Telecommunication System Design: Minimum-Cost Embeddings of Reliable Virtual Private Networks

Prepared by Andreas S. Schulz
Overview

- The Project's Origin
- The Problem's Origin
- An Integer Programming Model
- A Lagrangian Relaxation
- A Comparison of Lower Bounds
- A Branch & Bound Approach
- Computational Results
- Extensions
The Project’s Origin

- R&D Division, Deutsche Telekom AG, Darmstadt.
- Math. Dept., Berlin University of Technology:
  - Ewgenij Gawrilow (Programmer),
  - Olaf Jahn (Research Assistant),
  - Rolf H. Möhring (Principal Investigator),
  - Martin Oellrich (Research Assistant),
  - Andreas S. Schulz (Principal Investigator).
- Official Start: July 1, 1995. 1 year.
Prior to Deregulation.
Operation + management of network infrastructure **and** provision of network services organized as integrated process.

Post Deregulation.
Competing providers of network services lease required transport capacity from carriers of physical transmission networks.

Resulting subnetworks are independently operated.
A VPN appears to be exclusively controlled and managed by a customer alone.

In reality, it consists of a number of lines leased from a carrier.

VPNs are typically deployed as data service networks, or backbones for mobile and ATM networks.
The carrier has to balance two conflicting goals:

- On the one hand, customers require reliability.
- On the other hand, costs have to be kept at a minimum in order to be competitive.

**Note.** In network engineering, reliability is a key issue. Keeping their networks stable and operable are primary goals of all service providers.
A VPN is **reliable** when no single fault in one physical trunk can affect more than one of the logical connections in the VPN. One can accommodate this request by routing no two connections over a common trunk.

Technically speaking, the different leased lines must be routed **disjointly**.
THE MINIMUM COST DISJOINT PATHS PROBLEM.

Input: A graph $G = (V, E)$, $c : E \rightarrow \mathbb{N}$, $k$ pairs $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$ of terminals.

Goal: Find $k$ edge-disjoint paths $P_1, \ldots, P_k$ in $G$ connecting $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$ such that $\sum_{i=1}^{k} c(P_i)$ is minimum.
Computational Complexity

- How difficult is the **Minimum Cost Disjoint Paths Problem**?
- What if we disregard the disjointness constraint?
Mathematical Model

\[
\begin{align*}
\min & \sum_{\ell=1}^{k} \sum_{(i,j) \in A} c_{ij} x_{ij}^\ell \\
\text{s.t.} & \sum_{j} x_{ij}^\ell - \sum_{j} x_{ji}^\ell = b_i^\ell & \forall i \in V, \forall \ell \in \{1, \ldots, k\}, \\
& \sum_{\ell=1}^{k} (x_{ij}^\ell + x_{ji}^\ell) \leq 1 & \forall \{i, j\} \in E, \\
& x_{ij}^\ell \in \{0, 1\} & \forall (i, j) \in A, \forall \ell \in \{1, \ldots, k\}.
\end{align*}
\]
Mathematical Model

Directed Network:

\[ A := \{ (u, v), (v, u) \mid \{u, v\} \in E \} \]

Supplies/Demands:

\[ b_i^\ell := \begin{cases} 
1, & \text{if } i = s_\ell, \\
-1, & \text{if } i = t_\ell, \\
0, & \text{otherwise.}
\end{cases} \]

Decisions:

\[ x_{ij}^\ell := \begin{cases} 
1, & \text{if } \{i, j\} \text{ belongs to path } P_\ell, \\
0, & \text{otherwise.}
\end{cases} \]
Lower Bounds

- Proving optimality.
- Identifying near-optimal solutions.
- Reducing search space in enumerative approaches.

Your Ideas for Lower Bounds?

1. Combinatorial (Cheapest Paths)
2. Linear Programming Relaxation
3. Lagrangian Relaxation
Lagrangian Relaxation

Dualizing “Bad” Constraints

\[ \text{OP} : \min_{x \in \{0,1\}} \ c \ x \]
\[ \text{s.t.} \quad N x = b \]
\[ \quad A x \leq 1 \]

We know that optimization over the constraints “\( N x = b, \ x \in \{0,1\} \)” is easy.

The addition of the constraints “\( A x \leq 1 \)” makes the problem much more difficult.

Let \[ P = \left\{ x \in \{0,1\} : \ N x = b \right\}. \]
Lagrangian Relaxation

Dualizing “Bad” Constraints

$\text{OP : } \min_x c x$

$s.t. \quad Ax \leq 1$

$x \in P$

The Lagrangian is:

$L(x, u) = c x + u (Ax - 1) = -u 1 + (c + u A)x$

And the Lagrangian dual, for $u \geq 0$:

$L^*(u) := \min_x -u 1 + (c + u A)x$

$s.t. \quad x \in P$
Lagrangian Relaxation

\[
\begin{align*}
\min & \quad \sum_{\ell=1}^{k} \sum_{(i,j) \in A} c_{ij} x_{ij}^{\ell} + \sum_{\{i,j\} \in E} u_{ij} \left( \sum_{\ell=1}^{k} (x_{ij}^{\ell} + x_{ji}^{\ell}) - 1 \right) \\
\sum_{j} (x_{ij}^{\ell} - x_{ji}^{\ell}) &= b_i^{\ell} \quad \forall i, \forall \ell \\
x_{ij}^{\ell} &\in \{0, 1\} \quad \forall (i, j), \forall \ell
\end{align*}
\]

Repeatedly used arcs have higher (virtual) cost.

Unused arcs might become more attractive.
Lagrangian Relaxation

Solving the Dual

\[ L^*(u) := \min_x \ -u \ 1 + (c + u \ A)x \]
\[ \text{s.t.} \quad x \in P \]

\[ D : \max_u \ L^*(u) \]
\[ \text{s.t.} \quad u \geq 0 \]

Notice that \( L^*(u) \) is easy to evaluate for any value of \( u \), and so we attempt to get good lower bounds for \( \text{OP} \) by designing an algorithm to solve the dual problem \( D \).

Which algorithm could we choose?
The dual is a concave maximization problem.

The dual function $L^*(u)$ is piece-wise linear.

\[
L^*(u) = \min_{x \in P} L(x, u)
\]
\[
= \min\{-u 1 + (c + u A)x^t : t = 1, \ldots, T\}
\]

We may use a subgradient method to solve the dual.
Lagrangian Relaxation

Properties of the Dual

\[(c + uA)x^k\]

\[(c + uA)x^k\]

\[(c + uA)x^k\]

\[(c + uA)x^k\]
Theorem 1

Let $x^*$ be an optimal solution to $\min_{x \in P} L(x, u)$, for some $u \geq 0$. Then, $Ax^* - 1$ is a subgradient of $L^*(\cdot)$ in $u$.

Remember $d$ is a subgradient of $L^*$ in $u$ iff $L^*(v) - L^*(u) \leq d(v - u)$ for all $v$. 

© 2002 Massachusetts Institute of Technology.
\[ L^*(v) - L^*(u) = \min_{x \in P} L(x, v) - \min_{x \in P} L(x, u) \]

\[ = \min_{x \in P} L(x, v) - L(x^*, u) \]

\[ \leq L(x^*, v) - L(x^*, u) \]

\[ = (c x^* + v (A x^* - 1)) - (c x^* + u (A x^* - 1)) \]

\[ = (A x^* - 1) (v - u) . \]
Lagrangian Relaxation

Subgradient Algorithm

\[ u_{ij}^{h+1} := u_{ij}^h + \alpha_h \left( \sum_{\ell=1}^k \left( x_{ij}^\ell + x_{ji}^\ell \right) - 1 \right) \]

Theorem 2 (Polyak 1967)
Let \( L^*(u) \) be concave and bounded from above. If the sequence of step-lengths \( (\alpha_h)_{h \in \mathbb{N}} \) satisfies \( \alpha_h > 0, \lim_{h \to \infty} \alpha_h = 0 \), and \( \sum_h \alpha_h = \infty \), then the subgradient method converges to the maximum.
Lagrangian Relaxation

Typical Run

- Lower bound
- Number of multiply used edges

Best possible Lagrangian bound = 8303.5
Best bound reached = 6257.8

Sample trajectory using the harmonic series $\left(\frac{1}{n}\right)_n$
Use linear approximation $L(u) = cx(h) + u(Ax(h) - 1)$ to $L^*(u)$.

Let $L^*$ be the optimum value of the Lagrangian dual.

Set $\theta_h$ such that

$L(u^{h+1}) = cx(h) + (u^h + \theta_h(Ax(h) - 1))(Ax(h) - 1) = L^*$.

Hence, $\theta_h = \frac{L^* - L^*(u^h)}{\|Ax(h) - 1\|^2}$.

Therefore,

$\theta_h = \frac{\lambda_h(UB - L^*(u^h))}{\|Ax(h) - 1\|^2}$. 
**Lagrangian Relaxation**

Typical Run

- **Lower bound**
- **Number of multiply used edges**

- Best possible Lagrangian bound = 8303.5
- Best bound reached = 8303.08

Sample trajectory using the Newton step update

Almost feasible solutions

© 2002 Massachusetts Institute of Technology.
## Computational Results

### Comparison of Lower Bounds

<table>
<thead>
<tr>
<th>time[sec]</th>
<th>cheapest paths</th>
<th>Lagrangian</th>
<th>LP relaxation</th>
<th>optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bound time</td>
<td>bound time</td>
<td>bound time</td>
<td>value time</td>
</tr>
<tr>
<td>ger17e</td>
<td>54215 0.0</td>
<td>56336.5 1.3</td>
<td>56340.5 208</td>
<td>56467 390</td>
</tr>
<tr>
<td>ger17n</td>
<td>54679 0.0</td>
<td>58949.0 0.3</td>
<td>58949.0 321</td>
<td>58949 321</td>
</tr>
<tr>
<td>ger73e</td>
<td>88969 0.0</td>
<td>92132.0 1.7</td>
<td>92132.0 11198</td>
<td>92132 11198</td>
</tr>
<tr>
<td>ger73n</td>
<td>89601 0.0</td>
<td>98146.0 1.1</td>
<td>98146.0 502</td>
<td>98146 502</td>
</tr>
<tr>
<td>ger16e</td>
<td>53321 0.0</td>
<td>55265.8 2.9</td>
<td>55266.0 1771</td>
<td>55266 1771</td>
</tr>
<tr>
<td>ger16n</td>
<td>53423 0.0</td>
<td>57024.2 2.4</td>
<td>57086.0 1323</td>
<td>57086 1323</td>
</tr>
<tr>
<td>ger28e</td>
<td>80356 0.0</td>
<td>83095.0 4.6</td>
<td>83095.0 5634</td>
<td>83095 5634</td>
</tr>
<tr>
<td>ger28n</td>
<td>80125 0.0</td>
<td>87348.0 5.3</td>
<td>87404.4 8903</td>
<td>87498 9108</td>
</tr>
<tr>
<td>usa7e</td>
<td>60810 0.0</td>
<td>61498.8 0.6</td>
<td>61499.0 11</td>
<td>61499 11</td>
</tr>
<tr>
<td>usa7n</td>
<td>61214 0.0</td>
<td>66354.0 0.6</td>
<td>66398.0 8</td>
<td>69942 168</td>
</tr>
<tr>
<td>usa12e</td>
<td>67835 0.0</td>
<td>69898.9 0.5</td>
<td>69899.0 68</td>
<td>69899 162</td>
</tr>
<tr>
<td>usa12n</td>
<td>67835 0.0</td>
<td>73092.4 0.7</td>
<td>73104.7 43</td>
<td>74556 121</td>
</tr>
<tr>
<td>usa26e</td>
<td>104293 0.0</td>
<td>109493.7 3.9</td>
<td>109494 5029</td>
<td>109494 5029</td>
</tr>
<tr>
<td>usa26n</td>
<td>105108 0.0</td>
<td>123412.4 3.5</td>
<td>123601.1 2659</td>
<td>123835 5997</td>
</tr>
<tr>
<td>usa37e</td>
<td>105412 0.0</td>
<td>112486.0 2.1</td>
<td>112486.0 18036</td>
<td>112486 18252</td>
</tr>
<tr>
<td>usa37n</td>
<td>105412 0.0</td>
<td>121550.0 2.0</td>
<td>121550.0 10961</td>
<td>121550 10961</td>
</tr>
</tbody>
</table>
Theorem 3

$$\max_{u \geq 0} L^*(u) \geq \text{LP value}.$$ 

Equality holds if the polyhedron defined by $N x = b$, $0 \leq x \leq 1$ is integer.
\[
\max_{u \geq 0} L^*(u) = \max_{u \geq 0} \min_{N x = b, x \in \{0,1\}} L(x, u) \\
\geq \max_{u \geq 0} \min_{N x = b, x \geq 0} L(x, u) \\
= \max_{u \geq 0} \min_{N x = b, x \geq 0} (-u1 + (c + uA)x) \\
= \max_{u \geq 0} (-u1 + \min_{N x = b, x \geq 0} (c + uA)x) \\
= \max_{u \geq 0} (-u1 + \max_{y N \leq c + uA} yb) = \ldots
\]
Comparison of Lower Bounds

\[ \cdots = \max_{yN - uA \leq c, u \geq 0} (-u_1 + yb) \]

\[ = \min_{Nx = b, Ax \leq 1, x \geq 0} cx \]

\[ = \text{optimal LP value.} \]
Branch & Bound Ingredients

- Branching Strategy
- Exploration Strategy
- Lower Bounds
- Upper Bounds

[see picture]

Best Bound Search.
Lagrangian Dual.
Lagrangian Dual !!!
### Branch & Bound vs. CPLEX 3.0

<table>
<thead>
<tr>
<th>Time [min]</th>
<th>Size</th>
<th>Branch &amp; Bound</th>
<th>CPLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>ger17e</td>
<td>164</td>
<td>7.1</td>
<td>6.5</td>
</tr>
<tr>
<td>ger17n</td>
<td>nodes</td>
<td>0.6</td>
<td>5.4</td>
</tr>
<tr>
<td>ger73e</td>
<td>386</td>
<td>0.2</td>
<td>186.6</td>
</tr>
<tr>
<td>ger73n</td>
<td>edges</td>
<td>0.0</td>
<td>8.4</td>
</tr>
<tr>
<td>ger16e</td>
<td>434</td>
<td>8.2</td>
<td>29.5</td>
</tr>
<tr>
<td>ger16n</td>
<td>nodes</td>
<td>33.8</td>
<td>22.1</td>
</tr>
<tr>
<td>ger28e</td>
<td>978</td>
<td>13.5</td>
<td>93.9</td>
</tr>
<tr>
<td>ger28n</td>
<td>edges</td>
<td>94.8</td>
<td>151.8</td>
</tr>
<tr>
<td>usa7e</td>
<td>176</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>usa7n</td>
<td>nodes</td>
<td>0.6</td>
<td>2.8</td>
</tr>
<tr>
<td>usa12e</td>
<td>314</td>
<td>0.1</td>
<td>2.7</td>
</tr>
<tr>
<td>usa12n</td>
<td>edges</td>
<td>0.1</td>
<td>2.0</td>
</tr>
<tr>
<td>usa26e</td>
<td>617</td>
<td>0.1</td>
<td>83.8</td>
</tr>
<tr>
<td>usa26n</td>
<td>nodes</td>
<td>2.3</td>
<td>100.0</td>
</tr>
<tr>
<td>usa37e</td>
<td>1039</td>
<td>1.8</td>
<td>304.2</td>
</tr>
<tr>
<td>usa37n</td>
<td>edges</td>
<td>0.0</td>
<td>182.7</td>
</tr>
</tbody>
</table>
## Computational Results 2000

### Branch & Bound vs. CPLEX 6.5.3

<table>
<thead>
<tr>
<th>time [sec]</th>
<th>size</th>
<th>branch &amp; bound I</th>
<th>branch &amp; bound II</th>
<th>CPLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>base1e</td>
<td>176 nodes</td>
<td>0.2</td>
<td>0.2</td>
<td>0.5</td>
</tr>
<tr>
<td>base1n</td>
<td>314 edges</td>
<td>196.1</td>
<td>17.65</td>
<td>9.6</td>
</tr>
<tr>
<td>base2e</td>
<td>185 nodes</td>
<td>0.7</td>
<td>0.3</td>
<td>1.0</td>
</tr>
<tr>
<td>base2n</td>
<td>354 edges</td>
<td>21.7</td>
<td>6.1</td>
<td>8.2</td>
</tr>
<tr>
<td>base3e</td>
<td>164 nodes</td>
<td>78.2</td>
<td>6.9</td>
<td>1.3</td>
</tr>
<tr>
<td>base3n</td>
<td>368 edges</td>
<td>5.2</td>
<td>3.9</td>
<td>7.3</td>
</tr>
<tr>
<td>base4e</td>
<td>434 nodes</td>
<td>314.3</td>
<td>20.5</td>
<td>5.2</td>
</tr>
<tr>
<td>base4n</td>
<td>978 edges</td>
<td>152.0</td>
<td>28.3</td>
<td>19.9</td>
</tr>
<tr>
<td>base5e</td>
<td>617 nodes</td>
<td>3.5</td>
<td>1.8</td>
<td>5.1</td>
</tr>
<tr>
<td>base5n</td>
<td>1039 edges</td>
<td>102.8</td>
<td>9.7</td>
<td>43.6</td>
</tr>
<tr>
<td>base6e</td>
<td>6894 nodes</td>
<td>15.0</td>
<td>8.1</td>
<td></td>
</tr>
<tr>
<td>base6n</td>
<td>8072 edges</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reflection

Why was it successful?

- On the technical side:
  - Preprocessing
  - Pruning by infeasibility
  - Recycling of Lagrangian multipliers
  - Efficient solution of shortest paths problems
  - Data structures

- Project in itself:
  - Very good team in Berlin with complementary skills.
  - Competent partners at Deutsche Telekom.
  - Got data early enough.
Preprocessing

676 nodes 258
1107 edges 544
Preprocessing
Did we solve the right problem?
Extensions

- Edge capacities.
  - Dependent edges.
    - Varying bandwidth requirements.
  - Simultaneous embedding of many VPNs.
Typical Steps

- Problem identification.
  - Problem penetration.
    - Modeling.
      - Problem complexity.
    - Solution approach.
  - Implementation.
- Fine Tuning.
- Feedback.
Some of

Today’s Lessons

- Good (= fast **and** close) lower bounds pay off.
- Tradeoff **Lagrangian relaxation** vs. **LP relaxation**.
- Algorithm fine tuning involves mathematical insights as well as engineering.
- **Commercial software** becomes increasingly stronger.