Paths and Complexity

Ralf Borndörfer

2015 Workshop on
Combinatorial Optimization with Applications in
Transportation and Logistics

Beijing, 28.07.2015
Outline

- Traffic Optimization
- Bridges of Königsberg
- Travelling Salesmen
- S-Bahn Challenge
- Shortest Paths
- Fuel Efficient Aircraft Trajectories
Optimization in Public Transit

Slide of IVU

The IVU.suite for Public Transport

for all operational requirements

Planning | Dispatching | Fleet Management | Ticketing | Passenger Information | Controlling

Trip Scheduling | Vehicle Scheduling | Integrated Scheduling

with a continuous data flow

Line Planning  
Vehicle Scheduling  
Duty Rostering

Multicom. Flow  
Set Partitioning  
Integrated Scheduling  
Other
Overview

1. Paths and Complexity
2. Vehicle Scheduling and Multicommodity Flows
3. Crew Scheduling and Column Generation
4. Track Allocation and Configurations
5. Vehicle Rotation Planning and Hyperassignments
6. Line Planning and Path Connectivity
Leonhard Euler (1707-1783)

\[ e^\pi \text{i} \sin \cos \Sigma f(x) \]
... receives a letter of Mayor Carl Leonhard Ehler

Scan by Wladimir Velminski for his thesis in the History of Arts supervised by Prof. Bredekamp (HU Berlin, ~2008)
The problem, which I am told is widely known, is as follows: in Königsberg in Prussia, there is an island A, called „the Kneiphof“; the river which surrounds it is divided into two branches, as can be seen in Fig. 1, and these branches are crossed by seven bridges, a, b, c, d, e, f und g. Concerning these bridges, it was asked whether anyone could arrange a route in such a way that he would cross each bridge once and only once. I was told that some people asserted that this impossible, while others were in doubt; but nobody would actually assert that it could be done. From this, I formulated the general problem: whatever be the arrangement and division of the river into branches, and however many bridges there be, can one find out whether or not it is possible to cross each bridge exactly once?“
The Problem of the Königsberg Bridges (1736)

"As far as the problem of the seven bridges of Königsberg is concerned, it can be solved by making an exhaustive list of all possible routes, and then finding whether or not any route satisfies the conditions of the problem. Because of the number of possibilities, this method of solution would be too difficult and laborious, and in other problems with more bridges it would be impossible. Moreover, if this method is followed to its conclusion, many irrelevant routes will be found, which is the reason for the difficulty of this method. Hence I rejected it, and looked for another method concerned only with the problem of whether or not the specified route could be found.; I considered that such a method would be much simpler."
Combinatorial Problem

**Input:** A finite (implicitly given) set $N=\{1,\ldots,n\}$, a predicate $f: N \rightarrow \{\text{true, false}\}$.

**Question:** Is there an element $i$ s.t. $f(i) = \text{true}$?
Combinatorial Explosion

Number of routes (worst case)
Here: \( \frac{7!}{2} = 7 \times 6 \times 5 \times 4 \times 3 = 2.520 \)
In general: \( = O(n^n) \)

Stirling formula: \( \frac{n!}{2} \approx n^n e^{-n} \sqrt{2\pi n} / 2 \)

a b c
ab ac ba bc ca cb
abc acb bac bca cab cba
Geometria situs (Graph Theory)

- Node
- Edge
- Degree

Paths and Complexity | COeTL 2015
Geometria situs (Graph Theory)

- Node
- Edge
- Degree

Paths and Complexity | COεTL 2015
Theorem: An Euler tour can only exist if at most 2 nodes have odd degree.

Proof. Inner nodes are even.
**Theorem:** An Euler tour exists if and only if at most 2 nodes have odd degree.

**Proof**

\( \Rightarrow\): inner nodes even

\( \Leftarrow\): path + cycles
Sir William Rowan Hamilton (1805-1865)
The Icosian Game (1856)

Icosahedron (20) Dodecahedron (12)
Is there a closed roundtrip (Hamiltonian cycle)?
Thomas Penyngton Kirkman (1806-1895)
The Cell of the Bee
Enumeration?

- Number of Hamiltonian cycles (worst case)
  
  \[ n \text{ cities: } \frac{n!}{2} \approx n^n e^{-n} \sqrt{2\pi n} / 2 \text{ (Stirling formula)} \]

- Exponential effort: \( f(n) = 2^n, n^n \), etc.

- Polynomial effort: \( f(n) = p(n) = n, 1.000n, n^3, n^5 \), etc.

<table>
<thead>
<tr>
<th>Linear</th>
<th>Quadratic</th>
<th>Cubic</th>
<th>Exponential</th>
<th>Doubly Exp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>( n^2 )</td>
<td>( n^3 )</td>
<td>( 2^n )</td>
<td>( n^n )</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>1,000</td>
<td>1,024</td>
<td>10^{10}</td>
</tr>
<tr>
<td>100</td>
<td>10,000</td>
<td>10^6</td>
<td>10^{30}</td>
<td>10^{200}</td>
</tr>
<tr>
<td>1,000</td>
<td>10^6</td>
<td>10^9</td>
<td>10^{300}</td>
<td>10^{3000}</td>
</tr>
<tr>
<td>10,000</td>
<td>10^8</td>
<td>10^{12}</td>
<td>10^{3000}</td>
<td>10^{50000}</td>
</tr>
</tbody>
</table>

- Is there a polynomial method?
The Satisfiability Problem

**Input:** A set $U=\{x_1, ..., x_n\}$ of variables and a set $C=\{c_1, ..., c_m\}$ of conjunctions (product of ANDs) of variables from $U$.

**Question:** Is there a satisfying truth assignment for $C$ (assignment of values true or false to the variables in $U$ such that in each clause at least one variable is true)?

**Example:**

$$(x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor \neg x_3)$$
The Classes NP, NPC, and NPH

**Non-deterministic polynomial time algorithm A**
Input: Instance I of problem P of size n bits
Algorithm: Guess solution L and check in time poly(n) that L solves I.

**NP:** Class of decision problems (answer yes or no) for which such an algorithm exists.

**NPC:** Class of decision problems to which NP can be reduced.

**NPH:** Class of optimization problems to which NPC can be reduced.

**Theorem:** SAT $\in$ NPC.

**Theorem:** HC $\in$ NPC.
Millennium Problems

In order to celebrate mathematics in the new millennium, The Clay Mathematics Institute of Cambridge, Massachusetts (CMI) has named seven Prize Problems. The Scientific Advisory Board of CMI selected these problems, focusing on important classic questions that have resisted solution over the years. The Board of Directors of CMI designated a $7 million prize fund for the solution to these problems, with $1 million allocated to each. During the Millennium Meeting held on May 24, 2000 at the Collège de France, Timothy Gowers presented a lecture entitled The Importance of Mathematics, aimed for the general public, while John Tate and Michael Atiyah spoke on the problems. The CMI invited specialists to formulate each problem.

One hundred years earlier, on August 8, 1900, David Hilbert delivered his famous lecture about open mathematical problems at the second International Congress of Mathematicians in Paris. This influenced our decision to announce the millennium problems as the central theme of a Paris meeting.

The rules for the award of the prize have the endorsement of the CMI Scientific Advisory Board and the approval of the Directors. The members of these boards have the responsibility to preserve the nature, the integrity, and the spirit of this prize.

Paris, May 24, 2000

Please send inquiries regarding the Millennium Prize Problems to prize.problems@claymath.org.
Directed Hamiltonian Cycle

Directed Hamiltonian Cycle Problem

(Undirected) Hamiltonian Cycle Problem
The Satisfiability Problem

\[(x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor \neg x_3)\]
The Travelling Salesman Problem

Theorem: HC ∈ NPH.
Combinatorial Optimization Problem

**Input:** A finite (implicitly given) set $N = \{1, \ldots, n\}$, an objective function $f : N \to \mathbb{R}$.

**Question:** What is the minimum of $f$ over $N$?

$$\min f(x), \ x \in N$$
The Travelling Salesman Problem

D15112
TSP World Records

1954: dantzig42

1977: gr120

1987: gr666, pr2392

1994: pla7397

2001: d15112
Concorde TSP Solver

Concorde is a computer code for the symmetric traveling salesman problem (TSP) and some related network optimization problems. The code is written in the ANSI C programming language and it is available for academic research use; for other uses, contact William Cook for licensing options.

Concorde's TSP solver has been used to obtain the optimal solutions to 106 of the 110 TSPLIB instances; the largest having 85,900 cities.

The Concorde callable library includes over 700 functions permitting users to create specialized codes for TSP-like problems. All Concorde functions are thread-safe for programming in shared-memory parallel environments; the main TSP solver includes code for running over networks of UNIX workstations.

Concorde now supports the QSopt linear programming solver. Executable versions of concorde with qsopt for Linux and Solaris are now available.

Hans Mittelmann has created a NEOS Server for Concorde, allowing users to solve TSP instances online.
The S-Bahn Challenge

- The entire S-Bahn network must be visited in minimal time
- All stations and connections must be visited
- If several lines cover a connection, one suffices
- Walking and all timetabled means of transport are allowed
### Station Timetable

<table>
<thead>
<tr>
<th>Station</th>
<th>board line</th>
<th>Departure</th>
<th>Arrival</th>
<th>Changing time</th>
<th>Total</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bornholmer Strasse</td>
<td>25</td>
<td>05:24</td>
<td>05:47</td>
<td>0:02</td>
<td></td>
<td>On weekends the trains go all night every 30 min, so we can start basically anytime</td>
</tr>
<tr>
<td>Henningsdorf</td>
<td>25</td>
<td>05:49</td>
<td>06:07</td>
<td>0:15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schönholz</td>
<td>1</td>
<td>06:22</td>
<td>06:50</td>
<td>0:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oranienburg</td>
<td>1</td>
<td>06:51</td>
<td>07:00</td>
<td>0:04</td>
<td>1:36</td>
<td></td>
</tr>
<tr>
<td>Birkenwerder</td>
<td>8</td>
<td>07:04</td>
<td>07:24</td>
<td>0:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blankenburg</td>
<td>2</td>
<td>07:25</td>
<td>07:52</td>
<td>0:17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bernau</td>
<td>2</td>
<td>08:09</td>
<td>08:22</td>
<td>0:05</td>
<td>2:58</td>
<td></td>
</tr>
<tr>
<td>Gesundbrunnen</td>
<td>42</td>
<td>08:27</td>
<td>08:37</td>
<td>0:04</td>
<td></td>
<td>We are done with the Northern network branches</td>
</tr>
<tr>
<td>Westkreuz</td>
<td>5</td>
<td>08:41</td>
<td>08:55</td>
<td>0:03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spandau</td>
<td>5</td>
<td>08:58</td>
<td>09:11</td>
<td>0:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Westkreuz</td>
<td>7</td>
<td>09:12</td>
<td>09:28</td>
<td>0:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wannsee</td>
<td>1</td>
<td>09:29</td>
<td>09:35</td>
<td>0:05</td>
<td>4:11</td>
<td></td>
</tr>
<tr>
<td>Potsdam</td>
<td>1</td>
<td>09:40</td>
<td>10:15</td>
<td>0:03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schöneberg</td>
<td>41</td>
<td>10:18</td>
<td>10:28</td>
<td>0:04</td>
<td></td>
<td>From here...</td>
</tr>
<tr>
<td>Westkreuz</td>
<td>5 / 7</td>
<td>10:32</td>
<td>10:47</td>
<td>0:02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friedrichstrasse</td>
<td>1 / 2</td>
<td>10:49</td>
<td>11:00</td>
<td>0:04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yorkstrasse</td>
<td>1</td>
<td>11:04</td>
<td>11:07</td>
<td>0:02</td>
<td>5:43</td>
<td></td>
</tr>
<tr>
<td>Schöneberg</td>
<td>42</td>
<td>11:09</td>
<td>11:10</td>
<td>0:04</td>
<td></td>
<td>To here we move in the city centre</td>
</tr>
<tr>
<td>Südkreuz</td>
<td>25</td>
<td>11:14</td>
<td>11:32</td>
<td>0:04</td>
<td></td>
<td>From here...</td>
</tr>
<tr>
<td>Teltow Stadt</td>
<td>25</td>
<td>11:36</td>
<td>11:50</td>
<td>0:02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priestweg</td>
<td>2</td>
<td>11:52</td>
<td>12:12</td>
<td>0:14</td>
<td>6:48</td>
<td>To here we do the Southern branches</td>
</tr>
<tr>
<td>Blankenfelde</td>
<td>2</td>
<td>12:26</td>
<td>13:07</td>
<td>0:08</td>
<td></td>
<td>From here...</td>
</tr>
<tr>
<td>Gesundbrunnen</td>
<td>41</td>
<td>13:15</td>
<td>13:45</td>
<td>0:06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Südkreuz</td>
<td>42</td>
<td>13:51</td>
<td>14:00</td>
<td>0:04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sonnenallee</td>
<td>M41</td>
<td>14:04</td>
<td>14:07</td>
<td>0:03</td>
<td>8:43</td>
<td>To here we do the Southeastern branches</td>
</tr>
<tr>
<td>Köllnische Heide</td>
<td>M41 / S42 / S9</td>
<td>14:10</td>
<td>14:42</td>
<td>0:22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flughafen Berlin-Schönefeld</td>
<td>Bus EV or RB19</td>
<td>15:04</td>
<td>15:19</td>
<td>0:05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Königs Wusterhausen</td>
<td>46</td>
<td>15:19</td>
<td>15:44</td>
<td>0:11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schöneweide</td>
<td>8</td>
<td>15:55</td>
<td>16:01</td>
<td>0:03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spindlersfeld</td>
<td>47</td>
<td>16:04</td>
<td>16:28</td>
<td>0:07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ostkreuz</td>
<td>3</td>
<td>16:35</td>
<td>17:10</td>
<td>0:52</td>
<td>11:46</td>
<td>Here we do the Eastern branches. From here...</td>
</tr>
<tr>
<td>Erkner / Erkner ZOB</td>
<td>3</td>
<td>18:02</td>
<td>18:42</td>
<td>0:17</td>
<td>13:18</td>
<td></td>
</tr>
<tr>
<td>Ostbahnhof / Strausberg</td>
<td>75</td>
<td>18:59</td>
<td>19:09</td>
<td>0:06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strausberg Nord</td>
<td>5</td>
<td>19:15</td>
<td>19:55</td>
<td>0:06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friedrichsfelde Ost</td>
<td>7</td>
<td>20:01</td>
<td>20:14</td>
<td>0:06</td>
<td>14:50</td>
<td></td>
</tr>
<tr>
<td>Ahrensfelde</td>
<td>7</td>
<td>20:20</td>
<td>20:29</td>
<td>0:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Springpfuhl</td>
<td>75</td>
<td>20:30</td>
<td>20:39</td>
<td>0:07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wartenberg</td>
<td>75</td>
<td>20:46</td>
<td>21:09</td>
<td>0:11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ostkreuz</td>
<td>5 / 7</td>
<td>21:20</td>
<td>21:22</td>
<td>0:05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friedrichstrasse</td>
<td>2</td>
<td>21:27</td>
<td>21:35</td>
<td>0:06</td>
<td>16:11</td>
<td></td>
</tr>
<tr>
<td>Gesundbrunnen</td>
<td>41</td>
<td>21:35</td>
<td>21:37</td>
<td>0:04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schönhauser Allee</td>
<td>9</td>
<td>21:41</td>
<td>21:44</td>
<td>0:06</td>
<td>16:20</td>
<td></td>
</tr>
</tbody>
</table>

**Paths and Complexity | COεTL 2015**
<table>
<thead>
<tr>
<th>Station</th>
<th>in Linie</th>
<th>Ab</th>
<th>An</th>
<th>Um</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bornholmer Strasse</td>
<td>25</td>
<td>05:24</td>
<td>05:47</td>
<td>0:02</td>
<td></td>
</tr>
<tr>
<td>Henningsdorf</td>
<td>25</td>
<td>05:49</td>
<td>06:07</td>
<td>0:15</td>
<td></td>
</tr>
<tr>
<td>Schönholz</td>
<td>1</td>
<td>06:22</td>
<td>06:50</td>
<td>0:08</td>
<td></td>
</tr>
<tr>
<td>Oranienburg</td>
<td>1</td>
<td>06:51</td>
<td>07:00</td>
<td>0:09</td>
<td></td>
</tr>
<tr>
<td>Birkenwerder</td>
<td>8</td>
<td>07:04</td>
<td>07:24</td>
<td>0:17</td>
<td></td>
</tr>
<tr>
<td>Blankenburg</td>
<td>2</td>
<td>07:25</td>
<td>07:52</td>
<td>0:17</td>
<td></td>
</tr>
<tr>
<td>Bernau</td>
<td>2</td>
<td>08:09</td>
<td>08:22</td>
<td>0:15</td>
<td></td>
</tr>
<tr>
<td>Gesundbrunnen</td>
<td>42</td>
<td>08:27</td>
<td>08:37</td>
<td>0:04</td>
<td></td>
</tr>
<tr>
<td>Westkreuz</td>
<td>5</td>
<td>08:41</td>
<td>08:55</td>
<td>0:14</td>
<td></td>
</tr>
<tr>
<td>Spandau</td>
<td>5</td>
<td>08:58</td>
<td>09:11</td>
<td>0:13</td>
<td></td>
</tr>
<tr>
<td>Westkreuz</td>
<td>7</td>
<td>09:12</td>
<td>09:28</td>
<td>0:16</td>
<td></td>
</tr>
<tr>
<td>Wannsee</td>
<td>1</td>
<td>09:29</td>
<td>09:35</td>
<td>0:06</td>
<td></td>
</tr>
<tr>
<td>Potsdam</td>
<td>1</td>
<td>09:40</td>
<td>10:15</td>
<td>0:35</td>
<td></td>
</tr>
<tr>
<td>Schöneberg</td>
<td>41</td>
<td>10:18</td>
<td>10:28</td>
<td>0:10</td>
<td></td>
</tr>
<tr>
<td>Westkreuz</td>
<td>5 / 7</td>
<td>10:32</td>
<td>10:47</td>
<td>0:15</td>
<td></td>
</tr>
<tr>
<td>Friedrichstrasse</td>
<td>1 / 2</td>
<td>10:49</td>
<td>11:00</td>
<td>0:11</td>
<td></td>
</tr>
<tr>
<td>Yorkerstrasse</td>
<td>1</td>
<td>11:04</td>
<td>11:07</td>
<td>0:03</td>
<td></td>
</tr>
<tr>
<td>Schöneberg</td>
<td>42</td>
<td>11:09</td>
<td>11:10</td>
<td>0:09</td>
<td></td>
</tr>
<tr>
<td>Südkreuz</td>
<td>25</td>
<td>11:14</td>
<td>11:32</td>
<td>0:18</td>
<td></td>
</tr>
<tr>
<td>Teltow Stadt</td>
<td>25</td>
<td>11:36</td>
<td>11:50</td>
<td>0:14</td>
<td></td>
</tr>
<tr>
<td>Priesterweg</td>
<td>2</td>
<td>11:52</td>
<td>12:12</td>
<td>0:40</td>
<td></td>
</tr>
<tr>
<td>Blankenfelde</td>
<td>2</td>
<td>12:26</td>
<td>13:07</td>
<td>0:41</td>
<td></td>
</tr>
<tr>
<td>Gesundbrunnen</td>
<td>41</td>
<td>13:15</td>
<td>13:45</td>
<td>0:30</td>
<td></td>
</tr>
<tr>
<td>Südkreuz</td>
<td>42</td>
<td>13:51</td>
<td>14:00</td>
<td>0:09</td>
<td></td>
</tr>
<tr>
<td>Sonnenallee</td>
<td>M41</td>
<td>14:04</td>
<td>14:07</td>
<td>0:05</td>
<td></td>
</tr>
<tr>
<td>Köllnische Heide</td>
<td>M41 / S42 / S9</td>
<td>14:10</td>
<td>14:42</td>
<td>0:22</td>
<td></td>
</tr>
<tr>
<td>Flughafen Berlin-Schönefeld</td>
<td>Bus EV or RB19</td>
<td>15:04</td>
<td>15:19</td>
<td>0:05</td>
<td></td>
</tr>
<tr>
<td>Königs Wusterhausen</td>
<td>46</td>
<td>15:19</td>
<td>15:44</td>
<td>0:25</td>
<td></td>
</tr>
<tr>
<td>Schöneiche</td>
<td>8</td>
<td>15:55</td>
<td>16:01</td>
<td>0:06</td>
<td></td>
</tr>
<tr>
<td>Spindlersfelde</td>
<td>47</td>
<td>16:04</td>
<td>16:28</td>
<td>0:18</td>
<td></td>
</tr>
<tr>
<td>Ostkreuz</td>
<td>3</td>
<td>16:35</td>
<td>17:10</td>
<td>0:55</td>
<td></td>
</tr>
<tr>
<td>Erkner / Erkner ZOB</td>
<td>3</td>
<td>18:02</td>
<td>18:42</td>
<td>0:44</td>
<td></td>
</tr>
<tr>
<td>Ostbahnhof / Strausberg</td>
<td>75</td>
<td>18:59</td>
<td>19:09</td>
<td>0:10</td>
<td></td>
</tr>
<tr>
<td>Strausburg</td>
<td>75</td>
<td>19:15</td>
<td>19:55</td>
<td>0:40</td>
<td></td>
</tr>
<tr>
<td>Friedrichsfelde Ost</td>
<td>7</td>
<td>20:01</td>
<td>20:14</td>
<td>0:13</td>
<td></td>
</tr>
<tr>
<td>Ahrensfelde</td>
<td>7</td>
<td>20:20</td>
<td>20:29</td>
<td>0:09</td>
<td></td>
</tr>
<tr>
<td>Springpfuhl</td>
<td>75</td>
<td>20:30</td>
<td>20:39</td>
<td>0:09</td>
<td></td>
</tr>
<tr>
<td>Wartenberg</td>
<td>75</td>
<td>20:46</td>
<td>21:09</td>
<td>0:23</td>
<td></td>
</tr>
<tr>
<td>Ostkreuz</td>
<td>5 / 7</td>
<td>21:20</td>
<td>21:22</td>
<td>0:02</td>
<td></td>
</tr>
<tr>
<td>Friedrichstrasse</td>
<td>2</td>
<td>21:27</td>
<td>21:35</td>
<td>0:19</td>
<td></td>
</tr>
<tr>
<td>Gesundbrunnen</td>
<td>41</td>
<td>21:35</td>
<td>21:37</td>
<td>0:02</td>
<td></td>
</tr>
<tr>
<td>Schönehauer Allee</td>
<td>9</td>
<td>21:41</td>
<td>21:44</td>
<td>0:03</td>
<td></td>
</tr>
<tr>
<td>Bornholmer Strasse</td>
<td>DONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Guinness Book of Records

- **Fastest time to travel to all the Berlin U-Bahn metro stations**
  The fastest time to travel to all the Berlin U-Bahn metro stations is 7 hr 33 min 15 sec and was achieved by Oliver Ziemek, Henning Colsman-Freyberger, Michael Wurm and Rudolf von Grot (all Germany) at Hönow station, Berlin, Germany on 2 May 2014.

- **Fastest time to travel to all London Underground stations**
  The fastest time to travel to all London Underground network stations is 16 hr 14 min 10 sec, and was achieved by Clive Burgess and Ronan McDonald (both UK) in London, UK, on 19 February 2015. Clive and Ronan's record breaking journey began at Chesham and ended at Heathrow Terminal 5.

- **Fastest time to travel to all New York City Subway stations**
  The fastest time to travel the entire New York City Subway is 22 hr 26 min 02 sec and was achieved by Andi James, Steve Wilson, Peter Smyth, Martin Hazel, Glen Bryant and Adham Fisher (all UK) between 18 and 19 November 2013. Andi James, Steve Wilson and Martin Hazel are previous record holders of the record for the 'Fastest time to travel to all London Underground stations.'
The History of the Transit-Challenge

<table>
<thead>
<tr>
<th>Date</th>
<th>Record Holder(s)</th>
<th>Stations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>00/30.05.1940</td>
<td>Herman Rinke</td>
<td>All stations</td>
<td>25:00:00~</td>
</tr>
<tr>
<td>00/01.06.1966</td>
<td>Michael Feldman and James Brown</td>
<td>All stations</td>
<td>23:16:xx</td>
</tr>
<tr>
<td>25/26.10.1989</td>
<td>Kevin Foster</td>
<td>All stations</td>
<td>26:21:08</td>
</tr>
<tr>
<td>28/29.12.2006</td>
<td>Bill Amarosa Jr., Michael Boyle, Brian Brockmeyer, Stefan Karpinski, Jason Laska and Andrew Weir</td>
<td>All stations</td>
<td>24:54:03</td>
</tr>
<tr>
<td>22/23.01.2010</td>
<td>Matt Ferrisi and Chris Solarz</td>
<td>All stations</td>
<td>22:52:36</td>
</tr>
<tr>
<td>18/19.11.2013</td>
<td>Andi James, Steve Wilson, Martin Hazel, Glen Bryant, Peter Smyth and Adham Fisher</td>
<td>All stations</td>
<td>22:26:02</td>
</tr>
</tbody>
</table>

1. Ride that requires a rider to traverse every line, but not necessarily the entire line. (Class A)

2. Full-system ride that requires a rider to stop at each station. (Class B)

3. Skip-stop ride that only requires a rider to pass through each station. (Class C)
TRAVELLING THE NEW YORK CITY SUBWAY IN THE SHORTEST TIME

The following acts as a guide to the specific considerations and understandings, in addition to the general requirements as detailed in the General Rules of the Record Breakers' Pack, for any potential attempt on the above record.

They should be read and understood by all concerned – organisers, participants and witnesses – prior to the event.

Please note that, as detailed in the Agreement Regarding Record Attempts, these guidelines in no way provide any kind of safety advice or can be construed as providing any comfort that the record is free from risk.

GUIDELINES

This record is for travelling the entire MTA New York City Subway system in the least amount of time.

1 - All of the stations served by the subway system must be visited. To ‘visit’ a station, the challenger must arrive and/or depart by a subway train in normal public service. It is necessary for a train to stop at the station for the visit to count, although the challenger does not need to leave the train at that station. If a station is normally open only at certain times of the day, this must be taken into account during planning. Only if a station is temporarily closed (e.g. for rebuilding, or in an emergency) will a non-stop pass through a station be acceptable.

2 - It is only necessary to visit all the stations on the network, not to travel every stretch of line. Thus, if a station is served by more than one line it is not necessary to visit that station on each line.

3 - Challengers may travel the same stretch of track (and visit the same station) more than once if necessary.

4 - Attempts on this record must be continuous (i.e. any breaks or stops that are taken must be included in the final time).

5 - Transfers between subway lines must be made by scheduled public transport or on foot. The use of private motor vehicles, taxis or any other form of privately arranged transport (bicycles, skateboards etc) is not allowed.

AUTHENTICATION

For the purposes of verifying any claim, the following must be provided:

Witness Book

Any attempt must take place in view of the public, wherever possible, and a book must be available for independent witnesses to sign. The book should be set up so that the following details can be included for each potential witness:

Date & Time  Location  Name  Signature

For solo and unsupported attempts, we appreciate that it might not be possible to gain an unbroken line of witnesses for the attempt, but one should try to obtain as many as possible. For an attempt, which is supported by a backup team, we would expect it to be possible to gain sufficient numbers of independent witnesses to enable verification for the entire duration of the attempt. Where possible, local dignitaries and police should be sought to sign the book.

Log Book

A logbook detailing every stage of the journey, i.e. the time of arrival and departure from each station, line changes, commutes between lines and stations, etc. must be maintained. This book should illustrate clearly the route followed. All rest breaks or stoppages for whatever reason must also be fully detailed in the log.

To attest to the validity and genuineness of the claim, we require signed statements of authentication by two independent persons of some standing, one of whom should have attended the beginning of the event, and if possible the end.

These statements should originate directly from the witnesses (in their own hand) and be submitted where possible on their own headed notepaper and include full contact details.

Statements should not take the form of documents pre-prepared by those involved in the record attempt.
Directed rural postman problem (DRPP)

Generalized DRPP (GDRPP)

Directed TSP (ATSP)

Generalized ATSP (GATSP)
Transfers ...
... in a Frequency Timetable modulo 60

- Most lines run every 10 or 20 mins
- One line runs every 40 mins, regional trains every 60 mins
- One 40 and one 60 mins connection can be shifted into a 20 mins solution

```
duration: 26

time: 01
duration: 27
freq: 10
freq: 10
```
The Generalized Direct Rural Postman Problem

- Visit every group of connections at least once
From the GDRPP to the Generalized directed TSP (1)

- Turn edges into nodes and connect succeeding ones
- Visit every group of nodes at least once
Connect all nodes from **different** groups using two arcs, namely, nodes \( a \) and \( a' \) using arcs \((u, v)\) and \((u', v')\) s.t.

- arc \((a, a')\) with length \( l(u, v) + \text{length of shortest } (v, u')\)-path
- arc \((a', a)\) with length \( l(u', v') + \text{length of shortest } (v', u)\)-path
Visit every group of nodes exactly once → visit every node exactly once: connect every group of nodes using a directed cycle of length 0.
From the GATSP to the ATSP (2)

- Replace arcs between different groups of nodes by new arcs
  - New start node is the one that precedes in the cycle
  - Increase weight by a large constant $M_1$ that ensures that every group is visited exactly once, i.e., for a cycle $(u_1, \ldots, u_i, u_{i+1}, \ldots)$ set $w'(u_i, v_j) = w(u_{i+1}, v_j) + M_1$
From the ATSP to the TSP

- Make two copies \( v \) und \( v' \) of every node \( v \)
  - Connect \( v \) and \( v' \) via undirected edges with weight \(-M_2\)
  - Connect \( u \) and \( v' \) via undirected edge with weight \( w(u, v) \)
- Open tour with given start node
  - Add additional node 0 for tour start and end
  - Add suitable weights 0 and \( \infty \)
Preprocessing

- Only start, end, or transfer stations: 166 → 113 stations
- No transfer on parallel lines: 113 → 42 stations
- Assumption: equal travel and transfer times
Program TOPTraC
(http://www.zib.de/projects/s-bahn-challenge-berlin)

- Open or closed tours
- Start, end station, or both
- Timetable or constant transfer times
- Exceptions for minimum transfer times at large stations
- TSP mode (for visiting only stations)
Program TOPTraC
(http://www.zib.de/projects/s-bahn-challenge-berlin)

Input
- File with all edge and travel times
- File with all stations, departure times, lines, and frequencies
- Timetable period

Output
- TSP file

Solution
- Via Concorde on NEOS server

Result
- List of all edges in optimal solution (unsorted)
The Solution

Best found solution: 13:17
- Contains some 1-minute transfers at large stations, which are not feasible according to the BVG trip planner
- The last connection Strausberg-Strausberg Nord on the S5 is operated at a 40 mins frequency

More realistic solution: 13:44
- Start at Strausberg Nord
- Lower bounds on transfers at large stations
- Feasible according to BVG trip planner

With luck: 13:24
- If some infeasible transfers are caught
The Realistic Solution: 13:44
The Record Attempt: 10.01.2015, 09:55

- The first 7 hours worked according to the plan
- One infeasible transfer was caught: 20 mins ahead of schedule!
The Record Attempt: 10.01.2015, 09:55

- The first 7 hours went according to the plan
- One infeasible transfer was caught
- Thunderstorm Felix
- Severe service disruptions
- Finally 15:04 instead of 13:44
- 80 mins delay
- 2 hours faster than before
- Not yet (?) in Guinness Book
- 2 stations were visited by regional trains
- Legal or not?
Studentin fährt alle S-Bahnstationen in 15 Stunden ab

Eine niederländische Mathematik-Studentin fährt die 166 Haltestellen der Berliner S-Bahn ab – in Rekordzeit. Was nach einer verrückten Kneipenwette klingt, ist eine echte fachliche Herausforderung.

Von Thomas Fülling
Edsger Wybe Dijkstra (1930-2002)
Passenger Information

Verbindungen - Übersicht

Ihre Anfrage
Start: S+U Alexanderplatz Bhf
Ziel: U Dahlmen-Dorf
Datum: Sa, 13.05.06
Zeit: 19:00 (Ankunft)

Übersicht

<table>
<thead>
<tr>
<th>Karte</th>
<th>Bahnhof/Haltestelle</th>
<th>Datum</th>
<th>Zeit</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>S+U Alexanderplatz Bhf</td>
<td>13.05.06</td>
<td>ab 14:14</td>
</tr>
<tr>
<td>✔</td>
<td>U Dahlmen-Dorf</td>
<td>13.05.06</td>
<td>ab 14:17</td>
</tr>
<tr>
<td>🚆</td>
<td>S+U Alexanderplatz Bhf</td>
<td>13.05.06</td>
<td>ab 14:20</td>
</tr>
<tr>
<td>✔</td>
<td>U Dahlmen-Dorf</td>
<td>13.05.06</td>
<td>ab 15:04</td>
</tr>
</tbody>
</table>

Detailansicht

<table>
<thead>
<tr>
<th>Karte</th>
<th>Bahnhof/Haltestelle</th>
<th>Linie / Richtung</th>
<th>abf./Ank.</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>S+U Alexanderplatz Bhf</td>
<td>U 12 [1], U 30 [1]</td>
<td>ab 14:24</td>
</tr>
<tr>
<td>✔</td>
<td>U Wittenbergeplatz</td>
<td>U 8 [1], U 30 [1]</td>
<td>ab 14:46</td>
</tr>
<tr>
<td>✔</td>
<td>U Dahlmen-Dorf</td>
<td>U 8 [1], U 120 [1]</td>
<td>an 15:00</td>
</tr>
</tbody>
</table>

13.05.06; Dauer 0:36; fährt nicht täglich, Verkehrstage Tarifauskunft
Graph Theoretic Model

306 nodes, 445 edges
Preprocessing

80 nodes, 122 edges
Dijkstra's Algorithm (0)
Dijkstra's Algorithm (1)
Dijkstra's Algorithm (2)
Dijkstra's Algorithm (3)
Dijkstra's Algorithm (4)
Dijkstra's Algorithm (5)
Shortest Path Tree
Running Time of Dijkstra's Algorithm

- Set all node labels = 0,
  
  \[ \text{distances} = \infty, \text{predecessors} = \text{none} \quad \text{O}(n) \]

- Set distance at start node = 0, pred. = start \quad \text{O}(1)

- Repeat
  
  - Find unlabeled node with minimum distance or stop, done! \quad \text{O}(n)
  - Label it \quad \text{O}(1)
  - For all outbound edges
    
    - Update distance and predecessor labels \quad \text{O}(n)

\[ \text{O}(n^2) \]

\textbf{Theorem:} Dijkstra's algorithm runs in polynomial time.
Contraction (Kolman & Pangrac [2009])

Paths and Complexity | COeTL 2015
Table 1: Performance of various speedup techniques on Western Europe. Column source indicates the implementation tested for this survey.

<table>
<thead>
<tr>
<th>algorithm</th>
<th>source</th>
<th>space [GiB]</th>
<th>time [h:m]</th>
<th>scanned vertices</th>
<th>time [µs]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dijkstra</td>
<td>[65]</td>
<td>0.4</td>
<td>–</td>
<td>9 300 000</td>
<td>2 550 000</td>
</tr>
<tr>
<td>Bidir. Dijkstra</td>
<td>[65]</td>
<td>0.4</td>
<td>–</td>
<td>4 800 000</td>
<td>1 350 000</td>
</tr>
<tr>
<td>CRP</td>
<td>[67]</td>
<td>0.9</td>
<td>1:00</td>
<td>2 766</td>
<td>1 650</td>
</tr>
<tr>
<td>Arc Flags</td>
<td>[65]</td>
<td>0.6</td>
<td>0:20</td>
<td>2 646</td>
<td>408</td>
</tr>
<tr>
<td>CH</td>
<td>[67]</td>
<td>0.4</td>
<td>0:05</td>
<td>28</td>
<td>110</td>
</tr>
<tr>
<td>CHASE</td>
<td>[65]</td>
<td>0.6</td>
<td>0:30</td>
<td>28</td>
<td>5.76</td>
</tr>
<tr>
<td>HLC</td>
<td>[70]</td>
<td>1.8</td>
<td>0:50</td>
<td>–</td>
<td>2.55</td>
</tr>
<tr>
<td>TNR</td>
<td>[13]</td>
<td>2.5</td>
<td>0:20</td>
<td>–</td>
<td>1.25</td>
</tr>
<tr>
<td>TNR+AF</td>
<td>[37]</td>
<td>5.4</td>
<td>1:45</td>
<td>–</td>
<td>0.99</td>
</tr>
<tr>
<td>HL</td>
<td>[70]</td>
<td>18.8</td>
<td>0:37</td>
<td>–</td>
<td>0.56</td>
</tr>
<tr>
<td>HL-∞</td>
<td>[5]</td>
<td>17.7</td>
<td>60:00</td>
<td>–</td>
<td>0.25</td>
</tr>
<tr>
<td>table lookup</td>
<td>[65]</td>
<td>1 208 358.7</td>
<td>145:30</td>
<td>–</td>
<td>0.06</td>
</tr>
</tbody>
</table>
Airway network – Denmark
Constraints

Euler Tour

Hamiltonian Cycle

Shortest Path

Aircraft Trajectory
### 1. Shortest Paths with Pair Constraints

#### If EDDF is the departure airport, then BIBTI must not be visited:

\[
\text{EDDF} \in P \implies \text{BIBTI} \not\in P \text{ (easy because departure is known)}
\]

#### If MILKA is visited, then ALG must be visited:

\[
\text{MILKA} \in P \implies \text{ALG} \in P \text{ (hard because visits are not known)}
\]
Forbidden, Obligatory, and Binding Pairs
Forbidden, Obligatory, and Binding Pairs

Given an ATS network, two nodes (waypoints) $s$ and $t$, and a set of node-pairs $uv$, find a shortest $st$-path such that for every pair ...

Shortest-path problem with binding pairs: $u \in P \Rightarrow v \in P$

Shortest-path problem with forbidden pairs: $u \in P \Rightarrow v \notin P$

Shortest-path problem with obligatory pairs: $u \notin P \Rightarrow v \in P$
Shortest Paths with Forbidden Pairs

- Applications in automatic software testing and bioinformatics (Gabow et al. [1976], Chen et al. [2001])
- NP- and APX-hard (Gabow et al. [1976], Hajiaghayi et al. [2010])
- Efficient contraction/dynamic programming algorithms for networks with special structures (Kolman and Pangrac [2009], Kovac [2011])
- Graph-modifying algorithm for Shortest Paths with forbidden Subpaths (Ahmad & Lubiw [2009])

Diagram:

- General problem
- Overlapping structure
  - Ordered
  - Well-parenthesized
  - Halving
  - Nested
  - Disjoint

Example:

Paths and Complexity | COeTL 2015
Shortest Paths with Binding Pairs

- Applications in automatic software testing
- NP-Hard even on acyclic digraphs (Ntafos, Hakimi [1979])
- No further literature
Shortest Paths with Obligatory Pairs

- Related to group TSP
- NP-hard in general
- Polynomial for fixed number of pairs
Shortest Paths with Pair Constraints

min \ c^T x
\begin{align*}
x(\delta^- (v)) - x(\delta^+ (v)) &= \delta_{st} (v) & \forall v \in V \\
x(\delta^- (u)) + x(\delta^- (v)) &\leq 1 & \forall uv \in F \\
x(\delta^- (u)) - x(\delta^- (v)) &\leq 0 & \forall uv \in B \\
x(\delta^- (u)) + x(\delta^- (v)) &\geq 1 & \forall uv \in O
\end{align*}

x_{uv} \in \{0,1\} \quad \forall uv \in A

- No subtour elimination in acyclic digraphs

**Proposition (Brückner [2015])**

The SPPPC can be solved in polynomial time if the pairs are well-parenthesized.

**Theorem (Blanco, B, Brückner, Hoang [2015])**

A complete linear description of the SPPPC polytope can be obtained if the pairs are disjoint.
Network Orientation

- Delete network segments pointing in the "wrong direction"
  - that point backward over the cut halfway between origin & destination
  - that point back to the origin (in the origin's hemisphere)
  - that point away from the destination (in the destination's hemisphere)
  - Results in acyclic network
- Errors near airports, mostly (but not always) small
### Adjusted unit rates applicable to April 2014 flights

Please find hereunder the unit rates of route charges applicable to April 2014 flights, as well as the exchange rates used for their calculation, i.e. the average exchange rates for the month of March 2014 (monthly average of the “Closing Cross Rate” calculated by Reuters based on daily BID rate).

<table>
<thead>
<tr>
<th>Zone</th>
<th>Taux unitaire Unit rate</th>
<th>Taux de change Exchange rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EUR</td>
<td>1 EUR =</td>
</tr>
<tr>
<td>Portugal Santa Maria *</td>
<td>10.60</td>
<td>./.</td>
</tr>
<tr>
<td>Belg.-Luxembourg *</td>
<td>72.19</td>
<td>./.</td>
</tr>
<tr>
<td>Allemagne / Germany *</td>
<td>77.47</td>
<td>./.</td>
</tr>
<tr>
<td>Finlande / Finland *</td>
<td>52.21</td>
<td>./.</td>
</tr>
<tr>
<td>Royaume-Uni / United Kingdom</td>
<td>84.85</td>
<td>0.831957 GBP</td>
</tr>
<tr>
<td>Pays-Bas / Netherlands *</td>
<td>66.62</td>
<td>./.</td>
</tr>
<tr>
<td>Irlande / Ireland *</td>
<td>30.77</td>
<td>./.</td>
</tr>
<tr>
<td>Danemark / Denmark</td>
<td>71.35</td>
<td>7.46207 DKK</td>
</tr>
<tr>
<td>Norvège / Norway</td>
<td>51.96</td>
<td>8.29105 NOK</td>
</tr>
<tr>
<td>Pologne / Poland</td>
<td>35.26</td>
<td>4.19932 PLN</td>
</tr>
<tr>
<td>Suède / Sweden</td>
<td>72.25</td>
<td>8.86081 SEK</td>
</tr>
<tr>
<td>Lettonie / Latvia *</td>
<td>28.59</td>
<td>0.702804 *** LVL</td>
</tr>
<tr>
<td>Lituanie / Lithuania</td>
<td>45.92</td>
<td>3.45158 LTL</td>
</tr>
<tr>
<td>Espagne / Spain - Canarias *</td>
<td>58.51</td>
<td>./.</td>
</tr>
</tbody>
</table>
ATC Charges

- Rate per flown km: easy to integrate in any search algorithm
- Rate per km in the great circle segment between entry and exit points: hard (Lido "pretends" it's the first model)
ATC Charges

\[ d_{uv} \] \[ c_{uv} \] \[ c_{uw} \]
\[
\begin{align*}
\text{min} & \quad \sum_{(w,z) \in A} c_{wz} x_{wz} + \sum_{i} \sum_{(u,v) \in D_i} d^i_{uv} y_{uv} \\
\text{s.t.} & \quad \sum_{z: (w,z) \in A} x_{wz} - \sum_{u: (u,w) \in A} x_{uw} = \begin{cases} 1 & \text{if } w = s \\ -1 & \text{if } w = t \\ 0 & \text{else} \end{cases} \quad \forall w \in V \\
& \quad \sum_{w: (u,w) \in A_i} x_{uw} = \sum_{v: (u,v) \in D_i} y_{uv} \quad \forall u \in \partial^- (V_i), \forall i \\
& \quad \sum_{w: (w,v) \in A_i} x_{wv} = \sum_{u: (u,v) \in D_i} y_{uv} \quad \forall u \in \partial^+ (V_i), \forall i \\
& \quad \sum_{i} \sum_{v: (u,v) \in D_i} y_{uv} - \sum_{i} \sum_{w: (v,w) \in D_i} y_{vw} = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = t \\ 0 & \text{else} \end{cases} \quad \forall v \in \bigcup_{i} \partial^+/-(V_i) \\
& \quad x_{wz}, y^i_{uv} \in \{0, 1\} \quad \forall (w, z) \in A, \forall (u, v) \in D_i, \forall i
\end{align*}
\]
Proposition (Blanco [2014]): The Shortest Path Problem with ATC charges can be solved in polynomial time.
3. 3D-Shortest Paths
Horizontal segments are well defined.
Horizontal segments are well defined.
Horizontal segments are well defined.
Climb/descent segments?
Vertical segments vary by weight, weather, speed, etc.
Vertical segments vary by weight, weather, speed, etc.
Vertical segments vary by weight, weather, speed, etc.
Vertical segments vary by weight, weather, speed, etc.
Vertical segments vary by weight, weather, speed, etc.
Cruise Performance

Constant speed, altitude, current weight, and specific range function

\[ f(w_{curr}) = d \]

(how far can we fly with 1kg of fuel with weight \( w_{curr} \)). Then

\[ d = \int_{w_{end}}^{w_{curr}} f(s)ds = F(w_{curr}) - F(w_{end}), \]

where \( F \) is the primitive integral of \( f \), i.e.,

\[ w_{end} = F^{-1}(F(w_{curr}) - d). \]
Cruise Performance

![Graph showing distance vs. weight with points labeled as follows: \( F(w_{\text{curr}}) \), \( d \), \( F(w_{\text{curr}}) - d \), \( w_{\text{empty}} \), \( w_{\text{end}} \), and \( w_{\text{curr}} \).]
Overestimation

\[ F^0(w_{\text{curr}}) \]

\[ F^0(w_{\text{curr}}) - d \]

\[ w_{\text{empty}} \]

\[ w_{\text{end}} \]

\[ w_{\text{curr}} \]
Interpolation
Piecewise Linear Approximation

\[ F^{pw1}(w_{curr}) \]

\[ F^{pw1}(w_{curr}) - d \]

Distance

Paths and Complexity | COeTL 2015
Cruise consumption error (A380, 330Kn/0.85Mn)

IPO

PWL
Proposition (Blanco, B, Hoang, Spiegel [2015])

Consider an aircraft cruising along several segments $e_0, \ldots, e_k$ at a constant flight level at constant speed. Let $w$ be the actual weight after the cruise phase, and $w$ and $w^{\updownarrow}$ the values obtained by a piecewise linear underestimation of the primitive integral $F$ of the specific range function $f$ and a piecewise linear overestimation of its inverse $F^{-1}$ using the same breakpoints with approximation errors $K^{\downarrow}$ and $K^{\uparrow}$, respectively. Then

$$|w - w^{\updownarrow}| \leq \max(K^{\downarrow} \cdot ||f^{-1}||_\infty, K^{\downarrow-1}),$$

independently of the number of segments.
A380, speed 0.83MN/300KIAS, constant altitude FL300, departure time 06.03.2014, 19:30:25
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