Freight railcar routing problem arising in Russia

Ruslan Sadykov\textsuperscript{1}  
Vitaliy Shiryaev\textsuperscript{3}

Alexander A. Lazarev\textsuperscript{2}
Alexey Stratonnikov\textsuperscript{3}

\textsuperscript{1}INRIA Bordeaux, Talence, France  
\textsuperscript{2}Institute of Control Sciences, Moscow, Russia  
\textsuperscript{3}JSC Freight One, Moscow, Russia

EURO 2013  
Rome, Italy, July 2
Contents

Problem description

Solution approaches

Numerical results and conclusions
The freight car routing problem: overview

initial car distribution

transportation demands
Specificity of freight rail transportation in Russia

- The fleet of freight railcars is owned by independent freight companies
- Forming and scheduling or trains is done by the state company
  - It charges a cost for transferring cars and determines (estimated) travel times
  - Cost for the transfer of an empty car depends on the type of previously loaded product
- Distances are large, and average freight train speed is low ($\approx 300$ km/day): discretization in periods of 1 day is reasonable
The freight car routing problem: input and output

Input

- Railroad network (stations)
- Initial locations of cars (sources)
- Transportation demands and associated profits
- Costs: transfer costs and standing (waiting) daily rates;

Output: operational plan

- A set of accepted demands and their execution dates
- Empty and loaded cars movements to meet the demands (car routing)

Objective

Maximize the total net profit
Data: overview

- $T$ — planning horizon (set of time periods);
- $I$ — set of stations;
- $C$ — set of car types;
- $K$ — set of product types;
- $Q$ — set of demands;
- $S$ — set of sources (initial car locations);
- $M$ — empty transfer cost function;
- $D$ — empty transfer duration function;
Demands data

For each order \( q \in Q \)

- \( i^1_q, i^2_q \in I \) — origin and destination stations;
- \( k_q \in K \) — product type
- \( C_q \subseteq C \) — set of car types, which can be used for this demand
- \( n^\text{max}_q \) (\( n^\text{min}_q \)) — maximum (minimum) number of cars, needed to fulfill (partially) the demand
- \( r_q \in T \) — release time of demand
- \( \Delta_q \in \mathbb{Z}_+ \) — maximum delay for starting the transportation
- \( \rho_{qt} \) — profit from delivery of one car with the product, transportation of which started at period \( t, t \in [r_q, r_q + \Delta_q] \)
- \( d_q \in \mathbb{Z}_+ \) — transportation time of the demand
- \( w^1_q(w^2_q) \) — daily standing rate charged for one car waiting before loading (after unloading) the product at origin (destination) station
Sources and car types data

For each source $s \in S$

- $\vec{i}_s \in I$ — station where cars are located
- $\vec{c}_s \in C$ — type of cars
- $\vec{r}_s \in T$ — period, starting from which cars can be used
- $\vec{w}_s$ — daily standing rate charged for cars
- $\vec{k}_s \in K$ — type of the latest delivered product
- $\vec{n}_s \in \mathbb{N}$ — number of cars in the source

For each car type $c \in C$

- $Q_c$ — set of demands, which a car of type $c$ can fulfill
- $S_c$ — set of sources for car type $c$
Contents

Problem description

Solution approaches

Numerical results and conclusions
Commodity graph

Commodity $c \in C$ represents the flow (movements) of cars of type $c$.

Graph $G_c = (V_c, A_c)$ for commodity $c \in C$:
Graph definition

- **vertex** $v_{cit}^{wk}$ — stay of cars of type $c \in C$ at station $i \in I$ at daily waiting rate $w$ at period $t \in T$, where $k \in K$ is the type of unloaded product. **Flow balance** is

$$b(v_{cit}^{wk}) = \begin{cases} \vec{n}_{s}, & \exists s \in S_c : \vec{i}_{s} = i, \vec{r}_{s} = t, \vec{w}_{s} = w, \vec{k}_{s} = k, \\ 0, & \text{otherwise.} \end{cases}$$

- **waiting arc** $a_{cit}^{wk}$ — waiting of cars of type $c \in C$ from period $t \in T$ to $t + 1$ at station $i \in I$ at daily rate $w$, $k \in K$ is the type of previously loaded product. **Cost** $g(a)$ is $w$.

- **empty transfer arc** $a_{cijt}^{w'w''k}$ — transfer of empty cars of type $c \in C$ waiting at station $i \in I$ at daily rate $w'$ to station $j \in I$ where they will wait at daily rate $w''$, such that the type of latest unloaded product is $k \in K$, and transfer starts at period $t \in T$. **Cost** is $M(c, i, j, k)$.

- **loaded transfer arc** $a_{cqt}$ — transportation of demand $q \in Q$ by cars of type $c \in C$ starting at period $t \in T \cap [r_q, r_q + \Delta_q]$. **Cost** is $-\rho_{qt}$.
Multi-commodity flow formulation

Variables

- $x_a \in \mathbb{Z}_+$ — flow size along arc $a \in A_c, c \in C$
- $y_q \in \{0, 1\}$ — demand $q \in Q$ is accepted or not

\[
\min \sum_{c \in C} \sum_{a \in A_c} g(a) x_a
\]

\[
\sum_{c \in C_q} \sum_{a \in A_{cq}} x_a \leq n_q^{\text{max}} y_q \quad \forall q \in Q
\]

\[
\sum_{c \in C_q} \sum_{a \in A_{cq}} x_a \geq n_q^{\text{min}} y_q \quad \forall q \in Q
\]

\[
\sum_{a \in \delta^-(v)} x_a - \sum_{a \in \delta^+(v)} x_a = b(v) \quad \forall c \in C, v \in V_c
\]

$0 \leq x_a \quad \forall c \in C, a \in V_c$

$0 \leq y_q \leq 1 \quad \forall q \in Q$

We concentrate on solving its LP-relaxation
Path reformulation

- \( P_s \) — set of paths (car routes) from source \( s \in S \)

Variables

- \( \lambda_s \in \mathbb{Z}_+ \) — flow size along path \( p \in P_s, s \in S \)

\[
\begin{align*}
\min \quad & \sum_{c \in C} \sum_{s \in S_c} \sum_{p \in P_s} g_p^{\text{path}} \lambda_p \\
\sum_{c \in C} \sum_{s \in S_c} \sum_{p \in P_s} \lambda_p &= \tilde{n}_s \quad \forall c \in C, s \in S_c \\
\lambda_p &\in \mathbb{Z}_+ \quad \forall c \in C, s \in S_c, p \in P_s \\
y_q &\in \{0, 1\} \quad \forall q \in Q
\end{align*}
\]
Column generation for path reformulation

- Pricing problem decomposes into shortest path problems for each source
  - **slow**: number of sources are thousands
- To accelerate, for each commodity $c \in C$, we search for a shortest path in-tree to the terminal vertex from all sources in $S_c$
  - **drawback**: some demands are severely “overcovered”, bad convergence
- We developed iterative procedure which removes covered demands and cars assigned to them, and the repeats search for a shortest path in-tree
Iterative pricing procedure for commodity $c \in C$

```
foreach demand $q \in Q_c$ do \hspace{1em} unc$\text{vcCars}_q \leftarrow n^\text{max}_q$;
foreach source $s \in S_c$ do \hspace{1em} rm$\text{Cars}_s \leftarrow \tilde{n}_s$;
iter \leftarrow 0;
repeat
\hspace{1em} Find an in-tree to the terminal from sources $s \in S_c$, rm$\text{Cars}_s > 0$;
\hspace{1em} Sort paths $p$ in this tree by non-decreasing of their reduced cost;
\hspace{1em} foreach path $p$ in this order do
\hspace{2em} if $\bar{g}_p < 0$ and unc$\text{vcCars}_q > 0$, \forall $q \in Q_p^{\text{path}}$, then
\hspace{3em} Add variable $\lambda_p$ to the restricted master;
\hspace{3em} $s \leftarrow$ the source of $p$;
\hspace{3em} rm$\text{Cars}_s \leftarrow$ rm$\text{Cars}_s - \min\{rm$\text{Cars}_s$, unc$\text{vcCars}_q\}$;
\hspace{3em} unc$\text{vcCars}_q \leftarrow$ unc$\text{vcCars}_q - \min\{rm$\text{Cars}_s$, unc$\text{vcCars}_q\}$;
\hspace{2em} iter \leftarrow iter + 1;
until unc$\text{vcCars}_q > 0$, \forall $q \in Q_c$, or rm$\text{Cars}_s > 0$, \forall $s \in S_c$, or
iter = nbPricIter;
```
Flow enumeration reformulation

- $F_c$ — set of fixed flows for commodity $c \in C$

Variables

- $\omega_f \in \{0, 1\}$ — commodity $c$ is routed accordingly to flow $f \in F_c$ or not

\[
\begin{align*}
\min & \sum_{c \in C} \sum_{f \in F_s} g_f^{\text{flow}} \omega_f \\
\sum_{c \in C_q} \sum_{f \in F_c} \sum_{a \in A_{cq}} f_a \omega_f & \leq n_q^{\text{max}} y_q \quad \forall q \in Q \\
\sum_{c \in C_q} \sum_{f \in F_c} \sum_{a \in A_{cq}} f_a \omega_f & \geq n_q^{\text{min}} y_q \quad \forall q \in Q \\
\sum_{f \in F_c} \omega_f & = 1 \quad \forall c \in C \\
\omega_p & \in \{0, 1\} \quad \forall c \in C, f \in F_c \\
y_q & \in \{0, 1\} \quad \forall q \in Q
\end{align*}
\]
Approach CGEF

- Pricing problem decomposes into minimum cost flow problem for each commodity
  - slow: very bad convergence
- “Column generation for extended formulations” (CGEF) approach: we disaggregate the pricing problem solution into arc flow variables, which are added to the master.
- The master then becomes the multi-commodity flow formulation with restricter number of arc flow variables, i.e. “improving” variables are generated dynamically

Proposition

If an arc flow variable $x$ has a negative reduced cost, there exists a pricing problem solution in which $x > 0$.
(consequence of the theorem in [S. and Vanderbeck, 13])
Contents

Problem description

Solution approaches

Numerical results and conclusions
Tested approaches

- **DIRECT**: solution of the multi-commodity flow formulation by the *Clp* LP solver
  - Problem specific solver source code modifications
  - Problem specific preprocessing is applied (not public)
  - Tested inside the company

- **COLGEN**: solution of the path reformulation by column generation (*BaPCod* library and *Cplex* LP solver)
  - Initialization of the master by “doing nothing” routes
  - Stabilization by dual prices smoothing
  - Restricted master clean-up

- **COLGENEF**: “dynamic” solution of multi-commodity flow formulation by the CGEF approach (*BaPCod* library, *Lemon* min-cost flow solver and *Cplex* LP solver)
  - Initialization of the master by all waiting arcs
  - Only trivial preprocessing is applied
First test set of real-life instances

<table>
<thead>
<tr>
<th>Instance name</th>
<th>x3</th>
<th>x3double</th>
<th>5k0711q</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of stations</td>
<td>371</td>
<td>371</td>
<td>1'900</td>
</tr>
<tr>
<td>Number of demands</td>
<td>1’684</td>
<td>3’368</td>
<td>7’424</td>
</tr>
<tr>
<td>Number of car types</td>
<td>17</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>Number of cars</td>
<td>1’013</td>
<td>1’013</td>
<td>15’008</td>
</tr>
<tr>
<td>Number of sources</td>
<td>791</td>
<td>791</td>
<td>11’215</td>
</tr>
<tr>
<td>Time horizon, days</td>
<td>37</td>
<td>74</td>
<td>35</td>
</tr>
<tr>
<td>Number of vertices, thousands</td>
<td>62</td>
<td>152</td>
<td>22</td>
</tr>
<tr>
<td>Number of arcs, thousands</td>
<td>794</td>
<td>2’846</td>
<td>1’843</td>
</tr>
<tr>
<td>Solution time for DIRECT</td>
<td>20s</td>
<td>1h34m</td>
<td>55s</td>
</tr>
<tr>
<td>Solution time for COLGEN</td>
<td>22s</td>
<td>7m53s</td>
<td>8m59s</td>
</tr>
<tr>
<td>Solution time for COLGENEF</td>
<td>3m55s</td>
<td>&gt;2h</td>
<td>43s</td>
</tr>
</tbody>
</table>
Real-life instances with larger planning horizon

1'025 stations, up to 6’800 demands, 11 car types, 12’651 cars, and 8’232 sources.
Up to $\approx$ 300 thousands nodes and 10 millions arcs.

Convergence of COLGENEF in less than 15 iterations.
About 3% of arc flow variables at the last iteration.
Conclusions

- Three approaches tested for a freight car routing problem on real-life instances
- Approach COLGEN is the best for instances with small number of sources
- Problem-specific preprocessing is important: good results for DIRECT
- Approach COLGENEF is the best for large instances
- Combination of COLGENEF and problem-specific preprocessing would allow to increase discretization and improve solutions quality
Some practical considerations are not taken into account:

- Progressive standing daily rates
- Special stations for long-time stay (with lower rates)
- Compatibility between two consecutive types of loaded products.
- Penalties for refused demands
- Groups of cars are transferred faster and for lower unitary costs.