Operational Flexibility in Drayage Vehicle Routing

Guangming Zhang

Transportation Center
Industrial Engineering and Management Sciences
Northwestern University

February 8th, 2007
Outline

• Motivation
• Background
• Modeling and solution methods
• Conclusion
Motivation:

- Chicago is a US freight hub
  - 60% of US container movements
- 27 yards: 25,000 truck movements per day
- “Black Hole” of freight movements
  - Transfer times measured in days and weeks
- 40% of a 900 mile movement cost is incurred in drayage portions (< 50 miles)
Drayage Operations

Choice in empty trailer movements
Multi-Resource Routing Problem (MRRP) with Flexible Tasks

- Multiple Resources
  - Tractors + Drivers
  - Trailers
- Tasks: movements to be performed
  - Well-defined Tasks
  - Flexible Tasks

**Given:** a set of tasks with required resources and a network with travel times.

**Find:** a set of routes meeting a chosen objective function (minimizing fleet size, travel time) and observing operating rules for the tasks and resources.
Literature review

- Spasovic (1990) and Morlok & Spasovic (1994)
  - Drayage operations for a single rail carrier
  - Network flow model
- Walker (1992)
  - Short haul truckload problem
  - Network flow model
  - Does not account for the option of repositioning the empty trailers

Arc-based network flow model

Create a time-space network.

Issue: problem size
Node-based VRP representation

4 tasks to be performed:

1. Loaded trailer from S to I
2. Loaded trailer from I to C
3. Supply S with a trailer
4. Transport trailer from C

Solve a vehicle routing problem

- Node SI, IC, ES, and CE
- One execution between ES and CS
- One execution between CS and CE

Multiple Choice
Set Partitioning Formulation

Design vehicle routes to serve a set of tasks that can be implemented by several possible executions.

\[ T \quad \text{Set of tasks to performed} \]
\[ E_i \quad \text{Set of executions for task } i \in T \]
\[ R \quad \text{Set of feasible routes, with cost } c_r \text{ for } r \in R \]
\[ a_{er} = 1 \text{ if route } r \text{ covers execution } e \in E_i \text{; 0 otherwise} \]
\[ x_r = 1 \text{ if route } r \in R \text{ is chosen; 0 otherwise} \]

\[
\min \sum_{r \in R} c_r x_r \quad \text{minimize cost of all routes}
\]

subject to:

\[
\sum_{r \in R} \sum_{e \in E_i} a_{er} x_r = 1 \quad \forall i \in T \quad \text{cover 1 movement for each task}
\]
\[
x_r \in \{0,1\} \quad \forall r \in R \quad \text{binary decision variables}
\]
Branch-and-price scheme

- Replace full set $R$ with a subset $R'$
- Solve linear programming relaxation $LP$
  \[
  \min \sum_{r \in R'} c_r x_r \\
  \text{subject to:} \\
  \sum_{r \in R'} \sum_{e \in E_i} a_{er} x_r \geq 1 \quad \forall i \in T \\
  x_r \geq 0 \quad \forall r \in R'
  \]
- Column generation to iteratively add columns
  - Obtain pricing information for each task
  - Methods to generate routes
- Branch-and-bound to obtain integer solution

Step 0: Initialize

Step 1: Solve $LP$

Step 2: Add routes
Outline

• Motivation
• Background
• Modeling and solution methods
  – Generating executions
  – Generating routes
  – Choosing routes
  – Dynamic requests
• Conclusion
Define executions for flexible tasks

- Can’t enumerate all possible executions (set $E_i$)
- Tradeoff in the number of executions
  - Too few: miss some potential savings
  - Too many: takes too long to solve

\[
\begin{align*}
\min \sum_{r \in R'} c_r x_r \\
\text{subject to:} \\
\sum_{r \in R'} \sum_{e \in E_i} a_{er} x_r & \geq 1 \quad \forall i \in T \\
x_r & \geq 0 \quad \forall r \in R'
\end{align*}
\]

- Generating executions
  - Generating routes
  - Choosing routes
  - Dynamic requests
Variable-Radius method

- Build a neighborhood containing a set number of possible executions.
- Generate sufficient options to allow for low-cost solutions while maintaining reasonable problem size.

Generating executions
- Generating routes
- Choosing routes
- Dynamic requests
Greedy Randomized Procedure

Based on the trip insertion heuristic of Campbell & Savelsbergh (2004) for the VRP with time windows and worker shift constraints

Introduce randomization in the route generation phase to produce a richer set of routes.

But how about exact solution methods?
Branch-and-price solution method

- Generating executions
- Generating routes
- Choosing routes
- Dynamic requests

**Solve restricted master problem**
- Solve linear relaxation problem with initial route set

**Check for routes:**
- Solve pricing subproblem

- **Add routes**
  - \( R \): negative cost routes found
  - \( NR \): no negative cost routes found

**Branch or fathom**
Search for integer solutions
Outline

• Motivation
• Background
• Modeling and solution methods
  – Generating executions (VR)
  – Generating routes (GRP)
  – Choosing routes (B&P)
  – Dynamic requests
• Conclusion
Multiple Choice Elementary
Constrained Shortest Path Problem

- Generating executions
  - Generating routes
  - Choosing routes
  - Dynamic requests

Find the shortest \( (c_{ij}) \) path from node 0 to node 5, such that

- The total length \( (t_{ij}) \) of the solution path cannot exceed set limit
- Time window \([a_i, b_i]\) at a node is observed
- Partition nodes into subsets, \( \{N_I, N_{II}, \ldots \} \); visit at most one node in each subset

\( (t_{ij}, c_{ij}) \): arc length and cost

\([a_i, b_i]\): node time window
### Solution approaches

<table>
<thead>
<tr>
<th></th>
<th>Multiple choice constraints</th>
<th>Elementary constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MC-ECSPP</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Adapted K-cycle: Smilowitz (2006)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Two-phase method</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Aggregated bounding subproblem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Lower bound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Conservative upper bound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- One-node upper bound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Expansion subproblem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Generating executions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Generating routes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Choosing routes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Dynamic requests</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Two-phase method

- Phase I: Aggregate nodes within a subset
  - Lower bound: use best combination of parameters

Solve an elementary constrained shortest path problem:
Remove complexity of multiple choice
Two-phase method

Phase II: Expand aggregated solution path to obtain feasible solution

Note: graph is acyclic
Remove complexity of elementary problem
Embed within branch-and-price

- Generating executions
  - Generating routes
  - **Choosing routes**
  - Dynamic requests

Solve restricted master problem
Solve linear relaxation with initial route set \( R' \)

\[ duals \]

Check LB solution:
Solve lower bound aggregation problem

Check for routes:
Solve lower bound expansion and/or one-phase upper bound

Add routes
Add new routes to \( R' \)

Branch or fathom

\( R: \) negative cost routes found
\( NR: \) no negative cost routes found

Check for routes:
Solve full MC-ECSPP
Outline

• Motivation
• Background
• Modeling and solution methods
  – Generating executions (VR)
  – Generating routes (GRP; MC-ECSPP)
  – Choosing routes (B&P)
  – Dynamic requests
• Conclusion
Current research

- Dynamic Drayage Vehicle Routing Problem
  - It is typical that 60% of task requests are known before the day of operation and the remaining 40% become known during the day (Grosz 2003).

- Difficulties we are facing:
  - Potential problem size and complexity
  - High percentage of uncertainty
  - Large number of scenarios

• Generating executions
• Generating routes
• Choosing routes

- Dynamic requests
How to model the dynamic MRRP

- At each decision epoch,
  - Known requested tasks
  - Expected future tasks
- Partition network into subregions
  - Aggregate cross-region tasks
  - Estimate the number of expected tasks for each scenario
- Recourse function for solution $x$ for each scenario

- Generating executions
- Generating routes
- Choosing routes

Dynamic requests
Set Partitioning Formulation

\( \mathcal{T} \): Set of known static tasks

\( \mathcal{T}_e \): Set of expected dynamic tasks across subregions

\( \bar{m}_j \): Expected number of dynamic tasks of type \( j \in \mathcal{T}_e \) among all scenarios

\( \delta_j \): The extra number of dynamic tasks of type \( j \in \mathcal{T}_e \) for robustness

\( Q(x, s) \): The cost of recourse given \( x \) and one realization of scenario \( s \in \mathcal{S} \)

\[
\min \left\{ \sum_{r \in \mathcal{R}} c_r x_r + E_s[Q(x, s)] \right\}
\]

minimize cost of all routes

subject to

\[
\sum_{r \in \mathcal{R}} \sum_{e \in \mathcal{E}_i} a_{re} x_r = 1 \quad \forall i \in \mathcal{T}
\]

cover 1 movement for static task

\[
\sum_{r \in \mathcal{R}} a_{rj} x_r \geq \bar{m}_j + \delta_j \quad \forall j \in \mathcal{T}_e
\]

cover dynamic tasks

\[
x = (x_r) \in \{0, 1\} \quad \forall r \in \mathcal{R}
\]

binary decision

- Generating executions
- Generating routes
- Choosing routes
- Dynamic requests
Several areas to be explored

- How can we estimate the set of expected tasks? How can we evaluate the solutions for different scenarios?

- How frequently should the solutions be updated? How can the solutions be recoursed with new information?

- How to evaluate the new solution method compared to re-optimization?
Conclusions

Contributions

- New modeling and solution approaches for the MRRP with flexible tasks
- A new variation of the elementary constrained shortest path problem
- New approaches for the dynamic MRRP

Future plan:

- Implementation of the branch-and-price solution method with dynamic column generation
- Solution method improvements
- Further insights into strategic drayage operations
Thank you…