Optimizing truck dispatching decisions in open-pit mining using integer programming

Jim Luedtke
Amanda G. Smith, Jeff Linderoth

Industrial & Systems Engineering
University of Wisconsin–Madison

DEPLAMIN 2018
Truck dispatching in an open-pit mine

- HUGE trucks used to move ore from mining sites to processing and waste sites
- Dispatching: After loading or unloading ore, where to go next?
- High uncertainty in travel and loading/unloading time ⇒ Need real-time decision-making

Big Question

Can integer programming be effective for real-time decision-making?

Source: www.edumine.com/programs/professional-programs/mining/open-pit-mining-operations
Mining operations overview

- **Mine 1**: Quality: waste, Extraction target $\theta_1$
- **Mine 2**: Quality: 6%, Extraction target $\theta_2$
- **Mine 3**: Quality: 5%, Extraction target $\theta_3$

- **Buffer 1**: Rate target $p_1$, Quality target 5.5%
- **Buffer 2**: Rate target $p_2$, Quality target 5.5%
- **Proc. 1**:
- **Proc. 2**:

---

Smith, Linderoth, Luedtke (UW–Madison)

Optimizing truck dispatching decisions

November 17, 2018
Dispatching problem – General structure

Given:
- Processing rate targets
- Mining extraction targets
- Ore quality targets
- Ore quality at each mining location
- Maximum loading/unloading capacity

Decisions to make:
- Every time a truck finishes loading/unloading, where should it be dispatched?

Objectives:
- Keep processors operating at full speed
- Meet shift targets at mining sites
- Maintain good quality ore mixture at processing site buffers
Dispatching methods

Existing method:

- Solve aggregate LP to get target average flow rates of trucks between locations
- We propose some extensions to this approach
Dispatching methods

Existing method:
- Solve aggregate LP to get target average flow rates of trucks between locations
- We propose some extensions to this approach

Proposed discrete-time mixed-integer programming (MIP) approach:
- Solve MIP over medium-time scale to get dispatch decision
  - Makes current decisions based on full state of mine, anticipating queueing effects
- Use in model-predictive control loop (rolling horizon)
Variables:

- \( x_{ij} \) – average flow rate in trucks/hour from location \( i \) to location \( j \)
- \( L_i \) – average number of trucks waiting in queue or being (un)loaded at location \( i \)
Average flow-rate model – Variables and basic constraints

Variables:
- $x_{ij}$ – average flow rate in trucks/hour from location $i$ to location $j$
- $L_i$ – average number of trucks waiting in queue or being (un)loaded at location $i$

Basic Constraints:
- Balance rate of trucks entering and exiting each location
- Limited number of trucks in system
- Constraints for approximating time trucks spend in queue
Average flow-rate model – Three-phase hierarchical objective

Multiple objectives:
- Minimize processing rate shortage (usually zero)
- Minimize violation from mining extraction targets
- Minimize violation from quality targets
Average flow-rate model – Three-phase hierarchical objective

Multiple objectives:
- Minimize processing rate shortage (usually zero)
- Minimize violation from mining extraction targets
- Minimize violation from quality targets

\[
\text{PHASE I} \quad \text{min proc. rate viol.} \\
\text{PHASE II} \quad \text{min mining violation} \\
\text{PHASE III} \quad \text{min quality violation}
\]

\[
\text{proc viol} \equiv \text{min viol} \quad \text{mining viol} \leq \beta \times \text{min viol}
\]
Flow-rate dispatching policy – Greedy target-matching

Main idea: Simple way to match dispatching decisions to flow rate targets

- Let $\bar{x}_{ij}$ be target flow rate $i \rightarrow j$ from average flow-rate model

Dispatch truck available at site $i$ at time $t \leq T$ according to:

$$\hat{j} \in \arg \max_j \left\{ \bar{x}_{ij}(t/T) - f_{ij} + \max_k \{\bar{x}_{jk}(t/T) - f_{jk}\} \right\}$$

- Update $f_{i\hat{j}} \leftarrow f_{i\hat{j}} + 1$
- Find new target flow rates by re-solving flow-rate model at regular intervals
Flow-rate dispatching policy – Greedy target-matching

Main idea: Simple way to match dispatching decisions to flow rate targets

- Let $\bar{x}_{ij}$ be target flow rate $i \rightarrow j$ from average flow-rate model

Dispatch truck available at site $i$ at time $t \leq T$ according to:

$$\hat{j} \in \arg \max_j \left\{ \bar{x}_{ij}(t/T) - f_{ij} + \max_k \{\bar{x}_{jk}(t/T) - f_{jk}\} \right\}$$

- Update $f_{i\hat{j}} \leftarrow f_{i\hat{j}} + 1$
- Find new target flow rates by re-solving flow-rate model at regular intervals
MIP-based dispatching policy

**Idea:** Solve a short-term discrete-time MIP to find dispatching decision

- Ideal: re-solve MIP at every dispatching decision
- Implementation: Solve instance of the MIP dispatching model at regular intervals (e.g., 5 minutes)
- Based on MIP solution, for each location, create ordered list of “next destinations” for trucks finishing at that location
- When a truck finishes loading/unloading at a location, dispatch to next location in the list
Primary Decision Variables:

- $x_{ijt}$ – number of trucks sent from $i$ to $j$ in period $t \in T$
- $y_{it}^M$ – number of trucks loaded at mine $i$ in period $t \in T$
- $y_{ijt}^P$ – number of trucks unloaded at $j$, from mine $i$ in period $t \in T$
- $z_{jt}$ – production amount at processing site $j$ in period $t \in T$
Discrete-time MIP dispatching model

Primary Decision Variables:

- \( x_{ijt} \) – number of trucks sent from \( i \) to \( j \) in period \( t \in T \)
- \( y_{it}^M \) – number of trucks loaded at mine \( i \) in period \( t \in T \)
- \( y_{ijt}^P \) – number of trucks unloaded at \( j \), from mine \( i \) in period \( t \in T \)
- \( z_{jt} \) – production amount at processing site \( j \) in period \( t \in T \)

Basic constraints:

- Measure mine extraction target deviation (over full horizon)
- “Truck balance” constraints at mines and processing sites (trucks wait in queue until processed)
**Dispatching model – Max capacity**

**Goal:** Limit number of trucks loaded/unloaded based on loading capacity

- **Input:** $\bar{p}_i$: average number of trucks that can be loaded/unloaded in a period at site $i$
- $y_{it}^M$: number of trucks loaded at mine $i$ in period $t$
- **Challenge:** $\bar{p}_i$ is not integer, but $y_{it}^M, y_{ijt}^P$ are
- In particular:

\[ y_{it}^M \leq \bar{p}_i \implies y_{it}^M \leq \lfloor \bar{p}_i \rfloor \]

- Rounding down underestimates capacity
- If $\bar{p}_i < 1$, we can’t do anything
Dispatching model – Max capacity

Alternative approach
For every mine $i$ and every time period $t$:

$$y_{it}^M \leq \lceil \bar{p}_i \rceil \implies \text{overestimates capacity}$$
Dispatching model – Max capacity

**Alternative approach**

For every mine $i$ and every time period $t$:

$$ y_{it}^M \leq \lceil \bar{p}_i \rceil \implies \text{overestimates capacity} $$

**Overlapping intervals to limit overestimation**

| 0 | \cdots | $t$ | $t+1$ | $t+2$ | \cdots | $t+d-1$ | $t+d$ | $t+d+1$ |

Choose (small) $d$ for each site to minimize error from rounding up
Goal: Model effect of blending by tracking quality of ore processed at the processing sites

- Keep track of attribute level in buffer, as well as total ore: $\hat{C}_{j,t}$
- Ore quality is the ratio $\hat{C}_{j,t}/I_{j,t}$
**Goal:** Model effect of blending by tracking quality of ore processed at the processing sites

- Keep track of *attribute* level in buffer, as well as total ore: $\hat{C}_{j,t}$
- Ore quality is the ratio $\hat{C}_{j,t}/I_{j,t}$
- Challenge: Nonconvex model

$$\hat{C}_{jt} = \hat{C}_{j,t-1} + \sum_{i \in M} (Q_i^M \alpha y_{ijt}) - \frac{\hat{C}_{j,t-1}}{I_{j,t-1}} z_{jt}$$

- Convex reformulation is not possible $\Rightarrow$ Mixed-integer nonlinear (nonconvex) program!
**Idea:** Keep quality added to buffers close to target value over sequence of windows

- Measure *quality added* to buffers rather than *average quality* in buffer
- Allows modeling approximate quality as series of linear constraints
Multiple objectives:

- Always meet processing rate targets
- Try to meet mining extraction targets
- Try to meet quality targets
Multiple objectives:

- Always meet processing rate targets
- Try to meet mining extraction targets
- Try to meet quality targets

**PHASE I**

\[ \text{min proc. rate viol.} \]

**PHASE II**

\[ \text{min mining violation} \]

\[ \text{min viol} \leq \beta \times \text{min viol} \]

**PHASE III**

\[ \text{min quality violation} \]

\[ \text{proc viol} \equiv \text{min viol} \]
Relax-and-fix heuristic

PHASE I solves quickly ✓
PHASE II and PHASE III can be difficult to solve ❌

1. Set time limit on PHASE II and keep the best solution found ✓

2. Use Relax-and-Fix (RAF) heuristic to get a good solution to PHASE III quickly

RAF heuristic for PHASE III:
– Relax integrality constraints for all periods and solve PHASE III model
– If integer variable has value 0 in LP and in PHASE II IP, fix variable to 0 (variation of RINS algorithm [Danna et al., 2005])
– Solve restricted IP
Computational study – Experiment design

Evaluate each policy using discrete-event simulation

- Random: travel times, loading/unloading times, load sizes
- Parameters all provided by industry collaborator as “realistic”
- Processors slow to 60% speed when buffer level low
- Simulate one complete 12-hour shift, including change-over time
- Perform 30 replications to obtain confidence intervals
- Ore quality is measured using nonlinear mixture model
- Vary number of trucks in test instance
Computational study – Experiment design

Evaluate each policy using discrete-event simulation

- Random: travel times, loading/unloading times, load sizes
- Parameters all provided by industry collaborator as “realistic”
- Processors slow to 60% speed when buffer level low
- Simulate one complete 12-hour shift, including change-over time
- Perform 30 replications to obtain confidence intervals
- Ore quality is measured using nonlinear mixture model
- Vary number of trucks in test instance

Compare:

- G: Greedy target matching (based on targets from NLP model)
- M: MIP-based greedy target matching (not described, also based on NLP model targets)
- D: Discrete-time MIP (solved every 5 minutes)
Dispatching policy comparison

95% Confidence Intervals on Processing Volumes (higher is better)

- Target: 200. Subfigures: 15, 25, 35, and 45 trucks
- When “under-trucked”, only discrete-time MIP can meet processing targets
Dispatching policy comparison

95% Confidence Intervals on Mining Target Violations (lower is better)

- The discrete-time MIP-based policy is closest to meeting mining targets
Dispatching policy comparison

95% Confidence Intervals on Ore Quality Metrics (higher is better)

- Discrete-time MIP-based policy uniformly better at meeting ore quality targets
Conclusions

New approach to truck dispatching in open-pit mining
  • Found to be effective in many other instance variations

Conclusion:
  • MIP can be effective in real-time dynamic decision-making

Questions?

jim.luedtke@wisc.edu    amanda.smith@wisc.edu


For every mine $i$ at every time $t$:

- $q_{it}^M$: Number of trucks in queue at time $t$

\[
q_{it}^M = q_{it-1}^M - y_{it}^M + \sum_{j \in N} x_{jit-\tau_{ji}}
\]

Similar constraints for processing sites (except inbound queue keeps track of mine origin of trucks)
Discrete-time MIP dispatching model – Flow balance

For every mine $i$ at every time $t$:

- $q_{it}^M$: Number of trucks in queue at time $t$
- $g_{it}$: Number of trucks available for dispatching at time $t$

\[
\sum_{j \in N} x_{jit - \tau_{ji}} = g_{it} - 1 - \sum_{j \in P} x_{ijt} + y_{it}^M
\]

- Similar constraints for processing sites (except inbound queue keeps track of mine origin of trucks)