The Empty Freight Car Distribution Problem

in collaboration with

Katharina Beygang
Summer School On Integer Programming, Auckland
14 Feb 2011
Problem Definition

**Situation:**
- Loading and unloading of freight cars in different locations
- Unloading freight cars $\rightarrow$ supply
- Loading freight cars $\rightarrow$ demand

**Problem:**
At some places: # supply $\neq$ # demand

**Aim:**
Empty freight car movements

Empty Freight Car Distribution with minimum transport costs
<table>
<thead>
<tr>
<th>Supplies / Demands / Connections</th>
</tr>
</thead>
</table>

**Supply**
- Place of supply
- Number of cars
- Disposal time

**Demand**
- Place of demand
- Number of cars
- Required time

**Suitable train connection**

(Parts of) supply can satisfy demand (in parts)
Components:

1. Supply-Level
2. Demand-Level
3. Connection-Level

G=(V,R) with
- \( V = \{ \text{supply set } S, \text{ demand set } D \} \)
- \( R = \{(s,d) \in S \times D: \text{ if } s \text{ can satisfy } d\} \)
How can we model how much cars a supply / demand contains of?

Extension of the Network Model

- Add a global sink / source, edges and capacities $u$
- Add an excess function $b: V \rightarrow \mathbb{Z}$
Consider the distribution problem as a network flow problem

- Existence of efficient algorithms
- Integer Solution when integer input data
- Taking advantage of special network structures

Modelling generates a minimum cost flow problem:

\[
\begin{align*}
\min & \sum_{(i,j) \in R} c_{i,j} \cdot x_{i,j} \\
\text{subject to} & \sum_{i: (i,j) \in R} x_{i,j} - \sum_{i: (j,i) \in R} x_{j,i} = b(j) \text{ for all } j \in A \cup B \quad (\ast) \\
& 0 \leq x_{i,j} \leq u_{i,j} \text{ for all } (i,j) \in R \quad (\ast\ast)
\end{align*}
\]

(\ast) mass balance constraints
(\ast\ast) capacity constraints
Successive-Shortest-Path Algorithm (SSP)

- Conceptual easy approach
- Pseudo-polynomial running time
- Transformation into weak polynomial algorithms using scaling techniques

Procedure:

As long as there exists a path between unbalanced source and sink do:

1. Send flow from source to sink along a shortest path in the residual network
2. Update: network
Algorithmic Approach III

Procedure:

As long as there exists a path between unbalanced source and sink do:

(1) Send flow from source to sink along a shortest path in the residual network

(2) Update: network
Reduced-costs optimality condition

A flow $\mathbf{x}$ is optimal if and only if there exist a node potential $\mathbf{p}: V \rightarrow \mathbb{R}$ with $c^\mathbf{p}(r) := c(r) + p(u) - p(v) \geq 0$ for all $r = (u,v) \in G_\mathbf{x}$.

Transforms infeasible, super-optimal solution into feasible optimal solution

Shortest path computation w.r.t. positive reduced costs...Why?

- Shortest path computation with arbitrary costs: $O(nm) \subseteq O(n^3)$
- Shortest path computation with non-negative costs: $O(n \log n + m)$
Interpretation

„How can we transform the min cost flow into a feasible distribution?“

Every unit of flow in connection-level corresponds to a car movement

Problem Definition
Example
Supplies / Demands
Network Model
Algorithmic Approach
The Reality
The Reality

- European Distribution
- Car Types
- Substitution
- Priorization of demands
- Warmstarts

Continuity of the freight car distribution process
Car Types and Substitution

**Background:** Different car types

Given: different car types

"One car of type A can substitute two car of type B"

Use multiplier $\mu : R \to [0, \infty[$ on the edges

Restriction of the connection-level

"One car of type A can substitute one car of type B"

**Generalized Min Cost Flow Problem**

$$\min \sum_{(i,j) \in R} c_{i,j} \cdot x_{i,j}$$

$$\sum_{i: (i,j) \in R} \mu_{i,j} x_{i,j} - \sum_{i: (j,i) \in R} x_{j,i} = b(j) \text{ for all } j \in A \cup B$$

$$0 \leq x_{i,j} \leq u_{i,j} \text{ for all } (i, j) \in R$$

$$0 \leq \mu_{i,j} < \infty \text{ for all } (i, j) \in R$$
**Theorem** (Garey et al):

The generalized flow problem is NP-hard to solve.

- Remains NP-hard if we just allow multiplier 1 and 2.
- Optimal solutions with fractional values can occur.
- Existence of good heuristics.

But in series-parallel graphs, we can solve it in:

- Continuous Case \(\rightarrow\) Polynomial time
- Integral Case \(\rightarrow\) Pseudo-polynomial time
Further Requirements

- European Distribution  ➔  Huge input set
- Car Types  ➔  Restriction of the connection-level
- Substitution  ➔  Generalized Flow Problem
- Priorization of demands
- Warmstarts
- ...

Continuous freight car distribution process
Priorization of Demands

**Background:** Lack of empty freight cars

**Weak Priorization**
- No guarantee of satisfaction
- Via cost terms

**Strong Priorization**
- Guarantee of as much satisfaction as possible
- Not via cost terms

**Modified network modell and algorithmic approach**
Priorization of Demands: Network Modell

**Idea:** use two sinks

<table>
<thead>
<tr>
<th>i</th>
<th>s_i</th>
<th>d_i</th>
<th>Priority of demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>strong</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>weak</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>weak</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>strong</td>
</tr>
</tbody>
</table>

Problem Definition
Example
Supplies / Demands
Network Model
Algorithmic Approach
The Reality
**Priorization of Demands: Network Modell**

**Idea**: use two sinks

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>s_i</th>
<th>d_i</th>
<th>Priority of demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>strong</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>weak</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>weak</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>strong</td>
</tr>
</tbody>
</table>

![Network Diagram](image-url)
Priorization of Demands: Algorithmic Approach

**Procedure:**

**Strong Priorization**

As long as there exists a path between unbalanced source and sink „strong“ do:

1. Send flow from source to sink „strong“ along a shortest path in the residual network
2. Update: network

**Weak Priorization**

As long as there exists a path between unbalanced source and sink „weak“ do:

1. Send flow from source to sink „weak“ along a shortest path in the residual network
2. Update: network
Further Requirements

- European Distribution → Huge input set
- Car Types → Restriction of the connection-level
- Substitution → Generalized Flow Problem
- Priorization of demands → Modified SSP
- Warmstarts → Techniques from SSP
- ... → Continuity of the freight car distribution process

Problem Definition
Example
Supplies / Demands
Network Model
Algorithmic Approach
The Reality