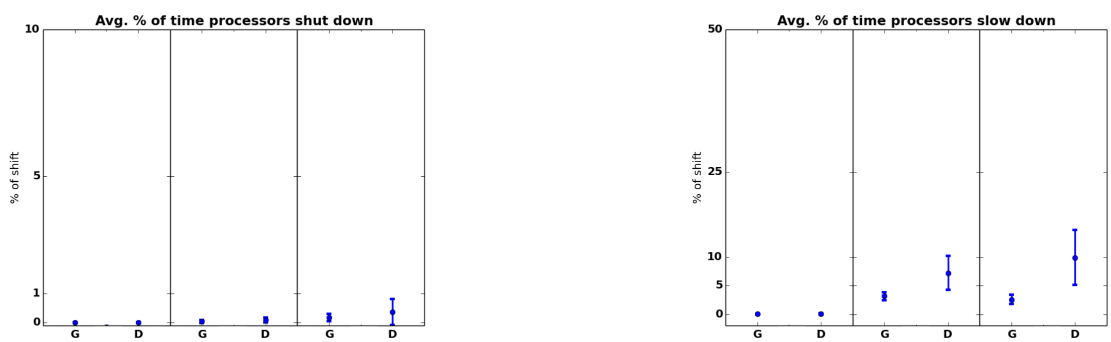
5.4.2 Varying the travel times

We next compare the two dispatching policies on instances with increasing variance in the travel times between pairs of locations. In the base mining instance, the travel times are randomly generated from Johnson S_U distributions, as described in Section 5.1. In the second instance, we keep the mean travel time the same between each pair of sites, but with probability $p = 0.2$, we double the travel time. In the third instance, we again keep the mean travel time fixed between each pair of locations but with probability $p = 0.4$, we double the travel time. This is meant to mimic the real-lift event of a truck encountering operational problems en-route to its destination. Figures 11-13 display the output for the policy comparison on these instances.

We first observe from Figure 11 that both policies must slow and shut down the processors slightly more often when there is higher variance in travel times. However, the average flow-rate target-matching policy outperforms the discrete-time MIP policy when the travel time variance is high.

On the mining metrics shown in Figure 12, the discrete-time MIP policy under-mines significantly less than the target-matching policy for all instances. In fact, at least one mine is completely ignored by the target-matching policy in all of the replications. The discrete-time MIP policy also over-mines significantly less than the target-matching policy on all instances. There is a noticeable increase in the maximum shortfall for the discrete-time MIP policy when travel time variance increases. Both policies over-mine less when travel time variance is high. We note that there is an initial change in mining metrics when the travel time variance is increased from the base instance, but further increasing the variance does not produce another similar change.

On the ore quality metrics display in Figure 13, we observe that the discrete-time MIP policy has the highest ore quality for all instances. There is a slight decrease in the ore quality as the variance of travel times increases for both policies. In this study, it is difficult to say which policy is to be preferred, as the average flow rate policy performs best with respect to the most important shutdown and slowdown metrics, but the MIP-based dispatching model performs better with respect to the remaining metrics.



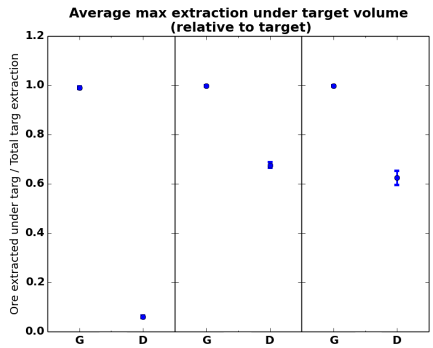
(a) 95% confidence intervals on fraction of time processors shut down for 30 replications.

(b) 95% confidence intervals on fraction of time processors slowed down for 30 replications.

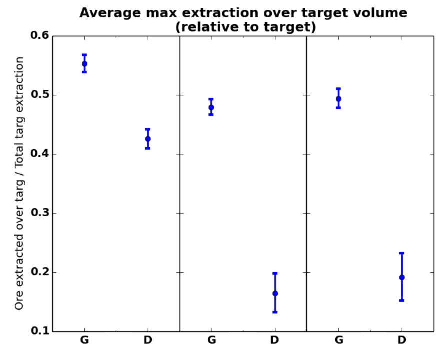
Figure 11: Comparing average flow-rate policy (G) and MIP-based policy (D) on processing metrics with low, medium, and high variance in travel times. Panels are for instances doubling travel times with probability 0, 0.2, 0.4, respectively.

5.4.3 Varying the ore quality at mining sites

We compared the dispatching policies on instances with increasing variance of ore quality between mining sites. In the base instance, each mining site has its average ore quality (M_{ir} in equation (65))

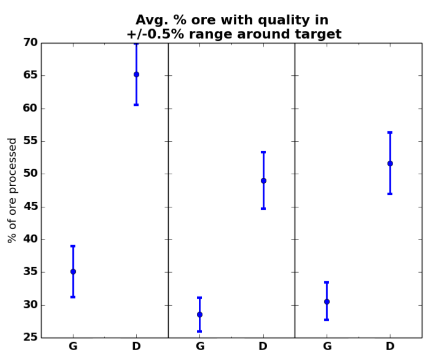


(a) 95% confidence intervals on maximum mining target shortfall across all mining sites (relative to target extraction) for 30 replications.

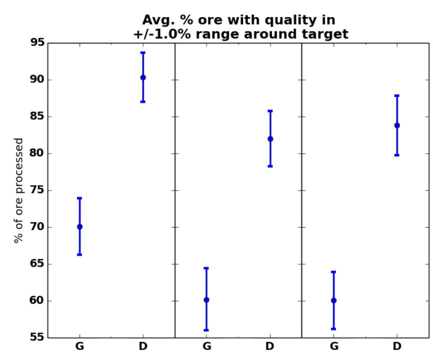


(b) 95% confidence intervals on maximum excess of mining targets across all mining sites (relative to target extraction) for 30 replications.

Figure 12: Comparing average flow-rate policy (G) and MIP-based policy (D) on mining extraction metrics with low, medium, and high variance in travel times. Panels are for instances doubling travel times with probability 0, 0.2, 0.4, respectively.



(a) 95% confidence intervals on percent of ore processed within ± 0.5% of ore quality target for 30 replications.

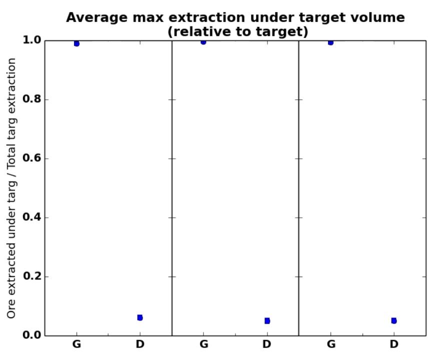


(b) 95% confidence intervals on percent of ore processed within ± 1.0% of ore quality target for 30 replications.

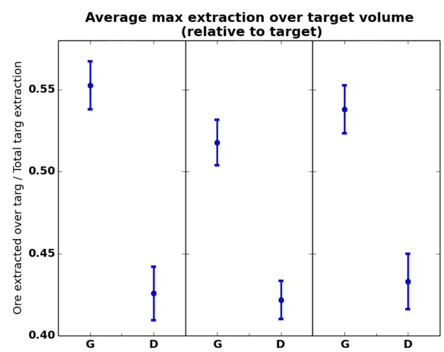
Figure 13: Comparing average flow-rate policy (G) and MIP-based policy (D) on ore quality metrics with low, medium, and high variance in travel times. Panels are for instances doubling travel times with probability 0, 0.2, 0.4, respectively.

close to the target ore quality for the processing sites. In the second instance, we keep the mean ore quality across all mining sites close to the target processing quality, but allow the quality at individual mining sites to deviate from the mean by as much as 0.4 times the target quality. In the third instance, we again keep the mean quality close to the target quality but allow the quality at individual mining sites to deviate from the mean by as much as 0.7 times the target quality.

Not surprisingly, since the potential throughput is not varied in this study, all replications of the simulation were able to keep the processors running at full capacity for the duration of the simulation. On the mining extraction metrics displayed in Figure 14, the discrete-time MIP policy both under-mines and over-mines significantly less than the target-matching policy for all instances. There appears to be a reduction in the maximum amount over-mined for both policies when the ore quality variance increases from the base instance. On the ore quality metrics displayed in Figure 15, we observe that the discrete-time MIP policy has the highest ore quality for all instances. As expected, we observe a decrease in the ore quality metrics as the inter-mine ore quality variance increases. However, the decrease is less significant for the discrete-time MIP policy than for the greedy target-matching policy. In this study, we can conclude that the MIP-based dispatching policy is clearly superior to the average flow-rate-based policy.



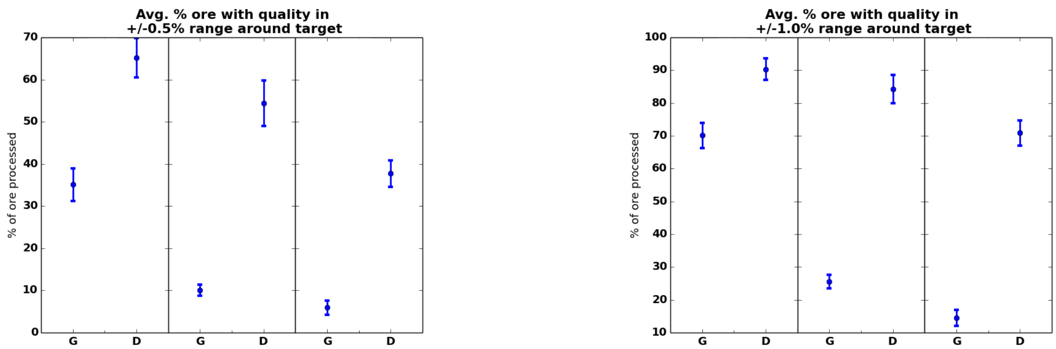
(a) 95% confidence intervals on maximum mining target shortfall across all mining sites (relative to target extraction) for 30 replications.



(b) 95% confidence intervals on maximum excess of mining targets across all mining sites (relative to target extraction) for 30 replications.

Figure 14: Comparing average flow-rate policy (G) and MIP-based policy (D) on mining extraction metrics with low, medium, and high variance in inter-mine ore-quality. Panels are for instances with the low, medium, and high levels of variance, respectively.

The results in this section allow us to draw some conclusions about the two dispatching policies. First, the discrete-time MIP dispatching policy consistently outperforms the average flow-rate based target matching policy on almost every metric in every instance. When the targets are easy to meet (e.g., when plenty of trucks are available), the discrete-time MIP policy does not outperform the target-matching as much. Finally, we note that when the discrete-time MIP model can take advantage of the system information, it performs better. When the dispatching model does not have access to the information, such as the instances with high travel time variance, it does not perform as well. However, it still outperforms the target-matching policy on two of the three key metrics. Thus, we conclude that the discrete-time MIP policy is a good dispatching policy to use for an open-pit mine with a wide range of characteristics.



(a) 95% confidence intervals on percent of ore processed within ± 0.5% of ore quality target for 30 replications.

(b) 95% confidence intervals on percent of ore processed within ± 1.0% of ore quality target for 30 replications.

Figure 15: Comparing average flow-rate policy (G) and MIP-based policy (D) on ore-quality metrics with low, medium, and high variance in inter-mine ore quality. Panels are for instances with the low, medium, and high levels of variance, respectively.

6 Conclusion

We develop two novel truck dispatching policies for use in open pit mining. The first policy solves a nonlinear optimization model incorporating queueing effects to compute optimal average flow rates and dispatches trucks in a manner that greedily matches these flow-rates as closely as possible. The second dispatching policy is based on a carefully-constructed time-discretized MIP model solved and implemented in a rolling-horizon fashion. Both policies are computationally efficient enough to be employed in a real-time dispatching setting. We demonstrate the relative effectiveness of each of the policies by embedding them in a discrete-event simulation of an open pit mine. By altering characteristics of the mine, we show that the discrete-time MIP-based policy consistently outperforms the average flow-rate matching policy. Thus we argue for more dispatching operations to consider using mixed integer programming as a computational tool in real-time settings.

References

Ahangaran D, Yasrebi A, Wetherelt A, Foster P (2012) Real–time dispatching modelling for trucks with different capacities in open pit mines. Archives of Mining Sciences 57(1):39–52, DOI 10.2478/v10267-012-0003-8

Arnold MH, White JW (1983) Computer-based truck dispatching. In: Proceeding of World Mining Congress, Serbia, pp 53–57

Ataeepour N, Baafi EY (1999) Arena simulation model for truck-shovel operation in despatching and non-despatching modes. International Journal of Surface Mining, Reclamation and Environment 13(3):125–129

Bonates E (1992) The development of assignment procedures for semi-automated truckshovel systems. PhD thesis, McGill University

Burt CN (2008) An optimisation approach to materials handling in surface mines. PhD thesis, Curtin University of Technology

Chatterjee PK, Brake DJ (1981) Truck dispatching and simulation methods in open-pit operations. CIM Bulletin 74(835):102–107

Choudhary RP (2015) Optimization of Load – Haul – Dump Mining System By OEE and Match Factor for Surface Mining. International Journal of Applied Engineering and Technology 5(2):96–102

Danna E, Rothberg E, Pape CL (2005) Exploring relaxation induced neighborhoods to improve mip solutions. Mathematical Programming 102(1):71–90, DOI 10.1007/s10107-004-0518-7, URL http://dx.doi.org/10.1007/s10107-004-0518-7

Elbrond J, Soumis F (1987a) Towards integrated production planning and truck dispatching in open pit mines. International Journal of Surface Mining, Reclamation and Environment 1(1):1–6, DOI 10.1080/09208118708944095

Elbrond J, Soumis F (1987b) Towards integrated production planning and truck dispatching in open pit mines. International Journal of Surface Mining, Reclamation and Environment 1(1):1–6

Ercelebi SG, Bascetin A (2009) Optimization of shovel-truck system for surface mining. Journal of the Southern African Institute of Mining and Metallurgy 109(7):433–439

Faraji R (2013) A comparison between linear programming and simulation models for a dispatching system in open pit mines. PhD thesis, University de Montreal

Forsman B, Rönnkvist E, Vagenas N (1993) Truck dispatch computer simulation in Aitik open pit mine. International Journal of Surface Mining, Reclamation and Environment 7(3):117–120

Goodman G, Sarin S (1988) Using mathematical programming to develop optimal overburden transport strategies in a surface coal mining operation. International Journal of Surface Mining, Reclamation and Environment 2(1):51–58

Hauck R (1979) Computer-controlled truck dispatching in open-pit mines. Computer Methods for the 80:735–742

Hauck RF (1973) A real-time dispatching algorithm for maximizing open-pit mine production under processing and blending requirements. In: Proceedings, Seminar on Scheduling in Mining, Smelting, and Steelmaking, Canadian Institute of Mining and Metallurgy

Kappas G, Yegulalp TM (1991) An application of closed queueing networks theory in truck-shovel systems. International Journal of Surface Mining, Reclamation and Environment 5(1):45–51

Li Z (1990) A methodology for the optimum control of shovel and truck operations in open-pit mining. Mining Science and Technology 10(3):337–340

Lizotte Y, Bonates E (1987) Truck and shovel dispatching rules assessment using simulation. Mining Science and technology 5(1):45–58

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, DOI 10.1016/j.ijmst.2013.01.017, URL `http://dx.doi.org/10.1016/j.ijmst.2013.01.017`

Moreno E, Rezakhah M, Newman A, Ferreira F (2017) Linear models for stockpiling in open-pit mine production scheduling problems. European Journal of Operational Research 260(1):212–221

Munirathinam M, Yingling J (1994) A review of computer-based truck dispatching strategies for surface mining operations. International Journal of Surface Mining and Reclamation 8(1):1–15

Naoum S, Haidar A (2000) A hybrid knowledge base system and genetic algorithms for equipment selection. Engineering Construction and Architectural Management 7(1):3–14

Oraee K, Asi B (2004) Fuzzy model for truck allocation in surface mines. In: Mine Planning and Equipment Selection 2004, Thirteenth International Symposium on Mine Planning and Equipment Selection, Taylor & Fancis (Routledge, USA), pp 585–593

Panagiotou G, Michalakopoulos T (1995) Bedisp – a computer-based truck dispatching system for small-medium scale mining operations. Mine planning and equipment selection pp 481–486

Sadler W (1988) Practical truck dispatch-a micro computer based approach. Computer applications in the mineral industry pp 495–500

Smith A (2019) New integer programming approaches to open-pit mining and metabolic engineering problems. PhD thesis, University of Wisconsin, Department of Industrial and Systems Engineering

Smith ML, Wicks SJ (2014) Medium-term Production Scheduling of the Lumwana Mining Complex using MIP with a Rolling Horizon. Interfaces 44(2):176–194, DOI 10.1287/inte.2014.0737

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

Subtil RF, Silva DM, Alves JC (2011) A Practical Approach to Truck Dispatch for Open Pit Mines. APCOM Symposium (September):765–777

Ta CH, Kresta JV, Forbes JF, Marquez HJ (2005) A stochastic optimization approach to mine truck allocation. International Journal of Surface Mining, Reclamation and Environment 19(3):162–175, DOI 10.1080/13895260500128914

Temeng V (1998) A Nonpreemptive Goal Programming Approach to Truck Dispatching in Open Pit Mines. Mineral resources engineering 07(2):59–67

Temeng V, Otuonye F, Jr JF (1997) Real-time truck dispatching using a transportation algorithm. International Journal of Surface Mining, Reclamation and Environment 11(4):203–207

Topal E, Ramazan S (2012) Mining truck scheduling with stochastic maintenance cost. Journal of Coal Science and Engineering 18(3):313–319, DOI 10.1007/s12404-012-0316-4

Tu JH, Hucka VJ (1985) Analysis of open-pit truck haulage system by use of a computer-model. CIM Bulletin 78(879):53–59

White JW, Arnold M, Clevenger J (1982) Automated open-pit truck dispatching at tyrone. E&MJ-Engineering and Mining Journal 183(6):76–84

White JW, Olson J, Vohnout S (1993) On improving truck/shovel productivity in open pit mines. CIM bulletin 86:43–43

A Instance Data

This appendix describes how we created families of instances for testing the models and dispatching methods in Sections 4 and 5. In Table 7, we give each instance parameter along with a brief description of how the parameter is used. We also give the range of values this parameter can take and the method we use to generate the parameter. If the parameter generation method is "Fx", then the parameter value is given in the Range column. If the parameter generation methods is "UC," then the parameters are randomly generated from a continuous uniform distribution with the bounds given in the Range column. "UI" parameters are randomly generated from an integer uniform distribution with the bounds given in the "Range" column. The total number of trucks at different locations are generated as specified in the table, and their specific locations are assigned at random. To generate mining extraction targets, each mine is given a "size" parameter, s_i, and its relative size $\alpha_i = s_i / \sum i s_i$ is used in the extraction target calculation. Mining locations in the mine are randomly generated on a circular map between 0.9 and 1.5 miles from the origin. Processing, stockpile, and dump sites are randomly generated on the map between 0.2 and 1.2 miles from the origin. The distance is calculated as the (fixed) Euclidean distance between two locations. The conversion factor of 4/3 in the translation to periods τ_{ij}, assumes that trucks travel at 40mph.

Symbol	Description	Range	Method
α_k	Size of truck type $k \in K$	[350, 420]	UC
$\bar{p}_i$	Maximum number of trucks that can be processed per 30-sec time period at site $i \in N$	Mines: 1.98; Processing and stockpiles: 2.0; Dump: 4.0;	Fx
Θ_j^P	Amount that should be processed by processing site $j \in P$ during each period	[9000/30, 12000/30]	UC
Θ_i^M	Amount that should be extracted from mine $i \in M$ during the planning horizon	$T^{\max} \times \alpha_i \times [120, 160]$	UC
C_j	Buffer $j \in P$ maximum capacity	[9000, 12000]	UC
I_{j0}	Amount in buffer $j \in P$ at time 0	[0, C_j]	UC
$dist_{i,j}$	Distance between $i \in N$ and $j \in N$	See above	F
τ_{ij}	Number of periods needed to travel from i to j, $i, j \in N$	$\lceil dist_{i,j}/1.333 \rceil$	Fx
Q_{jr}^P	Target grade-element ratio $r \in R$ of the ore at processing site $j \in P$	[0.05, 0.15]	UC
Q_{ir}^M	Grade-element ratio $r \in R$ at mine or stockpile $i \in M \cup S$	$[0.5 \times Q_{jr}^P,\ 1.5 \times Q_{jr}^P]$	UC
Q_{jr0}^I	Grade-element ratio $r \in R$ of the ore in buffer $j \in P$ at time 0	$[0.9 \times Q_{jr}^P,\ 1.1 \times Q_{jr}^P]$	UC
$\sum \hat{x}_{ijkt}$	# of trucks of type k that left from $i \in N$ to $j \in N$ at time $t = -\tau_{ij}, ..., 0$	[0 # trucks)]	UI
$\sum \hat{g}_{kj}$	# of trucks of type $k \in K$ available at $j \in N$ at $t = 0$	[0, # trucks - $\sum \hat{x}_{ijkt}$]	UI
$\sum \hat{q^M}_{ik}$	# of trucks of type $k \in K$ waiting in a queue at mine $i \in M$ at time $t = 0$	[0, # trucks - $\sum \hat{x}_{ijkt}$ - $\sum_{k \in K, j \in N} \hat{g}_{jk}$]	UI
$\sum \hat{q^P}_{ijk}$	# of trucks of type $k \in K$ waiting in a queue at $j \in P \cup D$ holding load from mine $i \in M$ at time $t = 0$	[0, # trucks - $\sum \hat{x}_{ijkt}$ - $\sum_{k \in K, j \in N} \hat{g}_{jk}$ - $\sum_{i \in M, k \in K} \hat{q^M}_{ik}$]	UI

Table 7: Parameter ranges for instances used in computational experiments