Routing Cars
in Rail Freight Service

Armin Fügenschuh - Zuse Institute Berlin, Germany

Henning Homfeld - Technische Universität Darmstadt, Germany
Alexander Martin - Technische Universität Darmstadt, Germany
Hanno Schülldorf - Technische Universität Darmstadt, Germany
& Deutsche Bahn AG, Frankfurt am Main, Germany

January 7, 2010
Project Partners

• Deutsche Bahn (G. Pfau, J. Wolfner)
  • Department for Company Development (GSU1)
  • Interested in long-term traffic & demand forecast & simulation
  • Code testing & result benchmarking

• Schenker (A. Below, C. Liebchen, T. Klingler)
  • DB subsidiary for freight service
  • Solving the operational routing problem manually
  • Providing real-life data

• Federal Ministry for Science and Education (BMBF)
  • Funding applied math projects (period 2007-2010)
  • OVERSYS:
    • U. Zimmermann, R. Hansmann (TU Braunschweig)
    • U. Clausen, A. Chmielewski, J. Baudach (Fraunhofer IML)
    • C. Helmberg, F. Fischer (TU Chemnitz)
    • R. Schultz (U Duisburg), G. Reinelt (U Heidelberg)
Introduction

• Railway freight transport has a market share of 20%.
• 100,000 Mil. ton km, of which:
  • 45% inland traffic,
  • 45% cross-border traffic,
  • 10% transit traffic.
• Deutsche Bahn offers whole trains (~30 cars) and individual cars.
• Several individual cars with different destinations are grouped to trains at classification yards.
• At the next classification yard, the cars are re-grouped, until they reached their destinations.
• **Main question:** what is the „best“ path for each car?
Facts and Figures

- Railway network length: 38,200 km
- 5000 trains per day, 150,000 cars
- Terminal stations: 2,200
- Classification yards:
  - Large („Rangierbahnhöfe“): 11
  - Medium („Knotenbahnhöfe“): 30
  - Small („Satellitenbahnhöfe“): 200
Classification Yards
Classification Yards

• Disintegration of trains
• Sorting the cars (with the help of gravity)
• Assembling of new trains
Survey of the Literature

• First models for the blocking problem emerged in the 1960s. Since then...
  • Bodin, Schuster & Golden (1980)
  • Assad (1983)
  • Crainic, Ferland & Rousseau (1984)
    Crainic & Rousseau (1986)
  • Keaton (1989, 1992)
  • Newton (1997)
    Newton, Barnhart & Vance (1998)
    Barnhart, Hong & Vance (2000)
  • Ahuja (2007)

• Main differences to our problem:
  • Special constraints due to DB operational rules („Leitwege“).
  • Costs are induced by trains (and not so much by cars).
The Bundling Effect

- Cost induced by cars
The Bundling Effect

- Cost induced by cars
The Bundling Effect

- Cost induced by cars
The Bundling Effect

- Cost induced by cars

![Diagram of the bundling effect with numbers indicating costs and connections between nodes.](image)
The Bundling Effect

- Cost induced by trains
The Bundling Effect

- Cost induced by trains
The Bundling Effect

• Cost induced by trains
The Bundling Effect

• Cost induced by trains
The Bundling Effect

- Cost induced by trains
The Bundling Effect

- Cost induced by trains

![Diagram showing cost induced by trains with values: 100, 10, 30, 30, 40, 60, 100, 130, 190]
The Bundling Effect

- Cost induced by trains
The Bundling Effect

- Cost induced by trains
Modes of Operation

• Three ways of sending cars from origin to destination:
  • Individual car routing
    • Assign a sequence of yards to each car
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing

Diagram showing a network of routes with distances and costs. Nodes and edges represent different points and connections in the network.
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Example: Individual Car Routing
Modes of Operation

• Three ways of sending cars from origin to destination:
  • Individual car routing
    • Assign a sequence of yards to each car
  • Blocking of cars
    • Assign a sequence of yards to each order
    • All cars in an order follow the same path through the network
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars

![Diagram of cars blocking a path](image)
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars
Example: Blocking of Cars

Diagram showing a network with nodes and connections, including numbers like 100, 200, 300, and 1200.
Example: Blocking of Cars

Diagram showing a network with various nodes and connections, indicating blocking of cars at different points with numerical values.
Example: Blocking of Cars
Example: Blocking of Cars
Modes of Operation

- Three ways of sending cars from origin to destination:
  - Individual car routing
    - Assign a sequence of yards to each car
  - Blocking of cars
    - Assign a sequence of yards to each order
    - All cars in an order follow the same path through the network
  - "Leitwege" - DB car routing
    - For all orders with the same destination and each yard assign a successor
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System

Diagram showing a network with various nodes and connections, indicating values such as 100, 200, 300, 500.
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System

Zuse Institute Berlin
Dep. Optimization
Armin Fügenschuh
January 7, 2010
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Example: DB „Leitwege“ System
Modes of Operation

• Three ways of sending cars from origin to destination:
  • Individual car routing
    • Assign a sequence of yards to each car
  • Blocking of cars
    • Assign a sequence of yards to each order
    • All cars in an order follow the same path through the network
  • „Leitwege“ - DB car routing
    • For all orders with the same destination and each yard assign a successor

• If „Leitwege“ are so expensive, then why do they do it?
  • Historical reasons
  • More robust / errors can easily be corrected
The Arc-Flow Model
The Arc-Flow Model

* Sets: stations $V$, precedence relations $A := V \times V$, orders $K$. 
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$. 
The Arc-Flow Model

- Sets: stations \( V \), precedence relations \( A := V \times V \), orders \( K \).
- Variables: car routes \( x_{i,j}^k \in \mathbb{B} \), sorting tracks \( y_{i,j} \in \mathbb{N} \), trains \( n_{i,j} \in \mathbb{N} \).
- Constraints
The Arc-Flow Model

- Sets: stations \( V \), precedence relations \( A := V \times V \), orders \( K \).
- Variables: car routes \( x^k_{i,j} \in \mathbb{B} \), sorting tracks \( y_{i,j} \in \mathbb{N} \), trains \( n_{i,j} \in \mathbb{N} \).
- Constraints
  - Orders:
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Orders:
    - Delivery: $\forall i \in V, k \in K : \sum_{j : (i,j) \in A} x_{i,j}^k - \sum_{j : (j,i) \in A} x_{j,i}^k = \begin{cases} 1, & i = o(k), \\ -1, & i = d(k), \\ 0, & \text{else.} \end{cases}$
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in B$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Orders:
    - Delivery: $\forall i \in V, k \in K : \sum_{j:(i,j)\in A} x_{i,j}^k - \sum_{j:(j,i)\in A} x_{j,i}^k = \begin{cases} 1, & i = o(k), \\ -1, & i = d(k), \\ 0, & \text{else.} \end{cases}$
  - Time limit: $\forall k \in K : \sum_{(i,j)\in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k$
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Orders:
    - Delivery: $\forall i \in V, k \in K : \sum_{j : (i,j) \in A} x_{i,j}^k - \sum_{j : (j,i) \in A} x_{j,i}^k = \begin{cases} 1, & i = o(k), \\ -1, & i = d(k), \\ 0, & \text{else}. \end{cases}$
  - Time limit: $\forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k$
  - Yards:
The Arc-Flow Model

• Sets: stations \( V \), precedence relations \( A := V \times V \), orders \( K \).
• Variables: car routes \( x_{i,j}^k \in \mathbb{B} \), sorting tracks \( y_{i,j} \in \mathbb{N} \), trains \( n_{i,j} \in \mathbb{N} \).
• Constraints
  • Orders:
    • Delivery: \( \forall i \in V, k \in K : \sum_{j:(i,j) \in A} x_{i,j}^k - \sum_{j:(j,i) \in A} x_{j,i}^k = \begin{cases} 1, & i = o(k), \\ -1, & i = d(k), \\ 0, & \text{else}. \end{cases} \)
  • Time limit: \( \forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k \)
  • Yards:
    • Number of sorting tracks: \( \forall i \in V : \sum_{j:(i,j) \in A} y_{i,j} \leq Y_i \)
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x^k_{i,j} \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Orders:
    - Delivery: $\forall i \in V, k \in K : \sum_{j: (i,j) \in A} x^k_{i,j} - \sum_{j: (j,i) \in A} x^k_{j,i} = \begin{cases} 1, & i = o(k), \\ -1, & i = d(k), \\ 0, & \text{else.} \end{cases}$
  - Time limit: $\forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x^k_{i,j} \leq T_k$
- Yards:
  - Number of sorting tracks: $\forall i \in V : \sum_{j: (i,j) \in A} y_{i,j} \leq Y_i$
  - Number of trains per track: $\forall (i,j) \in A : n_{i,j} \leq 6y_{i,j}$
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Orders:
    - Delivery: $\forall i \in V, k \in K : \sum_{j: (i,j) \in A} x_{i,j}^k - \sum_{j: (j,i) \in A} x_{j,i}^k = \begin{cases} 1, & i = o(k), \\ -1, & i = d(k), \\ 0, & \text{else}. \end{cases}$
  - Time limit: $\forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k$
- Yards:
  - Number of sorting tracks: $\forall i \in V : \sum_{j: (i,j) \in A} y_{i,j} \leq Y_i$
  - Number of trains per track: $\forall (i, j) \in A : n_{i,j} \leq 6y_{i,j}$
  - Hump capacity: $\forall i \in V : \sum_{k \in K, j: (i,j) \in A} v_k \cdot x_{i,j}^k \leq H_i$
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in B$, sorting tracks $y_{i,j} \in N$, trains $n_{i,j} \in N$.
- Constraints
  - Trains:
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Trains:
    - Max. length: $\forall (i, j) \in A : \sum_{k \in K} l_k \cdot x_{i,j}^k \leq 700 \cdot n_{i,j}$
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x^k_{i,j} \in B$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Trains:
    - Max. length: $\forall (i, j) \in A : \sum_{k \in K} l_k \cdot x^k_{i,j} \leq 700 \cdot n_{i,j}$
    - Max. weight: $\forall (i, j) \in A : \sum_{k \in K} w_k \cdot x^k_{i,j} \leq 1600 \cdot n_{i,j}$
The Arc-Flow Model

• Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
• Variables: car routes $x^k_{i,j} \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
• Constraints
  • Trains:
    • Max. length: $\forall (i, j) \in A : \sum_{k \in K} l_k \cdot x^k_{i,j} \leq 700 \cdot n_{i,j}$
    • Max. weight: $\forall (i, j) \in A : \sum_{k \in K} w_k \cdot x^k_{i,j} \leq 1600 \cdot n_{i,j}$
  • „Leitweg“ (unique successor) constraint („plain“):
    $\forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x^k_{i,j_1} + x^l_{i,j_2} \leq 1$
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x^k_{i,j} \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Trains:
    - Max. length: $\forall (i, j) \in A : \sum_{k \in K} l_k \cdot x^k_{i,j} \leq 700 \cdot n_{i,j}$
    - Max. weight: $\forall (i, j) \in A : \sum_{k \in K} w_k \cdot x^k_{i,j} \leq 1600 \cdot n_{i,j}$
  - „Leitweg“ (unique successor) constraint („plain“):
    $\forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x^k_{i,j_1} + x^l_{i,j_2} \leq 1$
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x_{i,j}^k \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Trains:
    - Max. length: $\forall (i, j) \in A : \sum_{k \in K} l_k \cdot x_{i,j}^k \leq 700 \cdot n_{i,j}$
    - Max. weight: $\forall (i, j) \in A : \sum_{k \in K} w_k \cdot x_{i,j}^k \leq 1600 \cdot n_{i,j}$
  - „Leitweg“ (unique successor) constraint („plain“):
    $\forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x_{i,j_1}^k + x_{i,j_2}^l \leq 1$
The Arc-Flow Model

- Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
- Variables: car routes $x^k_{i,j} \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$.
- Constraints
  - Trains:
    - Max. length: $\forall (i, j) \in A : \sum_{k \in K} l_k \cdot x^k_{i,j} \leq 700 \cdot n_{i,j}$
    - Max. weight: $\forall (i, j) \in A : \sum_{k \in K} w_k \cdot x^k_{i,j} \leq 1600 \cdot n_{i,j}$
  - „Leitweg“ (unique successor) constraint („plain“):
    $\forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x^k_{i,j_1} + x^l_{i,j_2} \leq 1$
The Arc-Flow Model

• Sets: stations $V$, precedence relations $A := V \times V$, orders $K$.
• Variables: car routes $x_{i,j}^k \in \mathbb{B}$, sorting tracks $y_{i,j} \in \mathbb{N}$, trains $n_{i,j} \in \mathbb{N}$. 
The Arc-Flow Model

- Sets: stations \( V \), precedence relations \( A := V \times V \), orders \( K \).
- Variables: car routes \( x_{i,j}^k \in \mathbb{B} \), sorting tracks \( y_{i,j} \in \mathbb{N} \), trains \( n_{i,j} \in \mathbb{N} \).
- Objective: \( C_1 \gg C_2 \gg C_3 \)

\[
C_1 \cdot \sum_{(i,j) \in A} n_{i,j} + C_2 \cdot \sum_{(i,j) \in A} y_{i,j} + C_3 \cdot \sum_{k \in K, (i,j) \in A} x_{i,j}^k \rightarrow \min
\]
Improving the Arc-Flow Model (1)
Improving the Arc-Flow Model (1)

• Strengthening of „Leitweg“-constraints
Improving the Arc-Flow Model (1)

- Strengthening of „Leitweg“-constraints
  - Model formulation:
Improving the Arc-Flow Model (1)

- Strengthening of „Leitweg“-constraints

  - Model formulation:
    \[
    \forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x_{i,j_1}^k + x_{i,j_2}^l \leq 1
    \]
Improving the Arc-Flow Model (1)

• Strengthening of "Leitweg"-constraints

• Model formulation:
  \[ \forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x^k_{i,j_1} + x^l_{i,j_2} \leq 1 \]
Improving the Arc-Flow Model (1)

• Strengthening of „Leitweg“-constraints

  • Model formulation:

    \[ \forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x_{i,j_1}^k + x_{i,j_2}^l \leq 1 \]

• Improved formulation („lifted“):

![Graph diagram showing flow constraints]
Improving the Arc-Flow Model (1)

- Strengthening of „Leitweg“-constraints

- Model formulation:
  \[ \forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x_{i,j_1}^k + x_{i,j_2}^l \leq 1 \]

- Improved formulation („lifted“):
Improving the Arc-Flow Model (1)

- Strengthening of „Leitweg“-constraints
  - Model formulation:
    \[ \forall k, l \in K, d(k) = d(l), i \in V, (i, j_1) \neq (i, j_2) \in A : x^k_{i,j_1} + x^l_{i,j_2} \leq 1 \]

- Improved formulation („lifted“):
  \[ \forall k, l \in K, d(k) = d(l), i \in V, V_1 \cup V_2 = V : \sum_{j_1 \in V_1} x^k_{i,j_1} + \sum_{j_2 \in V_2} x^l_{i,j_2} \leq 1 \]
Improving the Arc-Flow Model (2)

(communicated by A. Bley, 2008)
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

• Consider the structure of a feasible solution:
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

• Consider the structure of a feasible solution:
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

• Consider the structure of a feasible solution:

![Diagram of a network with nodes and arcs]
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

• Consider the structure of a feasible solution:
• Consider the structure of a feasible solution:

• The „Leitweg“-constraint (uniqueness of successor) leads to trees.
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

• Consider the structure of a feasible solution:

  ![Graph](image)

• The “Leitweg“-constraint (uniqueness of successor) leads to trees.
• Formulate trees directly and merge them with the model:
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

• Consider the structure of a feasible solution:

![Diagram of a tree structure]

• The „Leitweg“-constraint (uniqueness of successor) leads to trees.
• Formulate trees directly and merge them with the model:
  • Variable $z_{i,j}^\kappa \in \mathbb{B}$ for all $(i, j) \in A$ and $\kappa \in \mathcal{K} := \{d(k) : k \in K\}$. 
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

• Consider the structure of a feasible solution:

• The „Leitweg“-constraint (uniqueness of successor) leads to trees.
• Formulate trees directly and merge them with the model:
  • Variable \( z_{i,j}^\kappa \in \mathbb{B} \) for all \((i, j) \in A\) and \(\kappa \in \mathcal{K} := \{d(k) : k \in K\}\).
  • Unique successor: \( \forall i \in V, \kappa \in \mathcal{K} : \sum_{j: (i, j) \in A} z_{i,j}^\kappa \leq 1 \)
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

- Consider the structure of a feasible solution:
- The "Leitweg"-constraint (uniqueness of successor) leads to trees.
- Formulate trees directly and merge them with the model:
  - Variable $z_{i,j}^\kappa \in \mathbb{B}$ for all $(i, j) \in A$ and $\kappa \in \mathcal{K} := \{d(k) : k \in K\}$.
  - Unique successor: $\forall i \in V, \kappa \in \mathcal{K} : \sum_{j : (i,j) \in A} z_{i,j}^\kappa \leq 1$
  - Cars follow trees: $\forall (i, j) \in A, k \in K : x_{i,j}^k \leq z_{i,j}^{d(k)}$
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

• Consider the structure of a feasible solution:

• The „Leitweg“-constraint (uniqueness of successor) leads to trees.

• Formulate trees directly and merge them with the model:
  • Variable $z_{i,j}^{\kappa} \in \mathbb{B}$ for all $(i, j) \in A$ and $\kappa \in \mathcal{K} := \{d(k) : k \in K\}$.
  • Unique successor: $\forall i \in V, \kappa \in \mathcal{K} : \sum_{j : (i, j) \in A} z_{i,j}^{\kappa} \leq 1$
  • Cars follow trees: $\forall (i, j) \in A, k \in K : x_{i,j}^{k} \leq z_{i,j}^{d(k)}$

• Cons: more variables
• Consider the structure of a feasible solution:

![Diagram of a tree structure](image)

• The „Leitweg“-constraint (uniqueness of successor) leads to trees.

• Formulate trees directly and merge them with the model:
  • Variable $z_{i,j}^\kappa \in \mathbb{B}$ for all $(i, j) \in A$ and $\kappa \in \mathcal{K} := \{d(k) : k \in K\}$.
  • Unique successor: $\forall i \in V, \kappa \in \mathcal{K} : \sum_{j: (i,j) \in A} z_{i,j}^\kappa \leq 1$
  • Cons: more variables
  • Pros:

\[
x_k^{i,j} \leq z_{i,j}^{d(k)}
\]
Improving the Arc-Flow Model (2)
(communicated by A. Bley, 2008)

- Consider the structure of a feasible solution:

  - The „Leitweg“-constraint (uniqueness of successor) leads to trees.
  - Formulate trees directly and merge them with the model:
    - Variable $z_{i,j}^\kappa \in B$ for all $(i, j) \in A$ and $\kappa \in \mathcal{K} := \{d(k) : k \in K\}$.
    - Unique successor: $\forall i \in V, \kappa \in \mathcal{K} : \sum_{j : (i, j) \in A} z_{i,j}^\kappa \leq 1$
    - Cars follow trees: $\forall (i, j) \in A, k \in K : x_{i,j}^k \leq z_{i,j}^{d(k)}$
  - Cons: more variables
  - Pros:
    - Do not need „Leitweg“-constraint in $x$-variables.
• Consider the structure of a feasible solution:

• The „Leitweg“-constraint (uniqueness of successor) leads to trees.

• Formulate trees directly and merge them with the model:
  • Variable \( z^\kappa_{i,j} \in \mathbb{B} \) for all \((i, j) \in A\) and \( \kappa \in \mathcal{K} := \{d(k) : k \in K\} \).
  • Unique successor: \( \forall i \in V, \kappa \in \mathcal{K} : \sum_{j : (i, j) \in A} z^\kappa_{i,j} \leq 1 \)
  • Cars follow trees: \( \forall (i, j) \in A, k \in K : x^k_{i,j} \leq z^d_{i,j} \)

• Cons: more variables

• Pros:
  • Do not need „Leitweg“-constraint in \( x \)-variables.
  • Better LP-relaxation.
Heuristic Cuts: Hierarchy Constraints
Heuristic Cuts: Hierarchy Constraints

• Observation: Large yards „attract“ cars, because
  • They are at central locations in the network,
  • They can handle more cars (capacity & speed),
  • They can be connected to more yards.
Heuristic Cuts: Hierarchy Constraints

• Observation: Large yards „attract“ cars, because
  • They are at central locations in the network,
  • They can handle more cars (capacity & speed),
  • They can be connected to more yards.
• Hence a „typical“ path has the following structure:
Heuristic Cuts: Hierarchy Constraints

• Observation: Large yards „attract“ cars, because
  • They are at central locations in the network,
  • They can handle more cars (capacity & speed),
  • They can be connected to more yards.
• Hence a „typical“ path has the following structure:
Heuristic Cuts: Hierarchy Constraints

• Observation: Large yards „attract“ cars, because
  • They are at central locations in the network,
  • They can handle more cars (capacity & speed),
  • They can be connected to more yards.
• Hence a „typical“ path has the following structure:
Heuristic Cuts: Hierarchy Constraints

• Observation: Large yards „attract“ cars, because
  • They are at central locations in the network,
  • They can handle more cars (capacity & speed),
  • They can be connected to more yards.

• Hence a „typical“ path has the following structure:
Heuristic Cuts: Hierarchy Constraints

• Observation: Large yards „attract“ cars, because
  • They are at central locations in the network,
  • They can handle more cars (capacity & speed),
  • They can be connected to more yards.
• Hence a „typical“ path has the following structure:
Heuristic Cuts: Hierarchy Constraints

- Observation: Large yards „attract“ cars, because
  - They are at central locations in the network,
  - They can handle more cars (capacity & speed),
  - They can be connected to more yards.
- Hence a „typical“ path has the following structure:
Heuristic Cuts: Hierarchy Constraints

- Observation: Large yards „attract“ cars, because
  - They are at central locations in the network,
  - They can handle more cars (capacity & speed),
  - They can be connected to more yards.
- The following path is considered as „untypical“:
Heuristic Cuts: Hierarchy Constraints

- Observation: Large yards „attract“ cars, because
  - They are at central locations in the network,
  - They can handle more cars (capacity & speed),
  - They can be connected to more yards.
- The following path is considered as „untypical“:
Heuristic Cuts: Hierarchy Constraints

• Observation: Large yards „attract“ cars, because
  • They are at central locations in the network,
  • They can handle more cars (capacity & speed),
  • They can be connected to more yards.
Heuristic Cuts: Hierarchy Constraints

• Observation: Large yards „attract“ cars, because
  • They are at central locations in the network,
  • They can handle more cars (capacity & speed),
  • They can be connected to more yards.
Heuristic Cuts: Hierarchy Constraints

- Observation: Large yards „attract“ cars, because
  - They are at central locations in the network,
  - They can handle more cars (capacity & speed),
  - They can be connected to more yards.

- Analysis of historical data shows: 98% of all car paths are „monotone“.
Heuristic Cuts: Hierarchy Constraints

- Observation: Large yards „attract“ cars, because
  - They are at central locations in the network,
  - They can handle more cars (capacity & speed),
  - They can be connected to more yards.

- Analysis of historical data shows: 98% of all car paths are „monotone“.

- Separate „monotone“ paths from „zig-zag“ paths by new constraints:

\[
\forall j \in V, l \in K : \sum_{i: (i,j) \in A, h_i \leq h_j} x^l_{i,j} + \sum_{k: (j,k) \in A, h_j \geq h_k} x^l_{j,k} \leq 1
\]
Refining the Model: Turnover Times
Refining the Model: Turnover Times

• Consider again the time-limit constraint:

\[ \forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x^k_{i,j} \leq T_k \]
Refining the Model: Turnover Times

- Consider again the time-limit constraint:
  \[ \forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k \]
- Note: The turnover time \( u_i \) is a constant here.
Refining the Model: Turnover Times

• Consider again the time-limit constraint:

\[
\forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k
\]

• Note: The turnover time \( u_i \) is a constant here.

• In reality, the turnover time depends on the number of trains that are assembled on the resp. sorting track:
Refining the Model: Turnover Times

- Consider again the time-limit constraint:
  \[ \forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k \]

- Note: The turnover time \( u_i \) is a constant here.
- In reality, the turnover time depends on the number of trains that are assembled on the resp. sorting track:
Refining the Model: Turnover Times

- Consider again the time-limit constraint:
\[
\forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k
\]

- Note: The turnover time \( u_i \) is a constant here.

- In reality, the turnover time depends on the number of trains that are assembled on the resp. sorting track:

\[
u_{i,j}^k \in \mathbb{R}_+ \quad \text{is a variable, and} \quad u_{i,j}^k = \frac{24}{n_{i,j}} \cdot x_{i,j}^k.
\]
Refining the Model: Turnover Times

- Consider again the time-limit constraint:
  \[
  \forall k \in K : \sum_{(i,j) \in A} (u_i + t_{i,j}) \cdot x_{i,j}^k \leq T_k
  \]

- Note: The turnover time \( u_i \) is a constant here.

- In reality, the turnover time depends on the number of trains that are assembled on the resp. sorting track:

\[ u_{i,j}^k \in \mathbb{R}_+ \text{ is a variable, and } u_{i,j}^k = \frac{24}{n_{i,j}} \cdot x_{i,j}^k. \]

- Problem: We now have an MINLP.
Linearization 1
Linearization 1

• The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:
Linearization 1

• The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:
Linearization 1

• The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:

• Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}_+\).
Linearization 1

- The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:
- Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}^+\).
- Thus the convex hull is
Linearization 1

- The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:
- Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}^+\).
- Thus the convex hull is
  \[
  24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N
  \]
Linearization 1

- The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:

- Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}_+\).

- Thus the convex hull is

\[
24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N
\]

\[
24x \leq Nu
\]
Linearization 1

- The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:
- Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}^+\).
- Thus the convex hull is
  \[
  24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N
  
  24x \leq Nu
  \]
Linearization 1

- The set of feasible points $(u, n, x)$ with $u \cdot n = 24 \cdot x$ is:
- Remember that $x \in \{0, 1\}$ and $n \in \mathbb{Z}_+$.
- Thus the convex hull is
  
  \[24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N\]
  
  \[24x \leq Nu\]
  
  \[w \leq 24x\]
Linearization 1

• The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:
• Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}_+\).
• Thus the convex hull is
\[
24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N
\]
\[
24x \leq Nu
\]
\[
w \leq 24x
\]
Linearization 1

• The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:

• Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}_+\).

• Thus the convex hull is

\[
24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N \\
24x \leq Nu \\
w \leq 24x \\
Nu \leq 24N + 24x - 24n
\]
Linearization 1

- The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:
- Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}_+\).
- Thus the convex hull is:

\[
24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N
\]

\[
24x \leq Nu
\]

\[
w \leq 24x
\]

\[
Nu \leq 24N + 24x - 24n
\]

\[
x \leq 1
\]
Linearization 1

- The set of feasible points \((u, n, x)\) with \(u \cdot n = 24 \cdot x\) is:
- Remember that \(x \in \{0, 1\}\) and \(n \in \mathbb{Z}_+\).
- Thus the convex hull is
  \[
  24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N
  \]
  \[
  24x \leq Nu
  \]
  \[
  w \leq 24x
  \]
  \[
  Nu \leq 24N + 24x - 24n
  \]
  \[
  x \leq 1
  \]
Linearization 1

• The set of feasible points \( (u, n, x) \) with \( u \cdot n = 24 \cdot x \) is:

• Remember that \( x \in \{0, 1\} \) and \( n \in \mathbb{Z}_+ \).

• Thus the convex hull is

\[
24(2i + 1)x - 24n \leq i(i + 1)u, \quad i = 1, \ldots, N
\]

\[
24x \leq Nu
\]

\[
w \leq 24x
\]

\[
Nu \leq 24N + 24x - 24n
\]

\[
x \leq 1
\]

• Note that all lower-bound-cuts contain the origin, like „projective cuts“, c.f.
  • Frangioni, Gentile (2006, 2009)
  • Frangioni, Gentile, Grande, Pacifici (2009)
  • Günlük, Linderoth (2009)
Linearization 2

- Consider the nonlinear relation $u \cdot n = 24 \cdot x$, 
Linearization 2

• Consider the nonlinear relation $u \cdot n = 24 \cdot x$, and make it even „more“ nonlinear: $u \cdot n = 24 \cdot x^2$. 
Linearization 2

- Consider the nonlinear relation $u \cdot n = 24 \cdot x$, and make it even „more“ nonlinear: $u \cdot n = 24 \cdot x^2$. 

$u \cdot n = 24 \cdot x$
Linearization 2

- Consider the nonlinear relation \( u \cdot n = 24 \cdot x \), and make it even „more“ nonlinear: \( u \cdot n = 24 \cdot x^2 \).

\[
\begin{align*}
  u \cdot n &= 24 \cdot x \\
  u \cdot n &= 24 \cdot x^1
\end{align*}
\]
Linearization 2

- Consider the nonlinear relation \( u \cdot n = 24 \cdot x \), and make it even "more" nonlinear: \( u \cdot n = 24 \cdot x^2 \).
Linearization 2

- Consider the nonlinear relation $u \cdot n = 24 \cdot x$, and make it even „more“ nonlinear: $u \cdot n = 24 \cdot x^2$. 

$u \cdot n = 24 \cdot x$  

$u \cdot n = 24 \cdot x^2$
Linearization 2

- Consider the nonlinear relation $u \cdot n = 24 \cdot x$
  and make it even "more" nonlinear: $u \cdot n = 24 \cdot x^2$.

In the sequel we are interested in a lower bound on the waiting time:

$$u \cdot n \geq 24 \cdot x^2.$$
Linearization 2 (cont.)
• On $u \cdot n \geq 24 \cdot x^2$ we apply the following variable substitution

\[
\begin{align*}
u &= \tau + \alpha, \\
n &= \tau - \alpha, \\
24 \cdot x^2 &= \beta^2.
\end{align*}
\]
• On $u \cdot n \geq 24 \cdot x^2$ we apply the following variable substitution

\[
\begin{align*}
    u &= \tau + \alpha, \\
    n &= \tau - \alpha,
\end{align*}
\]

\[
24 \cdot x^2 = \beta^2.
\]
Linearization 2 (cont.)

• On $u \cdot n \geq 24 \cdot x^2$ we apply the following variable substitution

\[
\begin{align*}
u &= \tau + \alpha, \\
n &= \tau - \alpha, \\
24 \cdot x^2 &= \beta^2.
\end{align*}
\]
Linearization 2 (cont.)

• On $u \cdot n \geq 24 \cdot x^2$ we apply the following variable substitution

\[
\begin{align*}
u &= \tau + \alpha, \\
n &= \tau - \alpha, \\
24 \cdot x^2 &= \beta^2.
\end{align*}
\]
Linearization 2 (cont.)

- On $u \cdot n \geq 24 \cdot x^2$ we apply the following variable substitution
  
  $u = \tau + \alpha,$
  
  $n = \tau - \alpha,$
  
  $24 \cdot x^2 = \beta^2.$
Linearization 2 (cont.)

• On $u \cdot n \geq 24 \cdot x^2$ we apply the following variable substitution

$$u = \tau + \alpha,$$
$$n = \tau - \alpha,$$

$$24 \cdot x^2 = \beta^2.$$

• We obtain the Lorentz (second order) cone

$$\sqrt{\alpha^2 + \beta^2} \leq \tau.$$
Linear Approximation of the SOC
Linear Approximation of the SOC

• The SOC can be approximated by linear inequalities (Ben-Tal, Nemirovski 1998, Glineur 2000):
Linear Approximation of the SOC

- The SOC can be approximated by linear inequalities (Ben-Tal, Nemirovski 1998, Glineur 2000):

\[
\begin{align*}
\xi^0 & \geq |\alpha| \\
\eta^0 & \geq |\beta| \\
\xi_i &= \cos\left(\frac{\pi}{2i+1}\right)\xi_{i-1} + \sin\left(\frac{\pi}{2i+1}\right)\eta_{i-1} \\
\eta_i &\geq \left|\sin\left(\frac{\pi}{2i+1}\right)\xi_{i-1} + \cos\left(\frac{\pi}{2i+1}\right)\eta_{i-1}\right| \\
\xi_n &\leq \tau \\
\eta_n &\leq \tan\left(\frac{\pi}{2n+1}\right)\xi_n
\end{align*}
\]

\[i = 2, 3, \ldots, M\]
Linear Approximation of the SOC

- The SOC can be approximated by linear inequalities (Ben-Tal, Nemirovski 1998, Glineur 2000):

\[
\begin{align*}
\xi^0 & \geq |\alpha| \\
\eta^0 & \geq |\beta| \\
\xi_i &= \cos\left(\frac{\pi}{2i+1}\right)\xi_{i-1} + \sin\left(\frac{\pi}{2i+1}\right)\eta_{i-1} \\
\eta_i &\geq \left| - \sin\left(\frac{\pi}{2i+1}\right)\xi_{i-1} + \cos\left(\frac{\pi}{2i+1}\right)\eta_{i-1} \right| \\
\xi_n &\leq \tau \\
\eta_n &\leq \tan\left(\frac{\pi}{2n+1}\right)\xi_n
\end{align*}
\]

with accuracy

\[
\varepsilon(M) = \frac{1}{\cos\left(\frac{\pi}{2M+1}\right)} - 1 = O\left(\frac{1}{4^M}\right)
\]
Linear Approximation of the SOC

• The SOC can be approximated by linear inequalities (Ben-Tal, Nemirovski 1998, Glineur 2000):

\[
\begin{align*}
\xi^0 &\geq |\alpha| \\
\eta^0 &\geq |\beta| \\
\xi_i &= \cos\left(\frac{\pi}{2^{i+1}}\right)\xi_{i-1} + \sin\left(\frac{\pi}{2^{i+1}}\right)\eta_{i-1} \\
\eta_i &\geq \left| - \sin\left(\frac{\pi}{2^{i+1}}\right)\xi_{i-1} + \cos\left(\frac{\pi}{2^{i+1}}\right)\eta_{i-1} \right| \\
\xi_n &\leq \tau \\
\eta_n &\leq \tan\left(\frac{\pi}{2^{n+1}}\right)\xi_n \\
\end{align*}
\]

with accuracy

\[\epsilon(M) = \frac{1}{\cos\left(\frac{\pi}{2^{M+1}}\right)} - 1 = O\left(\frac{1}{4^M}\right)\]

• Good news: We are back in the MILP world!
Computational Results (1)

- Comparison of formulations on random instances
- CPLEX 11.2, default settings

<table>
<thead>
<tr>
<th>instance</th>
<th>plain</th>
<th>lifted</th>
<th>tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>338</td>
<td>47</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>956</td>
<td>136</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>7208</td>
<td>3019</td>
<td>6095</td>
</tr>
<tr>
<td>4</td>
<td>3422</td>
<td>796</td>
<td>190</td>
</tr>
<tr>
<td>5</td>
<td>83</td>
<td>43</td>
<td>34</td>
</tr>
</tbody>
</table>

- CPU times:

<table>
<thead>
<tr>
<th>instance</th>
<th>plain</th>
<th>lifted</th>
<th>tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>557</td>
<td>572</td>
<td>628</td>
</tr>
<tr>
<td>2</td>
<td>1347</td>
<td>1811</td>
<td>1428</td>
</tr>
<tr>
<td>3</td>
<td>63893</td>
<td>37806</td>
<td>52388</td>
</tr>
<tr>
<td>4</td>
<td>19555</td>
<td>21740</td>
<td>5454</td>
</tr>
<tr>
<td>5</td>
<td>8376</td>
<td>3832</td>
<td>1884</td>
</tr>
</tbody>
</table>

- B&B nodes:
Computational Results (2)

- National traffic in Germany
- 43 yards (medium and large)
- 1,600 orders
- CPLEX10
- 6 hours CPU time
- 8% gap
- Visible bundling effects - most traffic on main axes
Computational Results (3)

- Cross-border traffic to France (real-world instance)
- 26 yards, ~250 orders
- CPLEX 11.1
- 8 processors, 32 GByte RAM
- 1 hours CPU time
- Best feasible solution: 89,100
- With only small gap (2.58%)
Thank you for your attention!