Bus Scheduling and Multicommodity Flows

Ralf Borndörfer

2015 Workshop on Combinatorial Optimization with Applications in Transportation and Logistics

Beijing, 28.07.2015
Outline

- Optimal Assignments
- Single Depot Vehicle Scheduling
- Multiple Depot Vehicle Scheduling
- Lagrangean Relaxation
- Multicriteria Optimization
The Assignment Problem

The Problem

- Input: 3 sources, 3 destinations, costs
- Output: cost minimal assignment

The solution:

\[
\begin{array}{cccc}
1 & 2 & 3 \\
3 & 4 & 3 \\
3 & 7 & 6 \\
7 & 7 & 8 \\
8 & 10 & \\
\end{array}
\]

Cost = 20
A Heuristic

The Greedy Heuristic

- heuretikos (gr.): inventive
- heuriskein (gr.): to find

solution cost = 17
A Heuristic

The Greedy Heuristic

- heuretikos (gr.): inventive
- heuriskein (gr.): to find

solution
cost = 16
A Lower Bound

A "Relaxation"

bound

cost = 13

solution

cost = 17

guarantee

4/17 = 23%

4/13 = 30%
A Proof of Optimality

The "primal problem"
- Minimum cost assignment

The "dual problem"
- Maximum profit sales

optimum cost = 15
An Exact Algorithm

The "successive shortest path" algorithm
An Exact Algorithm

- The "successive shortest path" algorithm

Bound Costs: 0
Partial Sol. Costs: 0
An Exact Algorithm

- The "successive shortest path" algorithm

**Bound Costs:** 10
**Partial Sol. Costs:** 3
An Exact Algorithm

- The "successive shortest path" algorithm

Bound Costs: 10
Partial Sol. Costs: 3
The "successive shortest path" algorithm
An Exact Algorithm

- The "successive shortest path" algorithm

Bound Costs: 10
Partial Sol.
Costs: 6
An Exact Algorithm

The "successive shortest path" algorithm
The "successive shortest path" algorithm
An Exact Algorithm

The "successive shortest path" algorithm computes a shortest path for every source node, i.e., does n shortest path path calculations.

Bound Costs: 15
Solution Costs: 15
Guaranty: 0% (Optimal)
An Exact Algorithm

Theorem: The assignment problem can be solved in polynomial time of $O(n^3)$. 

Bound Costs: 15
Solution Costs: 15
Guaranty: 0% (Optimal)
Algebraic Model

Graphen theoretic model

Algebraic Model
"Integer Program"

min 3x_{11} + 3x_{12} + 4x_{13} + 3x_{21} + 7x_{22} + 6x_{23} + 7x_{31} + 8x_{32} + 10x_{33}

s.t. \begin{align*}
x_{11} + x_{12} + x_{13} &= 1 \\
x_{21} + x_{22} + x_{23} &= 1 \\
x_{31} + x_{32} + x_{33} &= 1 \\
x_{11} + x_{21} + x_{31} &= 1 \\
x_{12} + x_{22} + x_{32} &= 1 \\
x_{13} + x_{23} + x_{33} &= 1 \\
x_{11}, \ldots, x_{33} &\geq 0 \\
x_{11}, \ldots, x_{33} &\in \{0, 1\} \end{align*}
LP-Relaxation

Integer Program

\[
\begin{align*}
\text{min} & \quad 3x_{11} + 3x_{12} + 4x_{13} \\
& + 3x_{21} + 7x_{22} + 6x_{23} \\
& + 7x_{31} + 8x_{32} + 10x_{33} \\
\text{s.t.} & \quad x_{11} + x_{12} + x_{13} = 1 \\
& \quad x_{21} + x_{22} + x_{23} = 1 \\
& \quad x_{31} + x_{32} + x_{33} = 1 \\
& \quad x_{11} + x_{21} + x_{31} = 1 \\
& \quad x_{12} + x_{22} + x_{32} = 1 \\
& \quad x_{13} + x_{23} + x_{33} = 1 \\
& \quad x_{11}, \ldots, x_{33} \geq 0 \\
& \quad x_{11}, \ldots, x_{33} \in \{0,1\}
\end{align*}
\]

Linear Program

"LP-Relaxation" (here: integer)

\[
\begin{align*}
\text{min} & \quad 3x_{11} + 3x_{12} + 4x_{13} \\
& + 3x_{21} + 7x_{22} + 6x_{23} \\
& + 7x_{31} + 8x_{32} + 10x_{33} \\
\text{s.t.} & \quad x_{11} + x_{12} + x_{13} = 1 \\
& \quad x_{21} + x_{22} + x_{23} = 1 \\
& \quad x_{31} + x_{32} + x_{33} = 1 \\
& \quad x_{11} + x_{21} + x_{31} = 1 \\
& \quad x_{12} + x_{22} + x_{32} = 1 \\
& \quad x_{13} + x_{23} + x_{33} = 1 \\
& \quad x_{11}, \ldots, x_{33} \geq 0 \\
& \quad x_{11}, \ldots, x_{33} \leq 1
\end{align*}
\]
LP-Relaxation

Linear Program
"LP-Relaxation"

\[
\begin{align*}
\text{min} & \quad 3x_{11} + 3x_{12} + 4x_{13} \\
& \quad + 3x_{21} + 7x_{22} + 6x_{23} \\
& \quad + 7x_{31} + 8x_{32} + 10x_{33} \\
\text{s.t.} & \quad x_{11} + x_{12} + x_{13} = 1 \\
& \quad x_{21} + x_{22} + x_{23} = 1 \\
& \quad x_{31} + x_{32} + x_{33} = 1 \\
& \quad x_{11} + x_{21} + x_{31} = 1 \\
& \quad x_{12} + x_{22} + x_{32} = 1 \\
& \quad x_{13} + x_{23} + x_{33} = 1 \\
& \quad x_{11}, \ldots, x_{33} \geq 0 \\
& \quad x_{11}, \ldots, x_{33} \leq 1 
\end{align*}
\]

Eliminate \( x_{11} \)

Eliminate \( x_{12}, x_{13}, x_{21}, x_{31} \)

\[
\begin{align*}
\text{min} & \quad 3(1 - x_{12} - x_{13}) + 3x_{12} + 4x_{13} \\
& \quad + 3x_{21} + 7x_{22} + 6x_{23} \\
& \quad + 7x_{31} + 8x_{32} + 10x_{33} \\
\text{s.t.} & \quad 1 - x_{12} - x_{13} - x_{11} = 1 \\
& \quad x_{21} + x_{22} + x_{23} = 1 \\
& \quad x_{31} + x_{32} + x_{33} = 1 \\
& \quad 1 - x_{12} - x_{13} + x_{21} + x_{31} = 1 \\
& \quad x_{12} + x_{22} + x_{32} = 1 \\
& \quad x_{13} + x_{23} + x_{33} = 1 \\
& \quad 1 - x_{12} - x_{13} - x_{11} + \ldots + x \geq 0 \\
& \quad 1 - x_{12} - x_{13} - x_{11} - \ldots + x \leq 1 
\end{align*}
\]
LP-Relaxation

**Linear Program "LP-Relaxation"**

\[
\begin{align*}
\text{min} & \quad 4x_{22} + 2x_{23} + x_{32} + 2x_{33} + 14 \\
\text{s.t.} & \quad x_{22}, x_{23}, x_{32}, x_{33} \\
& \quad x_{22} + x_{23} + x_{32} + x_{33} \geq 1 \\
& \quad x_{22} + x_{23} \leq 1 \\
& \quad x_{32} + x_{33} \leq 1 \\
& \quad x_{22}, x_{23}, x_{32}, x_{33} \geq 0
\end{align*}
\]

**"Polyhedron"**
Mathematical Models

\[ x_{32} = 1 \]
\[ x_{21} = 1 \]
\[ x_{13} = 1 \]
Linear Programming

Min $x_1 + 2x_2$

$x_1 + x_2 \geq 2$

$x_1 - x_2 \leq 1$

$x_2 \leq 3$

$x_1 \geq 0$

$x_2 \geq 0$

Simplex Algorithm
Hardware

<table>
<thead>
<tr>
<th>Old Computer</th>
<th>New Computer</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun 3/50</td>
<td>Pentium 4, 1.7 GHz</td>
<td>800</td>
</tr>
<tr>
<td>Sun 3/50</td>
<td>Compaq Server ES 40, 667 MHz</td>
<td>900</td>
</tr>
<tr>
<td>Intel 386, 25 MHz</td>
<td>Compaq Server ES 40, 667 MHz</td>
<td>400</td>
</tr>
<tr>
<td>IBM 3090/108S</td>
<td>Compaq Server ES 40, 667 MHz</td>
<td>45</td>
</tr>
</tbody>
</table>

Software

<table>
<thead>
<tr>
<th>Old Code</th>
<th>New Code</th>
<th>Estimated Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMP</td>
<td>Cplex 1.0</td>
<td>4.7</td>
</tr>
<tr>
<td>Cplex 1.0</td>
<td>Cplex 5.0</td>
<td>22.0</td>
</tr>
<tr>
<td>Cplex 5.0</td>
<td>Cplex 7.1</td>
<td>3.7</td>
</tr>
<tr>
<td>XMP</td>
<td>Cplex 7.1</td>
<td>960</td>
</tr>
</tbody>
</table>

"A Model that might have taken a year to solve 10 years ago, can now solve in less than 10 seconds."
Linear Programming

Min $x_1 + 2x_2$

$x_1 + x_2 \geq 2$
$x_1 - x_2 \leq 1$
$x_2 \leq 3$
$x_1 \geq 0$
$x_2 \geq 0$

Polyhedron

Simplex Algorithm
Integer Programming

Min $x_1 + 2x_2$

$x_1 + x_2 \geq 2$
$x_1 - x_2 \leq 1$
$x_2 \leq 3$
$x_1 \geq 0$
$x_2 \geq 0$
$x_1, x_2$ integer

Branch-and-Bound

Polyhedron
Integer Programming

Min \( x_1 + 2x_2 \)

\[
\begin{align*}
    x_1 + x_2 & \geq 2 \\
    x_1 - x_2 & \leq 1 \\
    x_2 & \leq 3 \\
    x_1 & \geq 0 \\
    x_2 & \geq 0
\end{align*}
\]

Cutting Planes

Polyhedron
MIP-Speedup 1991-2010
(Bixby, Lecture on Mixed-Integer Programming, TU Berlin, 20.01.2010)

Cumulative Speedup

<table>
<thead>
<tr>
<th>Version to Version Speedup</th>
<th>V-V Speedup</th>
<th>Cumulative Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mature Dual Simplex: 1994</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1→3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3→4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4→5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5→6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6→6.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.5→7.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.1→8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8→9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9→10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10→11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mined Theoretical Backlog: 1998

Cumulative Speedup: 29530x
Sea Freight
(Koopmans [1965], 7 sources, 7 sinks, all sea links)

Efficient graph of ballast traffic
Based on world dry cargo movements in 1913.
Figures at ports: Net surplus of empty ships.
Figures along routes: Optimal flows of ballast traffic.
All figures in millions of metric tons per month.
Optimal Allocation of Scarce Resources
(Nobel Price in Economics 1975)

Leonid V. Kantorovich

Tjalling C. Koopmans
ECONOMIC POTENTIAL FUNCTION OF THE LOCATION OF A SHIP

Based on optimal routing of ballast traffic for 1913.
Figures at ports: Economic potential of the appearance of a ship.
Figures along routes: Sailing time in months required to traverse the route in ballast.
Planning Problems in Public Transit

Service Design

Operational Planning

Operations Control
too short to turn

too long to wait

best choice
### bus scheduling and multicommodity flows | CoTL 2015

**Camel Curve**

```
<table>
<thead>
<tr>
<th>Linien</th>
<th>Uhrzeit</th>
<th>SB</th>
<th>SL</th>
<th>Taxi</th>
<th>SIM</th>
<th>SGM</th>
<th>SI</th>
</tr>
</thead>
<tbody>
<tr>
<td>V20</td>
<td>04:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>05:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>102</td>
<td>06:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>103</td>
<td>07:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>104</td>
<td>08:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>105</td>
<td>09:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>106</td>
<td>10:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>107</td>
<td>11:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>119</td>
<td>12:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120</td>
<td>13:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>121N</td>
<td>04:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>122N</td>
<td>05:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>123N</td>
<td>06:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>124N</td>
<td>07:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125N</td>
<td>08:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>126N</td>
<td>09:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>127N</td>
<td>10:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>128N</td>
<td>11:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>129N</td>
<td>12:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

**Diagram:**

- **Legend:**
  - SB
  - SL
  - Taxi
  - SIM
  - SGM
  - SI

- **Color Codes:**
  - Green: SB
  - Blue: SL
  - Red: Taxi
  - Orange: SIM
  - Yellow: SGM
  - White: SI

**Note:**

- The diagram represents the distribution of traffic density over time and space, with color-coded areas indicating the volume of traffic for different modes of transportation. The data is collected and analyzed to optimize bus scheduling and improve traffic flow.
"Camel Curve"
(Ario, Böhring & Mojsilovic [1980])
Assignment Approach
(Single Depot Vehicle Scheduling)
Single Depot Vehicle Scheduling
(Assignment Approach)
Vehicle Scheduling Problem

- **Input**
  - Timetabled and deadhead trips
  - Vehicle types and depot capacities
  - Vehicle costs (fixed and variable)

- **Output**
  - Vehicle rotations

- **Problem**
  - Compute rotations to cover all timetabled trips

- **Goals**
  - Minimize number of vehicles
  - Minimize operation costs
  - Minimize line hopping etc.
Multicommodity Flow Model
Vehicle Scheduling (VS-OPT)
Integer Programming Model
(Multi-Commodity Flow Problem)

\[
\begin{align*}
\text{min} & \quad \sum \sum c_{dij} x_{dij} \\
\sum x_{dij} - \sum x_{djk} & = 0 \quad \forall j, d \quad \text{Vehicle Flow} \\
\sum \sum x_{dij} - \sum \sum x_{djk} & = 0 \quad \forall j \quad \text{Aggregate Flow} \\
\sum x_{dij} & = 1 \quad \forall j \quad \text{Trips} \\
\sum x_{dij} & \leq \kappa_d \quad \forall d \quad \text{ Capacities} \\
x_{dij} & \in \{0,1\} \quad \forall ij, d \quad \text{Integrality}
\end{align*}
\]
Observation: The LP relaxation of the Multicommodity Flow Problem is in general not integer.

Theorem: The Multicommodity Flow Problem is NP-hard.

Theorem (Tardos et. al.): There are pseudo-polynomial time approximation algorithms to solve the LP-relaxation of Multicommodity Flow Problems which are faster than general LP methods.
Integer Programming Model
(Multi-Commodity Flow Problem)

\[
\min \sum_{d} \sum_{ij} c_{ij}^{d} x_{ij}^{d}
\]

\[
\sum_{i} x_{ij}^{d} - \sum_{k} x_{jk}^{d} = 0 \quad \forall j, d \quad \text{Vehicle Flow}
\]

\[
\sum_{d} \sum_{i} x_{ij}^{d} - \sum_{d} \sum_{k} x_{jk}^{d} = 0 \quad \forall j \quad \text{Aggregate Flow}
\]

\[
\sum_{d} x_{ij}^{d} = 1 \quad \forall j \quad \text{Trips}
\]

\[
\sum_{d} x_{dj}^{d} \leq \kappa^{d} \quad \forall d \quad \text{Capacities}
\]

\[
x_{ij}^{d} \in \{0,1\} \quad \forall ij, d \quad \text{Integrality}
\]
Lagrangean Relaxation
(Subproblem is a Min Cost Flow Problem)

\[
\max \min_{\pi} \sum_{d} \sum_{ij} c_{ij}^{d} x_{ij}^{d} - \sum_{j,d} \pi_{j}^{d} \left( \sum_{i} x_{ij}^{d} - \sum_{i} x_{ji}^{d} \right)
\]

\[
\sum_{d} \sum_{i} x_{ij}^{d} - \sum_{d} \sum_{i} x_{ji}^{d} = 0 \quad \forall j \quad \text{Agg. Flow}
\]

\[
\sum_{d} x_{ij}^{d} = 1 \quad \forall j \quad \text{Trips}
\]

\[
\sum_{j} x_{dj}^{d} \leq \kappa^{d} \quad \forall d \quad \text{Capacities}
\]

\[
x_{ij}^{d} \in \{0,1\} \quad \forall ij,d \quad \text{Binary}
\]

Subproblem: Min-Cost Flow
Lagrangean Relaxation

\[
\begin{align*}
\min \quad & c^T x \\
A x & = b \\
B x & = d \\
x & \geq 0
\end{align*}
\]

\[
= \max_{\lambda} \quad f(\lambda)
\]

\[
\begin{align*}
\max \min \quad & c^T x + \lambda(b - Ax) \\
B x & = d \\
x & \geq 0
\end{align*}
\]

\[
= \max_{\lambda} \min_i \quad c^T x_i + \lambda(b - A x_i)
\]
Integer Programming Model
(Multi-Commodity Flow Problem)

\[
\sum \sum c_{ij}^d x_{ij}^d
\]

\[
\sum x_{ij}^d - \sum x_{jk}^d = 0 \quad \forall j, d \quad \text{Vehicle Flow}
\]

\[
\sum \sum x_{ij}^d - \sum \sum x_{jk}^d = 0 \quad \forall j \quad \text{Aggregate Flow}
\]

\[
\sum x_{ij}^d = 1 \quad \forall j \quad \text{Trips}
\]

\[
\sum x_{dj}^d \leq \kappa^d \quad \forall d \quad \text{Capacities}
\]

\[
x_{ij}^d \in \{0,1\} \quad \forall ij, d \quad \text{Integrality}
\]
Lagrangean Relaxation
(Subproblem is a Min Cost Flow Problem)

\[
\max_{\pi} \min \sum_{d} \sum_{ij} c_{ij}^d x_{ij}^d - \sum_{j,d} \pi_j^d \left( \sum_i x_{ij}^d - \sum_i x_{ji}^d \right)
\]

\[
\sum_{d} \sum_{i} x_{ij}^d - \sum_{d} \sum_{i} x_{ji}^d = 0 \quad \forall j \quad \text{Agg. Flow}
\]

\[
\sum_{d} x_{ij}^d = 1 \quad \forall j \quad \text{Trips}
\]

\[
\sum_{j} x_{dj}^d \leq \kappa^d \quad \forall d \quad \text{Capacities}
\]

\[
x_{ij}^d \in \{0,1\} \quad \forall ij, d \quad \text{Binary}
\]

Subproblem: Min-Cost Flow
Lagrangean Relaxation
(Subproblem is a Min Cost Flow Problem)
Network Flows

Delbert Ray Fulkerson

Lester Randolph Ford Jr.
Military Logistics
(Ford & Fulkerson [1955], Schrijver [2002])
MCF - A network simplex implementation.

Current Release
Version 1.3

Author
Andreas Löbel

References
SPEC CPU 2006: 429 mcf
SPEC CPU 2006: 181 mcf, a well investigated program.
The MCFClass Project by Antonio Frangioni

Documentation
README and README changes
mcf.ps, mcf.pdf, or mcf.dvi

Supported Platforms
Gnu make (e.g., Linux/Unix/Cygwin)
MS Visual C++ 6.0 (Windows)

License Conditions
Free of charge for academic use (ZIB ACADEMIC LICENSE)
Commercial use requires an individual license agreement!

Download
Gnu make binary packages (tgz): Linux, SunOS, and Cygwin
MSVC++ 6.0 binary packages: tgz, zip, and self-extracting .exe
Source packages: tgz, zip, and self-extracting .exe

Sites for other min cost flow solvers
ZIP elib
Hans D. Mittelmann's site

Disclaimer of warranty: No warranty whatsoever is given for the content of external links. The content of linked pages is solely the responsibility of the providers of these pages.
### CINT2006 (Integer Component of SPEC CPU2006):

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Language</th>
<th>Application Area</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400.perbench</td>
<td>C</td>
<td>Programming Language</td>
<td>Derived from Perl V5 &amp; 7. The workload includes SpamAssassin, MHonArc (an email indexer), and specdiff (SPEC’s tool that checks benchmark outputs).</td>
</tr>
<tr>
<td>401.bzfp2</td>
<td>C</td>
<td>Compression</td>
<td>Julian Seward’s bzfp2 version 1.0.3, modified to do most work in memory, rather than doing I/O.</td>
</tr>
<tr>
<td>403.gcc</td>
<td>C</td>
<td>C Compiler</td>
<td>Based on gcc Version 3.2, generates code for Opteron.</td>
</tr>
<tr>
<td>429.mcf</td>
<td>C</td>
<td>Combinatorial Optimization</td>
<td>Vehicle scheduling. Uses a network simplex algorithm (which is also used in commercial products) to schedule public transport.</td>
</tr>
<tr>
<td>445.gobmk</td>
<td>C</td>
<td>Artificial Intelligence: Go</td>
<td>Plays the game of Go, a simply described but deeply complex game.</td>
</tr>
<tr>
<td>456.hmmer</td>
<td>C</td>
<td>Search Gene Sequence</td>
<td>Protein sequence analysis using profile hidden Markov models (profile HMMs).</td>
</tr>
<tr>
<td>458.sjeng</td>
<td>C</td>
<td>Artificial Intelligence: chess</td>
<td>A highly-ranked chess program that also plays several chess variants.</td>
</tr>
<tr>
<td>464.h264ref</td>
<td>C</td>
<td>Video Compression</td>
<td>A reference implementation of H.264/AVC, encodes a videostream using 2 parameter sets. The H.264/AVC standard is expected to replace MPEG2.</td>
</tr>
<tr>
<td>471.omnetpp</td>
<td>C++</td>
<td>Discrete Event Simulation</td>
<td>Uses the OMNet++ discrete event simulator to model a large Ethernet campus network.</td>
</tr>
<tr>
<td>473.astar</td>
<td>C++</td>
<td>Path-finding Algorithms</td>
<td>Pathfinding library for 2D maps, including the well known A* algorithm.</td>
</tr>
<tr>
<td>483.xalanbmk</td>
<td>C++</td>
<td>XML Processing</td>
<td>A modified version of Xalan-C++, which transforms XML documents to other document types.</td>
</tr>
</tbody>
</table>
Lagrangean Pricing Algorithm
(Löbel [1997])
\begin{align*}
\min & \quad \sum_{d} \sum_{ij} c_{ij}^d x_{ij}^d \\
\sum x_{ij}^d - \sum x_{jk}^d & = 0 \quad \forall j, d \quad \text{Vehicle flow} \\
\sum_{d} \sum_{i} x_{ij}^d - \sum_{d} \sum_{k} x_{jk}^d & = 0 \quad \forall j \quad \text{Aggregated flow} \\
\sum_{d} \sum_{i} x_{ij}^d & = 1 \quad \forall j \quad \text{Timetabled trips} \\
\sum_{j} x_{0j}^d & \leq \kappa_d \quad \forall d \quad \text{Depot capacities} \\
x_{ij}^d & \in \{0,1\} \quad \forall ij, d \quad \text{Deadhead trips}
\end{align*}
Subproblem: Several independent Min-Cost-Flows (single-depot)
Heuristics

Cluster First – Schedule Second
- "Nearest-depot" heuristic
- Lagrange Relaxation II + tie breaker

Schedule First – Cluster Second
- Lagrange relaxation I

Schedule – Cluster – Reschedule
- Schedule: Lagrange relaxation I
- Cluster: Look at paths
- Solve a final min-cost flow

Plus tabu search
IVU.plan / IVU.crew / IVU.vehicle
System Overview with Optimization Modules

IVU.plan
- timetable
- stops
- routes

Integrated Planning
- BS-OPT
- VS-OPT
- IS-OPT
- DS-OPT

IVU.crew
- vehicle workings
- duties

IVU.vehicle
- vehicle roster
- duty roster

Optimization Modules
- R-OPT
- AFD
- APD
- WS-OPT

Depot Management
- payroll accounting
Systematischer Einsatz


BVG (Berlin)

Auf Sparkurs zum Ziel

Das Berliner Busnetz kreuzt jährlich mehr als 300.000 Kilometer. Mit Hilfe moderner Software könnte man auf geringe Zuschüsse verzichten.

BVG (Berlin)

Bus Scheduling and Multicommodity Flows | COεTL 2015

62
## Urban Scenarios

<table>
<thead>
<tr>
<th></th>
<th>BVG</th>
<th>HHA</th>
<th>VHH</th>
</tr>
</thead>
<tbody>
<tr>
<td>depots</td>
<td>10</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>vehicle types</td>
<td>44</td>
<td>40</td>
<td>19</td>
</tr>
<tr>
<td>timetabled trips</td>
<td>25 000</td>
<td>16 000</td>
<td>5 500</td>
</tr>
<tr>
<td>deadheads</td>
<td>70 000 000</td>
<td>15 100 000</td>
<td>10 000 000</td>
</tr>
<tr>
<td>cpu mins</td>
<td>200</td>
<td>50</td>
<td>28</td>
</tr>
</tbody>
</table>
**Integer Programming Model**

*(Multi-Commodity Flow Problem)*

\[
\min \sum_{d} \sum_{ij} c_{ij}^{d} x_{ij}^{d}
\]

\[
\sum_{i} x_{ij}^{d} - \sum_{k} x_{jk}^{d} = 0 \quad \forall j, d \quad \text{Vehicle Flow}
\]

\[
\sum_{d} \sum_{i} x_{ij}^{d} - \sum_{d} \sum_{k} x_{jk}^{d} = 0 \quad \forall j \quad \text{Aggregate Flow}
\]

\[
\sum_{d} x_{ij}^{d} = 1 \quad \forall j \quad \text{Trips}
\]

\[
\sum_{j} x_{dj}^{d} \leq \kappa_{j}^{d} \quad \forall d \quad \text{Capacities}
\]

\[
x_{ij}^{d} \in \{0, 1\} \quad \forall ij, d \quad \text{Integrality}
\]
Lagrangean Relaxation

\[
\begin{align*}
\min & \quad c^T x \\
A x & = b \\
B x & = d \\
x & \geq 0 \\
\end{align*}
\]

\[
\begin{align*}
= & \quad \max_{\lambda} f(\lambda) \\
= & \quad \max_{\lambda} \min_i c^T x_i + \lambda(b - A x_i)
\end{align*}
\]
Subgradient Method

\[
\text{max } f(\lambda) := \min_{x \in X} c^T x + \lambda^T (b - Ax)
\]

\[X = \text{conv } \{x_\mu\} \text{ polyhedral (piecewise linear)}\]

\[
\bar{f}_\mu(\lambda) = c^T x_\mu + \lambda^T (b - Ax_\mu)
\]

\[
\hat{f}_k(\lambda) := \min_{\mu \in J_k} \bar{f}_\mu(\lambda)
\]

\[
\lambda_{k+1} := \lambda_k + u_k (b - Ax_{\mu_k}) \quad \text{subgradient}
\]
Bundle Method
(Kiwieli [1990], Helmberg [2000])

$$\max \quad f(\lambda) := \min_{x \in X} c^T x + \lambda^T (b - Ax)$$

$$X = \text{conv} \{x_{\mu}\} \text{ polyhedral (piecewise linear)}$$

$$\bar{f}_\mu(\lambda) = c^T x_{\mu} + \lambda^T (b - Ax_{\mu})$$

$$\hat{f}_k(\lambda) := \min_{\mu \in J_k} \bar{f}_\mu(\lambda)$$

$$\lambda_{k+1} = \arg\max_{\lambda} \hat{f}_k(\lambda) - \frac{u_k}{2} \|\lambda - \hat{\lambda}_k\|^2$$
Theorem:

\[
\lambda_{k+1} = \hat{\lambda}_k + \frac{1}{u} \sum_{\mu \in J_k} \alpha_\mu (b - A x_\mu)
\]

\[
\tilde{x}_{k+1} = \sum_{\mu \in J_k} \alpha_\mu x_\mu
\]

\[
\|b - \tilde{A}_k\|_\infty \rightarrow 0 \quad (k \rightarrow \infty)
\]

\[
\Rightarrow (\tilde{x}_k)_{k \in \mathbb{N}} \quad \text{converges to a point} \quad \tilde{x} \in \{ x : A x = b, x \in X \} \]
Quadratic Subproblem

(1) \[ \max \hat{f}_k(\lambda) - \frac{u_k}{2} \| \lambda - \hat{\lambda}_k \|^2 \]

(2) \[ \max v - \frac{u_k}{2} \| \lambda - \hat{\lambda}_k \|^2 \]
\text{s.t.} \quad v \leq \bar{f}_\mu(\lambda), \text{ for all } \mu \in J_k

(3) \[ \min \sum_{\mu \in J_k} \alpha_\mu \bar{f}_\mu(\hat{\lambda}) - \frac{1}{2u_k} \left\| \sum_{\mu \in J_k} \alpha_\mu (b - Ax_\mu) \right\|^2 \]
\text{s.t.} \quad \sum_{\mu \in J_k} \alpha_\mu = 1
\quad 0 \leq \alpha_\mu \leq 1, \quad \text{for all } \mu \in J_k
Bundle Method

(IHU 41 838,500 x 3,570, 10.5 NNEs per column)

Bus Scheduling and Multicommodity Flows | COeTL 2015
Heuristic Evaluation of $f$
Thank you for your attention

Ralf Borndörfer

Freie Universität Berlin
Zuse-Institute Berlin
Takustr. 7
14195 Berlin-Dahlem

Fon (+49 30) 84185-243
Fax (+49 30) 84185-269

borndoerfer@zib.de

www.zib.de/borndoerfer