Mathematics of Infrastructure Planning (ADM III)

TU Berlin
Summer Semester 2012
First Lecture on April 12, 2012

Ralf Borndörer & Martin Grötschel

ZIB, TU, and MATHEON, Berlin

Martin Grötschel
- Institut für Mathematik, Technische Universität Berlin (TUB)
- DFG-Forschungszentrum "Mathematik für Schlüsseltechnologien" (MATHEON)
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

groetschel@zib.de http://www.zib.de/groetschel

Mathematics of Infrastructure Planning (ADM III)
Prologue

TU Berlin
Summer Semester 2012
First Lecture on April 12, 2012

Ralf Borndörer & Martin Grötschel

ZIB, TU, and MATHEON, Berlin

Martin Grötschel
- Institut für Mathematik, Technische Universität Berlin (TUB)
- DFG-Forschungszentrum "Mathematik für Schlüsseltechnologien" (MATHEON)
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

groetschel@zib.de http://www.zib.de/groetschel

Remark

- This set ppt-slides has basically the same contents as the set shown in the classes on April 12 and 16. However, for copyright purposes some pictures, in particular those, whose origin/copyright holders could not be correctly identified, have been removed.

Martin Grötschel

Why LP / IP / MIP Survey?

- Almost all infrastructure planning problems discussed in this class are of some combinatorial nature - plus some nonlinearities and stochastic aspects.
- In most of the cases treated, in the end, some integer or mixed-integer programs have to be solved.
- In almost all of the solution approaches linear programming problems arise, usually as sub-problems and very often of very large scale.
- It is therefore necessary to understand the LP/IP/MIP solution technology and to know what which approaches are able to "deliver".
- That is why we start with this survey (and provide a brief preview of some of the topics to be covered).

Martin Grötschel

What drives my research

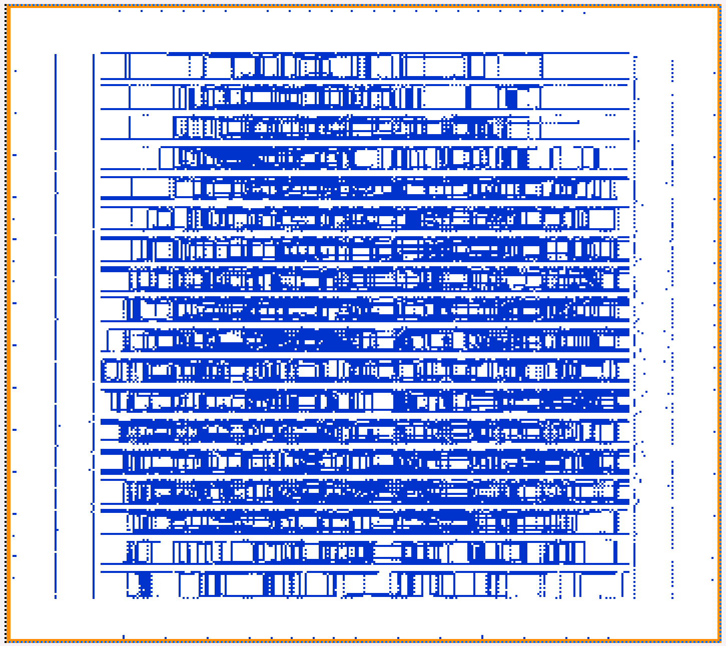
- I like integer programming and combinatorial optimization.
- I am really interested in real applications.
- The world is full of important, difficult, and very large scale optimization problem. I like to contribute to their solution.
- Almost every practically relevant problem creates new research problems that need new theory.
- Application driven approach is an excellent way to combine theory and practice.
- ZIB: We make our algorithmic advances freely available.

Martin Grötschel

What is large scale?

LP/IP Sizes:

- Largest TSP solved to optimality: 85900 cities
- Largest TSP "in operation"

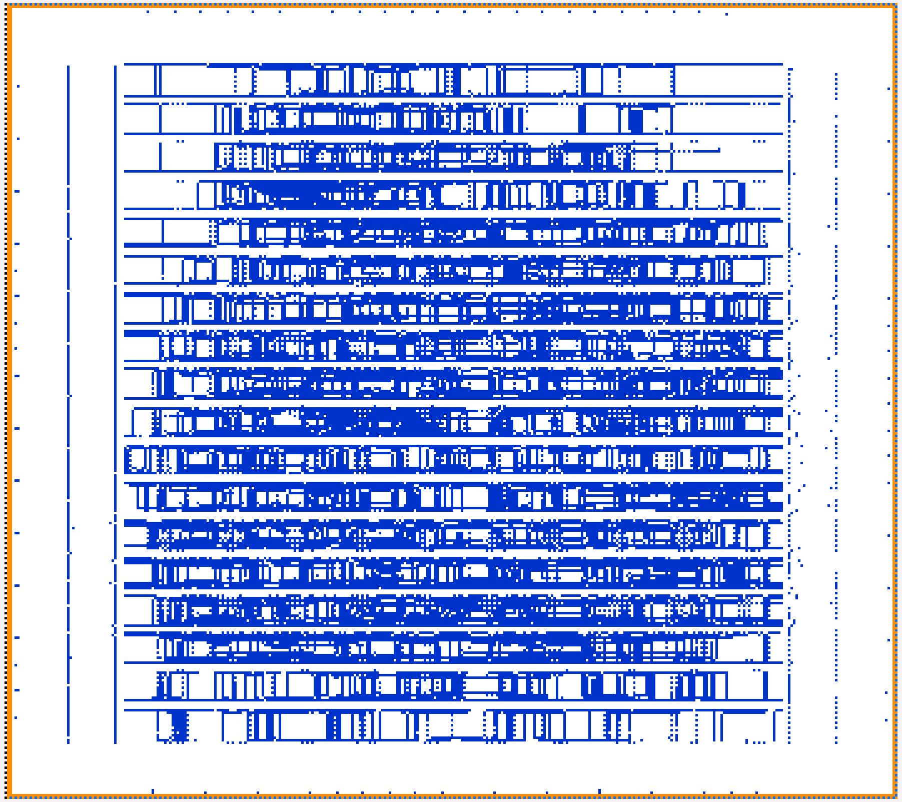


85,900 Locations in a VLSI Application
Solved in 2006

- See http://www.tsp.gatech.edu/optimal/index.html

LP / IP Sizes handled: the TSP case

- Largest TSP solved to optimality: 85,900 cities (in 2006)
- Number of integer variables:
  3,689,362,050
  ~ 3.5 x 10^9

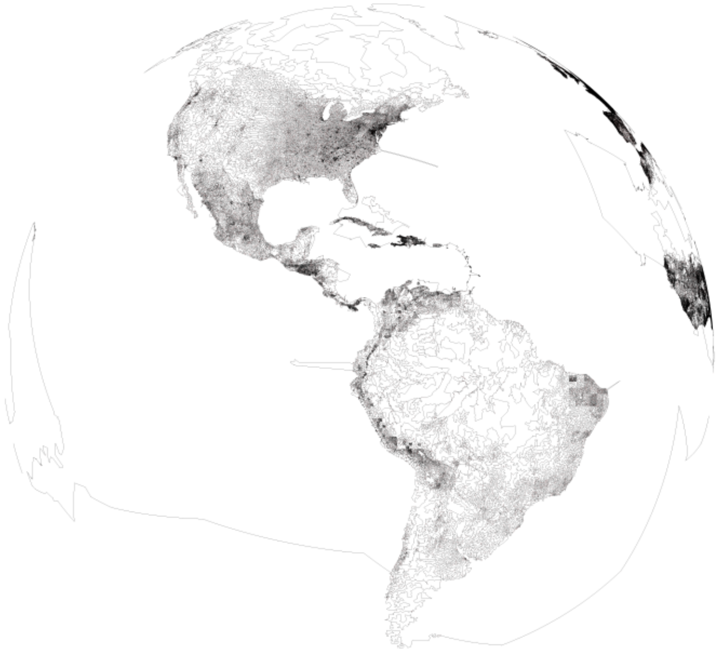


85,900 Locations in a VLSI Application
Solved in 2006

Martin Grötschel

LP / IP Sizes handled: the TSP case

- Largest TSP "in operation": 1,904,711 cities
- Number of integer variables: 1,813,961,044,405 ~ 1.8 x 10^{12}
- Number of constraints (conservative lower bound): $2^{1,904,710}$ ~ $10^{573,317}$



- http://www.tsp.gatech.edu/world/pictures.html

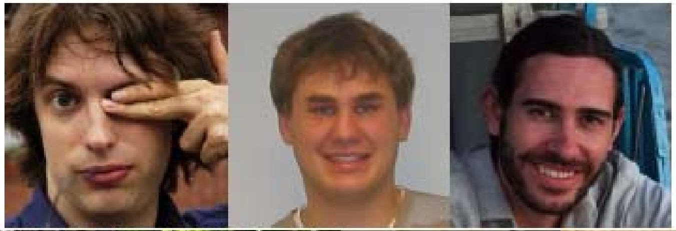
best known solution
optimality gap below 0.1%

Martin Grötschel

ZIB LP/MIP Group

- ▷ Tobias Achterberg (IBM)
- ▷ Thorsten Koch
- ▷ Marc Pfetsch (TU Braunschweig)

- ▷ Timo Berthold
- ▷ Gerald Gamrath
- ▷ Ambros Gleixner
- ▷ Stefan Heinz
- ▷ Yuji Shinano
- ▷ Stefan Vigerske
- ▷ Kati Wolter



- ▷ Gregor Hendel
- ▷ Robert Waniek
- ▷ Michael Winkler

Einstein Center for Mathematics Berlin (application to be decided in May 2012)

- Innovation Area
  „Mathematics in Metropolitan Infrastructure"
  Skutella Lecture (March 21, 2012)

Martin Grötschel

Mathematics of Infrastructure Planning (ADM III)
Part I: Solving Linear Programs

TU Berlin
Summer Semester 2012
First Lecture on April 12, 2012

Ralf Borndörer & Martin Grötschel

ZIB, TU, and MATHEON, Berlin

Martin Grötschel
- Institut für Mathematik, Technische Universität Berlin (TUB)
- DFG-Forschungszentrum "Mathematik für Schlüsseltechnologien" (MATHEON)
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

groetschel@zib.de http://www.zib.de/groetschel

Contents

Martin Grötschel

Contents

typical optimization problems

$$\max\ f(x)\ \ or\ \min\ f(x)$$
$$g_i(x) = 0, \quad i = 1, 2, \ldots, k$$
$$h_j(x) \leq 0, \quad j = 1, 2, \ldots, m$$
$$x \in \mathbb{R}^n \text{ (and } x \in S\text{)}$$

„general" (nonlinear) program NLP

$$\min c^T x$$
$$Ax = a$$
$$Bx \leq b$$
$$x \geq 0$$
$$(x \in \mathbb{R}^n)$$
$$(x \in \Bbbk^n)$$

linear program LP

$$\min c^T x$$
$$Ax = a$$
$$Bx \leq b$$
$$x \geq 0$$
$$\text{some } x_j \in \mathbb{Z}$$
$$(x \in \{0,1\}^n)$$

(linear) 0/1-mixed-integer program IP, MIP

program = optimization problem

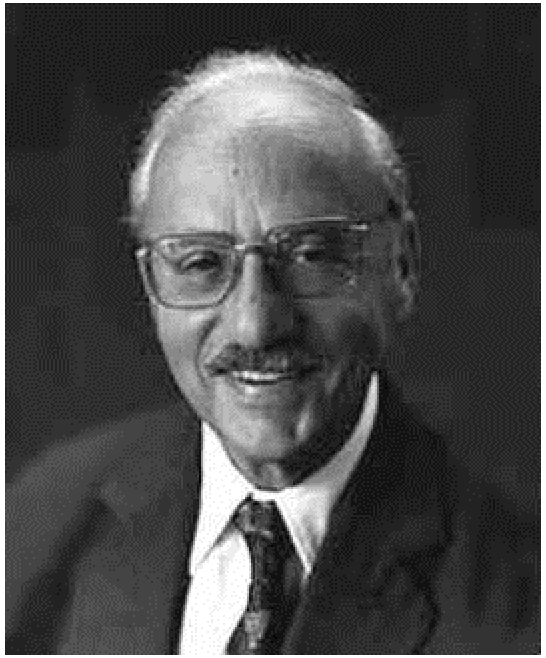
Martin Grötschel

Contents

Linear Programming: a very brief history

- 1826/1827 Jean Baptiste Joseph Fourier (1786-1830): rudimentory form of the simplex method in 3 dimensions.
- 1939 L. V. Kantorovitch (1912-1986): Foundations of linear programming (Nobel Prize 1975)
- 1947 G. B. Dantzig (1914-2005): Invention of the (primal) simplex algorithm
- 1954 C.E. Lemke: Dual simplex algorithm
- 1953 G.B. Dantzig,
  1954 W. Orchard Hays, and
  1954 G. B. Dantzig & W. Orchard Hays:
  Revised simplex algorithm

$$\max c^T x$$
$$Ax = b$$
$$x \geq 0$$



Dantzig and Bixby

George Dantzig and
Bob Bixby
(founder of CPLEX and GUROBI)
at the International Symposium on Mathematical Programming,
Atlanta, August 2000

This lecture employs a lot of information I obtained from Bob and some of his slides.

Martin Grötschel

Optimal use of scarce ressources

foundation and economic interpretation of LP

Leonid V. Kantorovich Tjalling C. Koopmans

Nobel Prize for Economics 1975

Martin Grötschel

Stiglers „Diet Problem“: „The first linear program“

Min $x1 + x2$	costs
$2x1 + x2 \geq 3$	protein
$x1 + 2x2 \geq 3$	carbohydrates
$x1 \geq 0$	potatoes
$x2 \geq 0$	beans

minimizing the cost of food

George J. Stigler
Nobel Prize in economics 1982

Sets n nutrients / calorie thousands , protein grams , calcium grams , iron milligrams vitamin-a thousand ius, vitamin-b1 milligrams, vitamin-b2 milligrams, niacin milligrams , vitamin-c milligrams /

f foods / wheat , cornmeal , cannedmilk, margarine , cheese , peanut-b , lard liver , porkroast, salmon , greenbeans, cabbage , onions , potatoes spinach, sweet-pot, peaches , prunes , limabeans, navybeans /

Parameter b(n) required daily allowances of nutrients / calorie 3, protein 70 , calcium .8 , iron 12 vitamin-a 5, vitamin-b1 1.8, vitamin-b2 2.7, niacin 18, vitamin-c 75 /

Table a(f,n) nutritive value of foods (per dollar spent)

	calorie	protein	calcium	iron	vitamin-a	vitamin-b1	vitamin-b2	niacin	vitamin-c
	(1000)	(g)	(g)	(mg)	(1000iu)	(mg)	(mg)	(mg)	(mg)
wheat	44.7	1411	2.0	365		55.4	33.3	441	
cornmeal	36	897	1.7	99	30.9	17.4	7.9	106	
cannedmilk	8.4	422	15.1	9	26	3	23.5	11	60
margarine	20.6	17	.6	6	55.8	.2			
cheese	7.4	448	16.4	19	28.1	.8	10.3	4	
peanut-b	15.7	661	1	48		9.6	8.1	471	
lard	41.7				.2		.5	5	
liver	2.2	333	.2	139	169.2	6.4	50.8	316	525
porkroast	4.4	249	.3	37		18.2	3.6	79	
salmon	5.8	705	6.8	45	3.5	1	4.9	209	
greenbeans	2.4	138	3.7	80	69	4.3	5.8	37	862
cabbage	2.6	125	4	36	7.2	9	4.5	26	5369
onions	5.8	166	3.8	59	16.6	4.7	5.9	21	1184
potatoes	14.3	336	1.8	118	6.7	29.4	7.1	198	2522
spinach	1.1	106		138	918.4	5.7	13.8	33	2755
sweet-pot	9.6	138	2.7	54	290.7	8.4	5.4	83	1912
peaches	8.5	87	1.7	173	86.8	1.2	4.3	55	57
prunes	12.8	99	2.5	154	85.7	3.9	4.3	65	257
limabeans	17.4	1055	3.7	459	5.1	26.9	38.2	93	
navybeans	26.9	1691	11.4	792		38.4	24.6	217	

Positive Variable x(f) dollars of food f to be purchased daily (dollars)

Free Variable cost total food bill (dollars)

Equations nb(n) nutrient balance (units), cb cost balance (dollars) ;

nb(n).. sum(f, a(f,n)*x(f)) =g= b(n); cb.. cost=e= sum(f, x(f));

Model diet stiglers diet problem / nb,cb /;

http://www.gams.com/modlib/libhtml/diet.htm

Solution of the Diet Problem

Goal: Find the cheapest combination of foods that will satisfy the daily requirements of a person!

The problem motivated by the army's desire to meet nutritional requirements of the soldiers at minimum cost.

Army's problem had 77 unknowns and 9 constraints.
Stigler solved problem using a heuristic: $39.93/year (1939)
Laderman (1947) used simplex: $39.69/year (1939 prices)

→ first "large-scale computation"
took 120 man days on hand operated desk calculators (10 human "computers")

http://www.mcs.anl.gov/home/otc/Guide/CaseStudies/diet/index.html

Martin Grötschel

Commercial software

William Orchard-Hayes (in the period 1953-1954)

The first commercial LP-Code was on the market in 1954 (almost 60 years ago) and available on an IBM CPC (card programmable calculator):

Code: Simplex Algorithm with explicit basis inverse, that was recomputed in each step.

Shortly after, Orchard-Hayes implemented a version with product form of the inverse (idea of A. Orden),
Record: 71 variables, 26 constraints, 8 h running time

About 1960: LP became commercially viable, used largely by oil companies.

Martin Grötschel

The Decade of the 70's: Theory

- V. Klee and G. J. Minty, „How good is the simplex algorithm?", in O. Shisha (ed.), Inequalities III, Academic Press, New York, 1972, 159-172
- K. H. Borgwardt, „Untersuchungen zur Asymptotik der mittleren Schrittzahl von Simplexverfahren in der linearen Optimierung", Dissertation, U Kaiserslautern, 1977
- L. G. Khachiyan, „A polynomial algorithm in linear programming", (Russian), Doklady Akademii Nauk SSR 244 (1979) 1093-1096

The Decade of the 70's: Practice

- Interest in optimization flowered
  - Large scale planning applications particularly popular
- Significant difficulties emerged
  - Building applications was very expensive and very risky
  - Technology just wasn't ready:
    - LP was slow and
    - Mixed Integer Programming was impossible.
- OR could not really "deliver" – with some exceptions, of course
- The ellipsoid method of 1979 was no practical success.

The Decade of the 80’s and beyond

- Mid 80’s:
  - There was perception was that LP software had progressed about as far as it could.
- There were several key developments
  - IBM PC introduced in 1981
    - Brought personal computing to business
  - Relational databases developed. ERP systems introduced.
  - 1984, major theoretical breakthrough in LP
    N. Karmarkar, “A new polynomial-time algorithm for linear programming”, Combinatorica 4 (1984) 373-395
    (Interior Point Methods, front page New York Times)
- The last ~20 years: Remarkable progress
  - We now have three competitive algorithms:
    Primal & Dual Simplex, Barrier (interior points)

My opinion on Linear Programming

- From an commercial/economic point of view:

Linear programming is the most important development of mathematics in the 20th century.

Martin Grötschel

Contents

Contents

Martin Grötschel

Linear Programming

$$\begin{aligned} &\min c^T x \\ &Ax = a \\ &Bx \le b \\ &x \ge 0 \end{aligned}$$

$$\begin{aligned} &\min c^T x \\ &Ax = a \\ &Bx \le b \\ &x \ge 0 \end{aligned}$$

$$\begin{aligned} &\min c^T x \\ &Ax = a \\ &x \ge 0 \end{aligned}$$

$$\begin{aligned} &\min c^T x \\ &Bx \le b \end{aligned}$$

Linear program in various forms.
They are all equivalent!
There are more versions!

Martin Grötschel

Optimizers‘ dream: Duality theorems

- Max-Flow Min-Cut Theorem

  The value of a maximal (s,t)-flow in a capacitated network is equal to the minimal capacity of an (s,t)-cut.

- The Farkas Lemma

- The Duality Theorem of Linear Programming

$$\begin{array}{lcl} \max c^T x & = & \min y^T b \\ Ax \leq b & & y^T A \geq c^T \\ x \geq 0 & & y \geq 0 \end{array}$$

Important theorems

- Complementary slackness theorems
- Redundancy characterizations
- Polyhedral theory

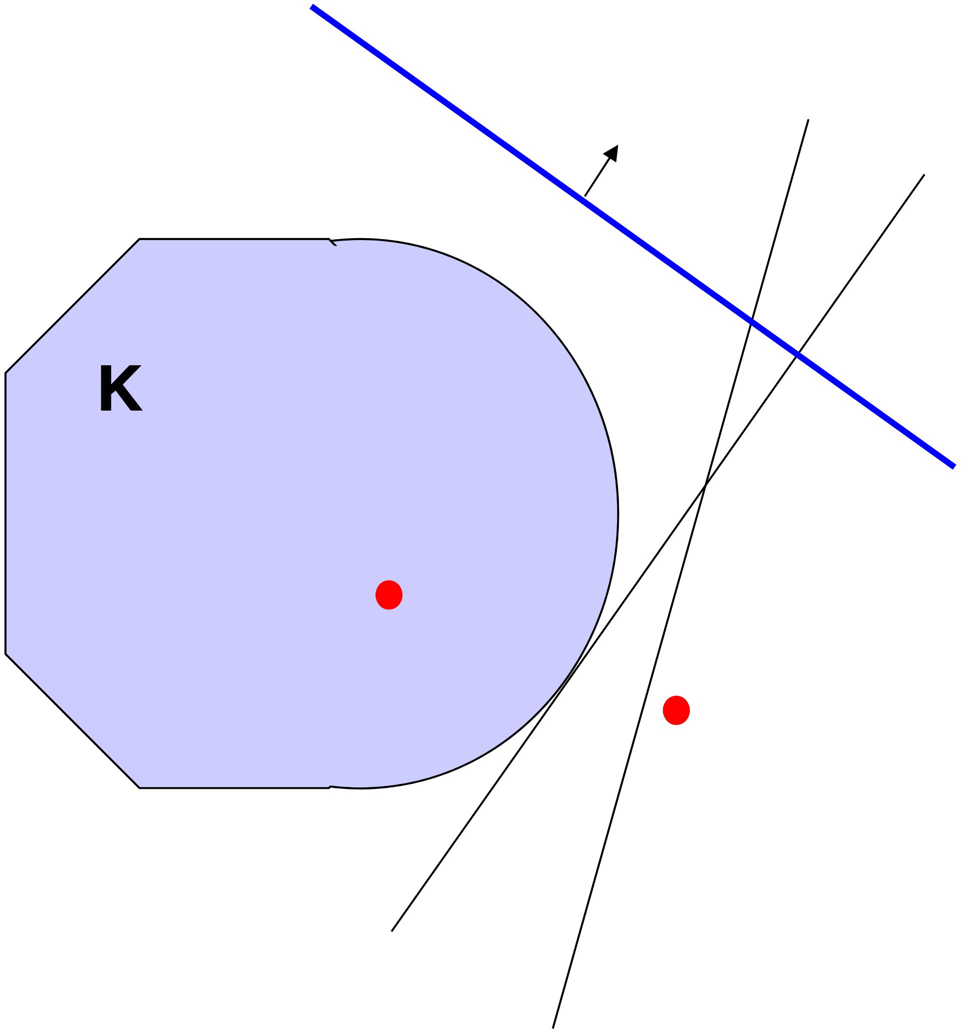
Martin
Grötschel

LP Solvability

- I assume that the audience is somewhat familiar with complexity theory:
  - Polynomial time solvability, solvability in strongly polynomial time
  - Classes: $\mathcal{P}$ and $\mathcal{NP}$, $\mathcal{NP}$-completeness , $\mathcal{NP}$-hardness , etc.
- Linear programs can be solved in polynomial time with
  - the Ellipsoid Method (Khachiyan, 1979)
  - Interior Points Methods (Karmarkar, 1984, and others)
- Open: Is there a strongly polynomial time algorithm for the solution of LPs?
- Certain variants of the Simplex Algorithm run – under certain conditions – in expected polynomial time (Borgwardt, 1977…)
- Open: Is there a polynomial time variant of the Simplex Algorithm?

Martin Grötschel

Separation



Martin Grötschel

LP Solvability: Generalizations

Theorem (GLS 1981, 1988) (modulo technical details) **:** There exists a polynomial time algorithm to minimize convex functions (e.g., linear functions) over the elements of a class of convex bodies **K** (e. g. polyhedra) if and only if, there exists a polynomial time algorithm that decides, for any given point **x**, whether **x** is in **K**, and that, when **x** is not in **K**, finds a hyperplane that separates **x** from **K**.

Short version:

Optimization and Separation are polynomial-time equivalent.

Consequence: Theoretical Foundation of cutting plane algorithms.

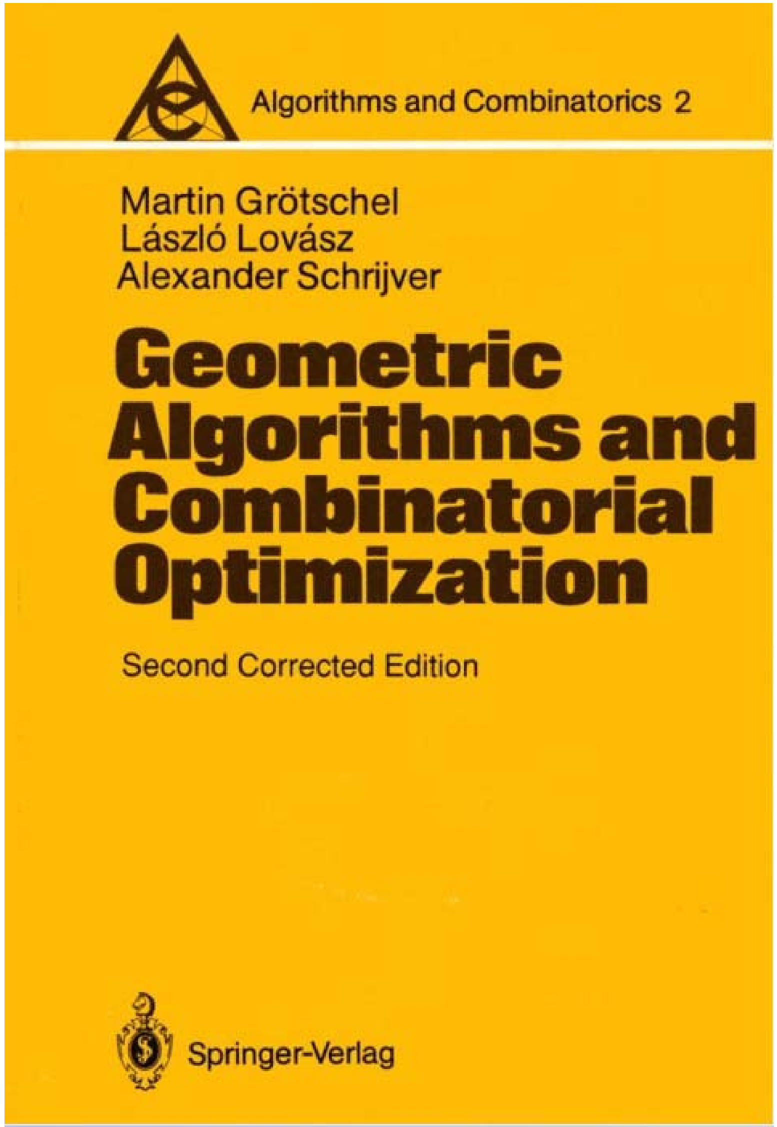
Particular special case: Polynomial time separation algorithm for the set of positive semi-definite matrices.

Consequences:

- Polynomial time algorithm for stable sets in perfect graphs.
- The beginning of semi-definite programming

Martin Grötschel

You can download this book from the publications list on my Web page.



http://www.zib.de/groetschel/pubnew/paper/groetschellovaszschrijver1988.pdf

Contents

Martin Grötschel

Algorithms for nonlinear programming

- Iterative methods that solve the equation and inequality sytems representing the necessary local optimality conditions (e.g., KKT).

$$x_{i+1} = x_i + \lambda_i d_i$$
$$d_i \sim \text{"descent direction"}$$
$$\lambda_i \sim \text{"steplength"}$$

$$d_i = -\nabla f(x_i)$$ Steepest descent

$$d_i = -(H(x_i))^{-1} \nabla f(x_i)$$ Newton

(Quasi-Newton, conjugate-gradient-, SQP-, subgradient…methods)

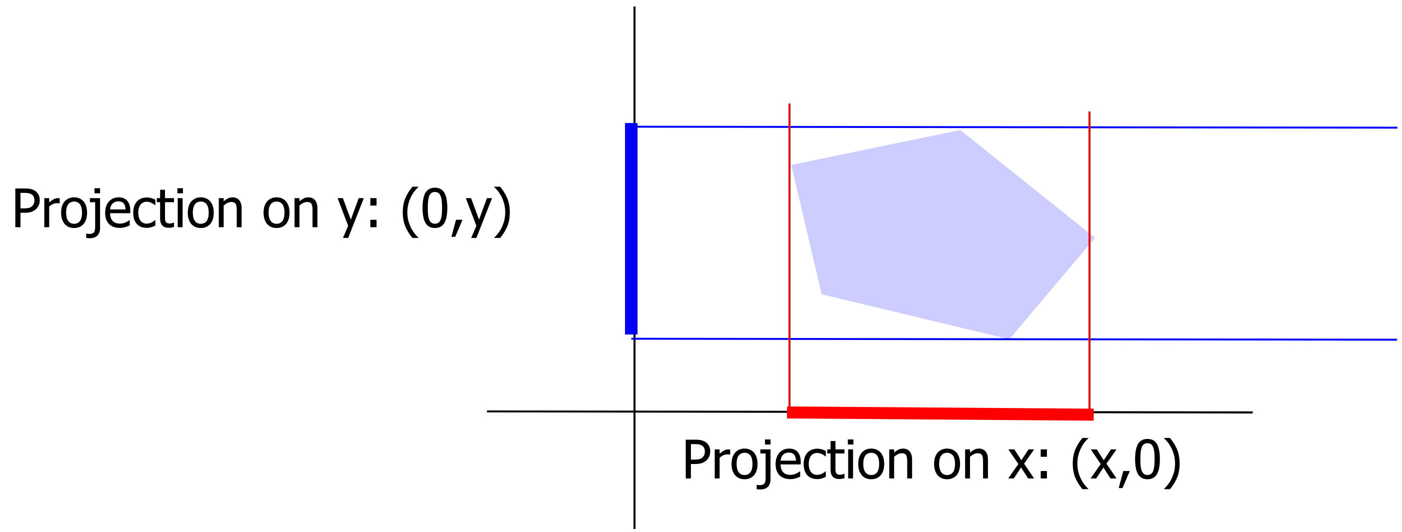
- Sufficient conditions are rarely checked.

Martin Grötschel

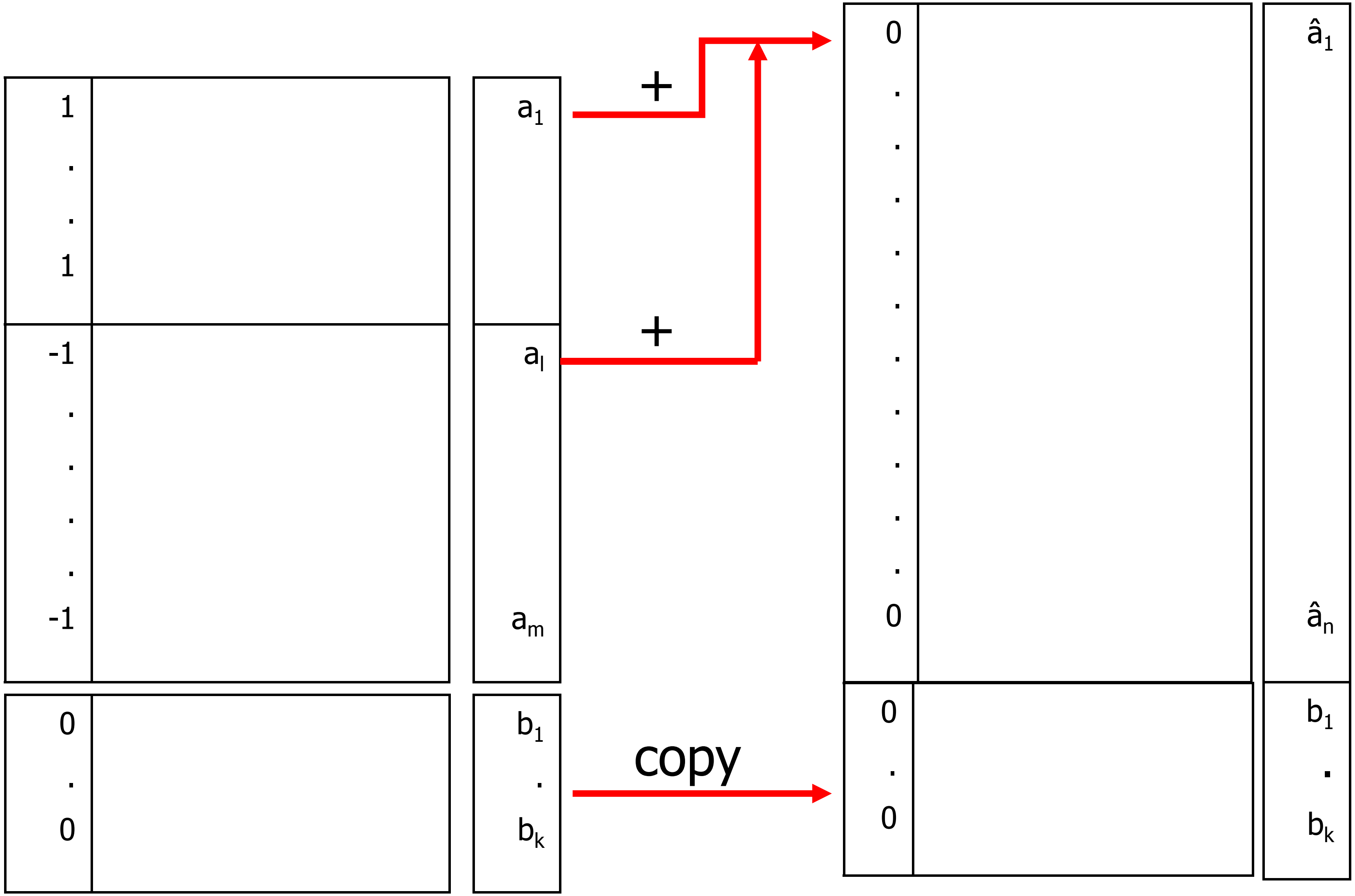
Contents

Fourier-Motzkin Elimination

- Fourier, 1826/1827
- Motzkin, 1938
- **Method**: successive projection of a polyhedron in n-dimensional space into a vector space of dimension n-1 by elimination of one variable.



A Fourier-Motzkin step

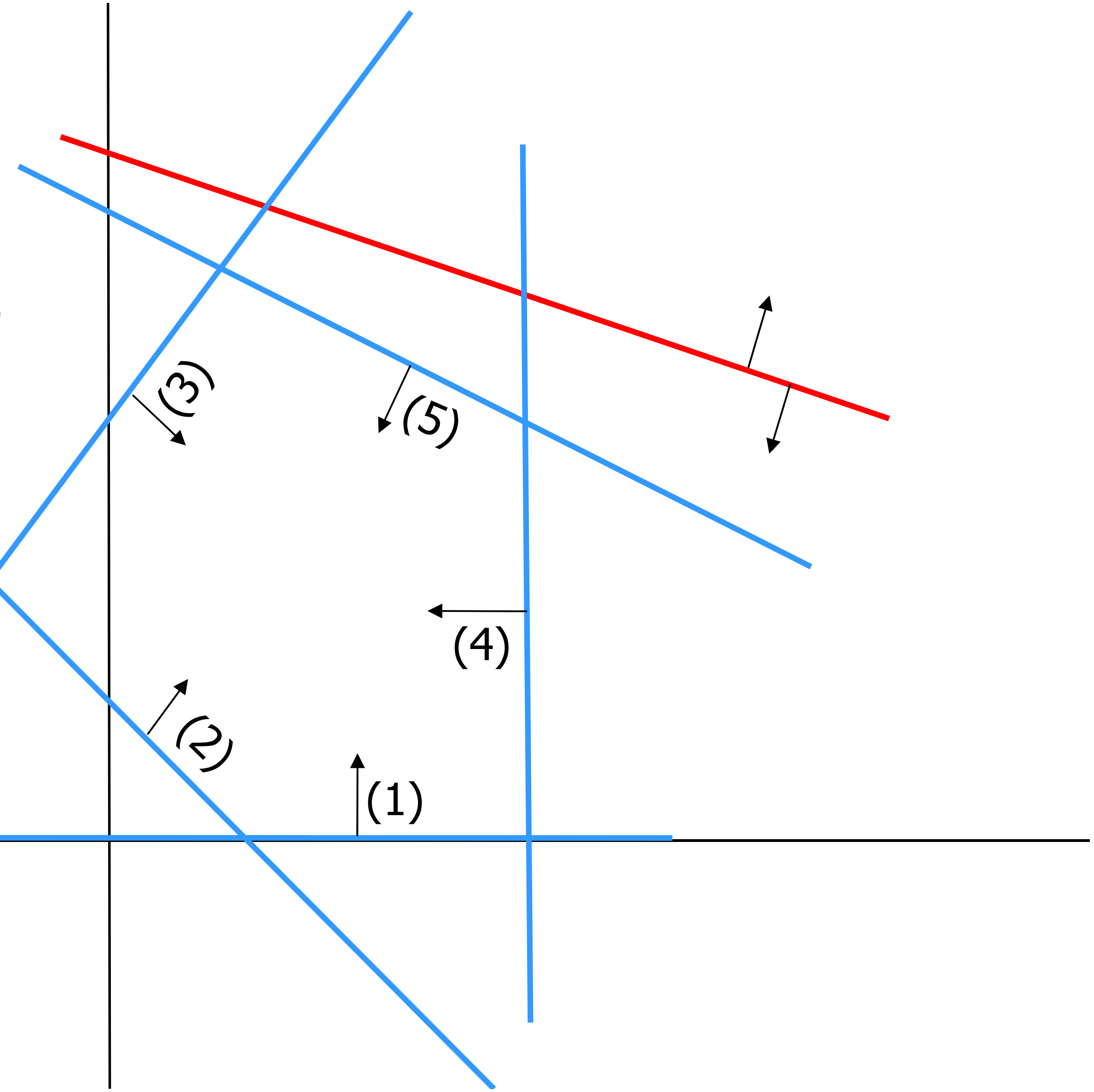


Martin Grötschel

Fourier-Motzkin Elimination: an example

```
min/max + x1 + 3x2

(1)        -   x2 <= 0
(2) - x1 -   x2 <=-1
(3) - x1 +   x2 <= 3
(4) + x1        <= 3
(5) + x1 + 2x2 <= 9
```



Martin Grötschel

Fourier-Motzkin Elimination: an example, call of PORTA (Polymake)

```
min/max + x1 + 3x2

(1)       -   x2 <= 0
(2) - x1 -   x2 <=-1
(3) - x1 +   x2 <= 3
(4) + x1         <= 3
(5) + x1 + 2x2 <= 9
```

```
DIM = 3

INEQUALITIES_SECTION

(1)       -   x2        <= 0
(2) - x1 -   x2        <=-1
(3) - x1 +   x2        <= 3
(4) + x1               <= 3
(5) + x1 + 2x2        <= 9
(6) + x1 + 3x2 - x3 <= 0
(7) - x1 - 3x2 + x3 <= 0


ELIMINATION_ORDER
1 0 0
```

Martin Grötschel

Fourier-Motzkin Elimination: an example

```
DIM = 3

INEQUALITIES_SECTION

(1)    (1) - x2        <=  0
(2,4)  (2) - x2        <=  2
(2,5)  (3) + x2        <=  8
(2,6)  (4) +2x2 - x3   <= -1
(3,4)  (5) + x2        <=  6
(3,5)  (6) + x2        <=  4
(3,6)  (7) +4x2 - x3   <=  3
(7,4)  (8) -3x2 + x3   <=  3
(7,5)  (9) - x2 + x3   <=  9
(7,6)
```

```
DIM = 3

INEQUALITIES_SECTION

(1)        -  x2        <= 0
(2) - x1 -    x2        <=-1
(3) - x1 +    x2        <= 3
(4) + x1                <= 3
(5) + x1 + 2x2          <= 9
(6) + x1 + 3x2 - x3     <= 0
(7) - x1 - 3x2 + x3     <= 0

ELIMINATION_ORDER
1 0 0
```

Martin Grötschel

Fourier-Motzkin Elimination: an example

```
DIM = 3

INEQUALITIES_SECTION

(1)    (1) - x2         <=  0
(2,4)  (2) - x2         <=  2
(2,5)  (3) + x2         <=  8
(2,6)  (4) +2x2 - x3 <= -1
(3,4)  (5) + x2         <=  6
(3,5)  (6) + x2         <=  4
(3,6)  (7) +4x2 - x3 <=  3
(7,4)  (8) -3x2 + x3 <=  3
(7,5)  (9) - x2 + x3 <=  9
(7,6)

ELIMINATION_ORDER
0 1 0
```

```
(1,4) (  1) -x3 <=  -1
(1,7) (  2) -x3 <=   3
(2,4) (  3) -x3 <=   3
(2,7) (  4) -x3 <=  11
(8,3) (  5) +x3 <=  27
(8,4) (  6) -x3 <=   3
(8,5) (  7) +x3 <=  21
(8,6) (  8) +x3 <=  15
(8,7) (  9) +x3 <=  21
(9,3) ( 10) +x3 <=  17
(9,4) ( 11) +x3 <=  17
(9,5) ( 12) +x3 <=  15
(9,6) ( 13) +x3 <=  13
(9,7) ( 14)+3x3 <=  39
```

min = 1 <= x3 <= 13 = max

x1 = 1
x2 = 0

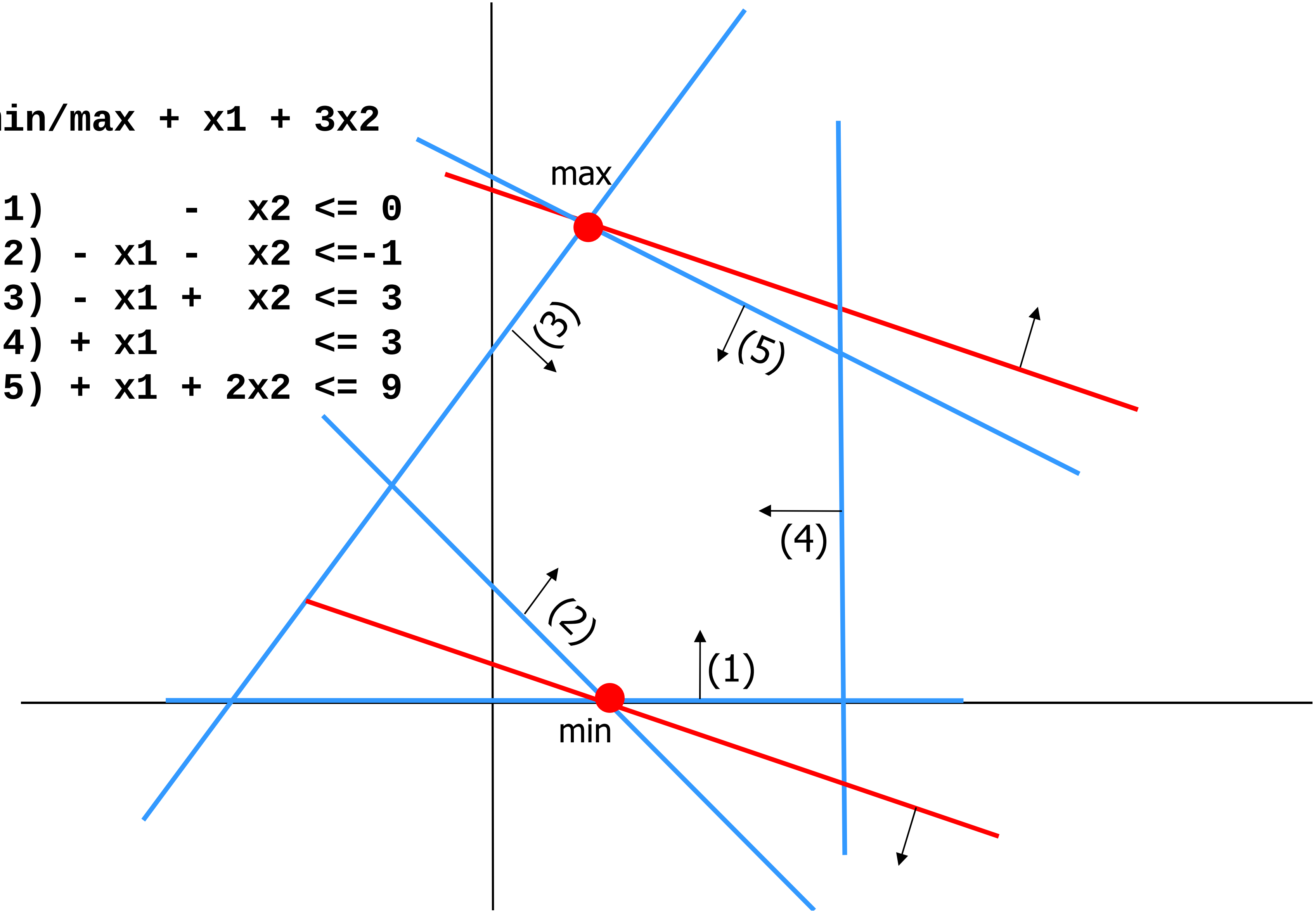
x1 = 1
x2 = 4

Martin Grötschel

Fourier-Motzkin Elimination: an example

```
min/max + x1 + 3x2

(1)         -   x2 <= 0
(2) - x1 -   x2 <=-1
(3) - x1 +   x2 <= 3
(4) + x1         <= 3
(5) + x1 + 2x2 <= 9
```



Martin Grötschel

Fourier-Motzkin Elimination

- FME is a wonderful constructive proof method.
- Elimination of all variables of a given inequality system directly yields the Farkas Lemma:

$Ax \leq b$ has a solution or $y^T A = 0, y^T b < 0$ has a solution but not both.

- FME is computationally lousy.

Martin Grötschel

Contents

Martin Grötschel

The Simplex Method

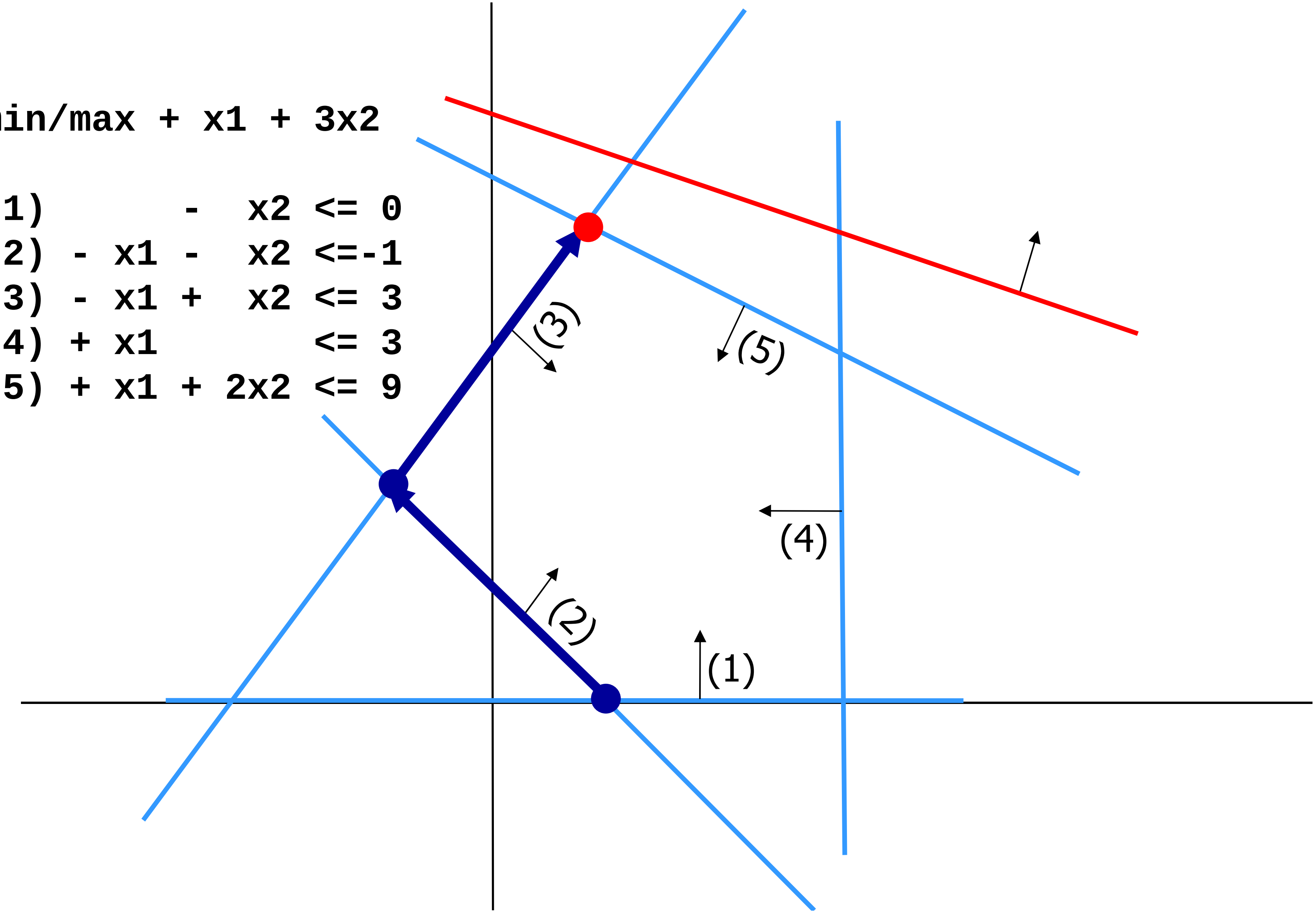
- Dantzig, 1947: primal Simplex Method
- Dantzig, 1953: revised Simplex Method
- Lemke, 1954; Beale, 1954: dual Simplex Method
- ….
- **Underlying Idea**: Find a vertex of the set of feasible LP solutions (polyhedron) and move to a better neighbouring vertex, if possible (Fourier's idea 1826/27).

Martin Grötschel

The Simplex Method: an example

```
min/max + x1 + 3x2

(1)        -   x2 <= 0
(2) - x1 -   x2 <=-1
(3) - x1 +   x2 <= 3
(4) + x1        <= 3
(5) + x1 + 2x2 <= 9
```

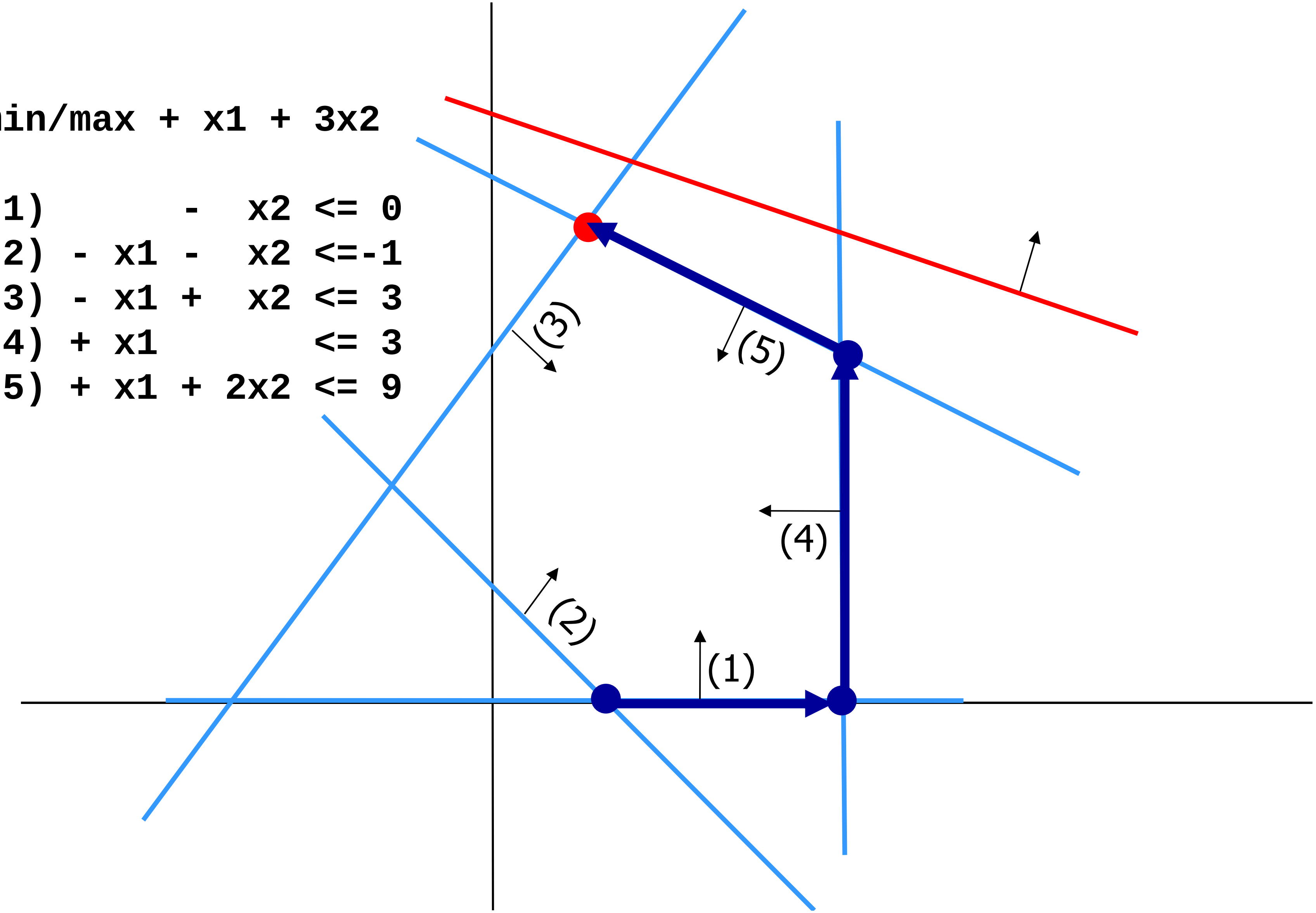


Martin
Grötschel

The Simplex Method: an example

```
min/max + x1 + 3x2

(1)        -   x2 <= 0
(2) - x1 -   x2 <=-1
(3) - x1 +   x2 <= 3
(4) + x1          <= 3
(5) + x1 + 2x2 <= 9
```



Martin Grötschel

Computationally important idea of the Simplex Method

Let a (m,n)-Matrix A with full row rank m, an m-vector b and an n-vector c with m<n be given. For every vertex y of the polyhedron of feasible solutions of the LP,

$$\max c^T x$$
$$Ax = b$$
$$x \geq 0$$

A =

B	N

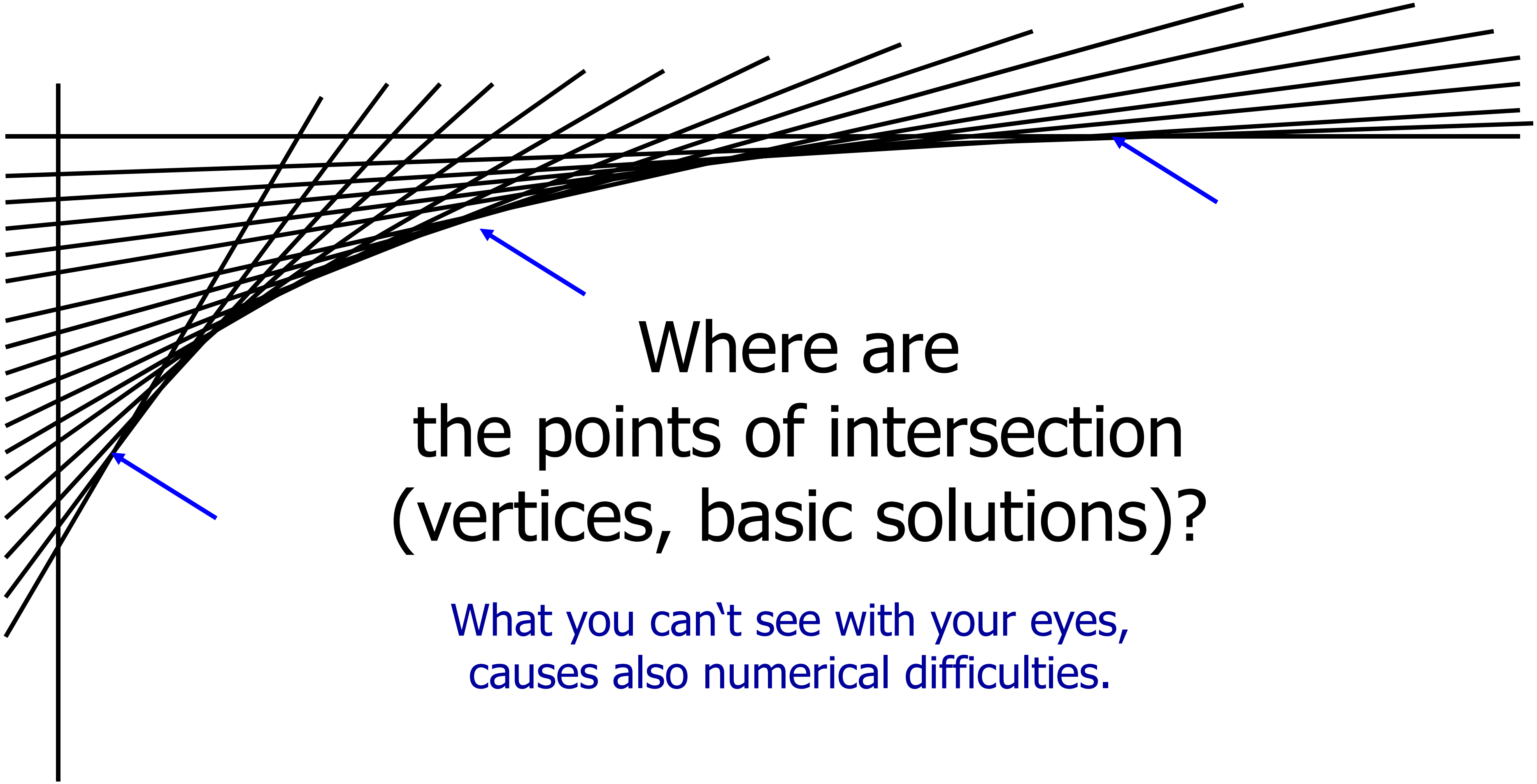
there is a non-singular (m,m)-submatrix B (called basis) of A representing the vertex y (basic solution) as follows

$$y_B = B^{-1}b, \quad y_N = 0$$

Many computational consequences:
Update-formulas, reduced cost calculations, number of non-zeros of a vertex,…

Martin Grötschel

Numerical trouble often has geometric reasons



Where are
the points of intersection
(vertices, basic solutions)?

What you can‘t see with your eyes,
causes also numerical difficulties.

Dual Simplex Method

- The Dual Simplex Method is the (Primal) Simplex Method applied to the dual of a given linear program.

Surprise in the mid-nineties:

- The Dual Simplex Method is faster than the Primal in practice.
  One key: Goldfarb's steepest edge pivoting rule!
- A wonderful observation for the cutting plane methods of integer programming!

Martin Grötschel

Contents

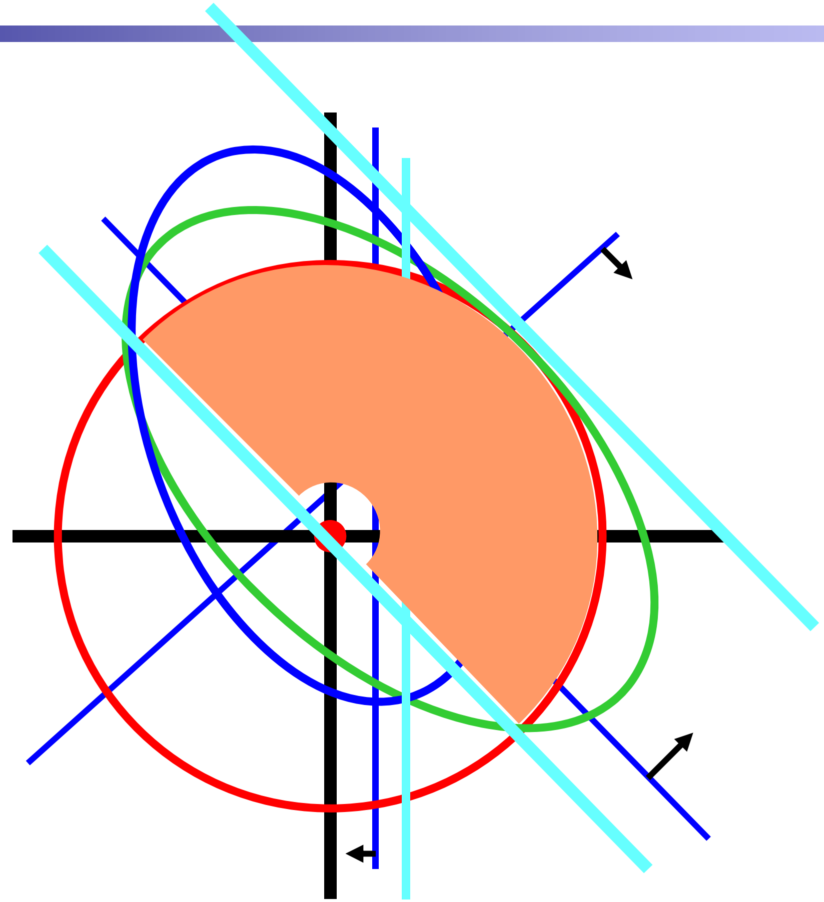
Martin Grötschel

The Ellipsoid Method

- Shor, 1970 - 1979
- Yudin & Nemirovskii, 1976
- Khachiyan, 1979
- M. Grötschel, L. Lovász, A. Schrijver,
  Geometric Algorithms and Combinatorial Optimization
  Algorithms and Combinatorics 2, Springer, 1988

Martin
Grötschel

The Ellipsoid Method: an example



Martin
Grötschel

$$P := \{x \mid Cx \le d\}$$

$$k := 0,$$
$$N := 2n((2n+1)\langle C\rangle + n\langle d\rangle - n^3)$$
$$A_0 := R^2 I \quad \text{with } R := \sqrt{n}\, 2^{\langle C,d\rangle - n^2}$$
$$a_0 := 0$$

Initialization

If $k = N$, *STOP!* (Declare P empty.) — Stopping criterion

If $a_k \in P$, *STOP!* (A feasible solution is found.) — Feasibility check

If $a_k \notin P$, *then choose an inequality, say* $c^T x \le \gamma$, *of the system* $Cx \le d$ *that is violated by* a_k. — Cutting plane choice

$$b := \frac{1}{\sqrt{c^T A_k c}} A_k c$$

$$a_{k+1} := a_k - \frac{1}{n+1} b$$

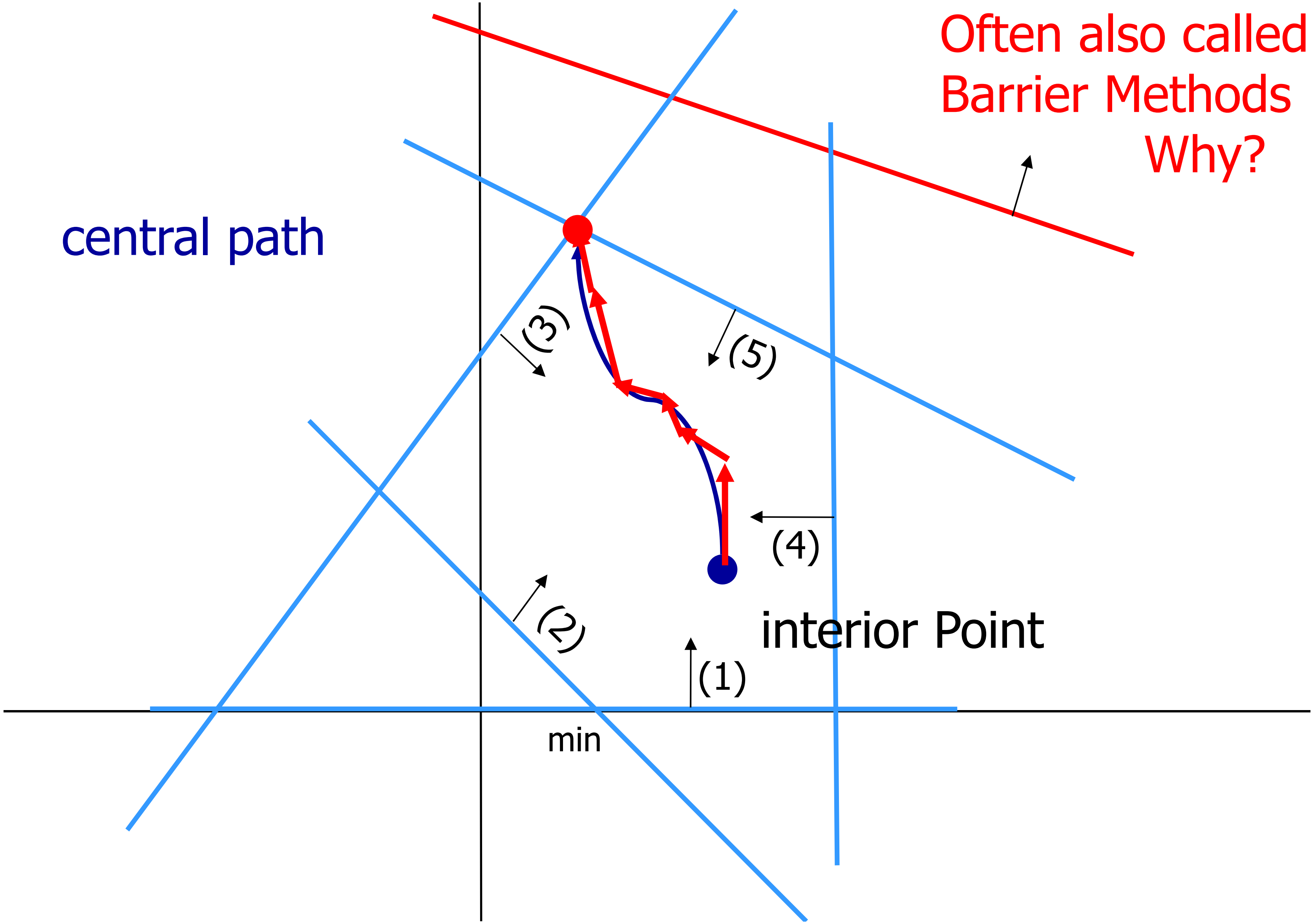
$$A_{k+1} := \frac{n^2}{n^2-1}\left(A_k - \frac{2}{n+1} b b^T\right)$$

Update

The Ellipsoid Method

Contents

Interior-Point Methods: an example



Martin
Grötschel

Milestones for Interior Point Methods (IPMs)

- 1984 Projective IPM: Karmarkar – efficient in practice!?
- 1989 $O(n^3L)$ for IPMs: Renegar – best complexity
- 1989 Primal–Dual IPMs: Kojima ... – dominant since then
- 1989 Self-Concordant Barrier: Nesterov–Nemirovskii – extensions to smooth convex optimozation
- 1992 Semi-Definite Optimization (SDO) and Second Order Conic Optimization (SOCO): Alizadeh, Nesterov–Nemirovskii –new applications, approximations, software
- 1998 Robust LO: Ben Tal–Nemirovskii

Martin Grötschel

Contents

Martin Grötschel

Lagrangean Relaxation & Non-differentiable Optimization

- Approach for very large scale and structured LPs
- Methods:
  - subgradient
  - bundle
  - bundle trust region

  or any other nondifferentiable NLP method that looks promising

Martin Grötschel

Lagrangian Relaxation

- Turning an LP into a nonlinear nondifferentiable optimization problem

$$\min c^T x$$
$$Ax = b$$
$$\left.\begin{array}{l} Dx \le d \\ x \ge 0 \end{array}\right\} =: Q$$

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

(14.25) Satz. *Sei Q nicht leer und endlich und $f(\lambda) := \min_{x \in Q}(c^T x + \lambda^T(Ax - b))$, so gilt folgendes: Setzen wir für $\lambda_0 \in \mathbb{R}^m$, $L_0 := \{x_0 \in \mathbb{R}^m \mid f(\lambda_0) = c^T x_0 + \lambda_0^T(Ax_0 - b)\}$, so ist*

$$\partial f(\lambda_0) = \operatorname{conv}\{(Ax_0 - b) \mid x_0 \in L_0\}.$$

Martin Grötschel

Algorithms for nonlinear nondifferential programming

$$x_{i+1} = x_i + s_i d_i$$

d_i = subgradient (instead of gradient)

or element of ε-subdifferential (bundle)

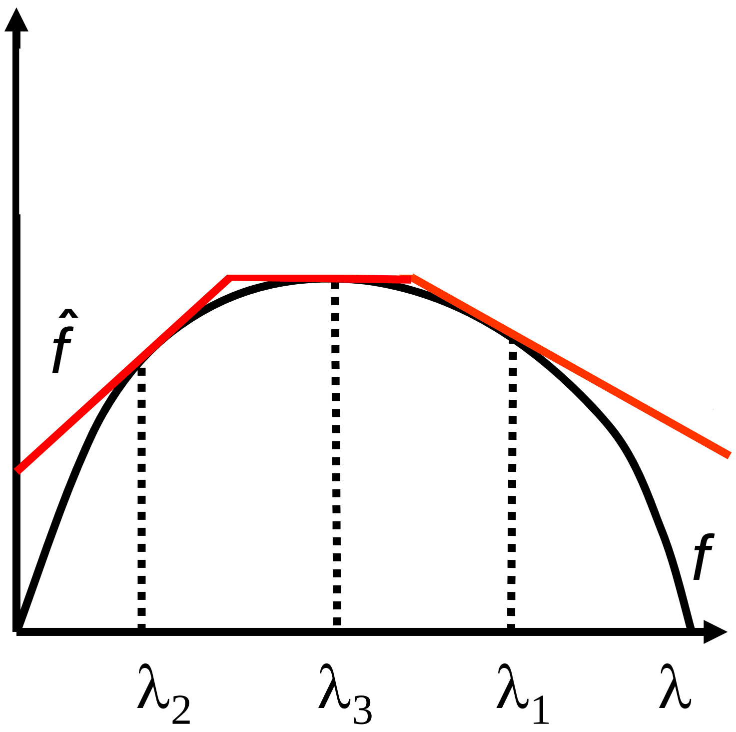
s_i = steplength

Martin Grötschel

Bundle Method

(Kiwiel [1990], Helmberg [2000])

- Max $f(\lambda) := \min_{x \in X} c^{\mathrm{T}} x + \lambda^{\mathrm{T}} (b - Ax)$
  X polyhedral (piecewise linear)



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

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

$$\lambda_{k+1} = \operatorname*{argmax}_{\lambda} \hat{f}_k(\lambda) - \frac{u_k}{2} \left\| \lambda - \hat{\lambda}_k \right\|^2$$

Ralf Borndörfer

Quadratic Subproblem

$$(1) \quad \max \hat{f}_k(\lambda) - \frac{u_k}{2}\left\|\lambda - \hat{\lambda}_k\right\|^2$$

$$\Leftrightarrow \quad (2) \quad \max \quad v - \frac{u_k}{2}\left\|\lambda - \hat{\lambda}^k\right\|^2$$
$$\text{s.t.} \quad v \le \bar{f}_\mu(\lambda), \text{ for all } \mu \in J_k$$

$$\Leftrightarrow \quad (3) \quad \max \quad \sum_{\mu \in J_k} \alpha_\mu \bar{f}_\mu(\hat{\lambda}) - \frac{1}{2u_k}\left\|\sum_{\mu \in J_k} \alpha_\mu (b - Ax_\mu)\right\|^2$$
$$\text{s.t.} \quad \sum_{\mu \in J_k} \alpha_\mu = 1$$
$$0 \le \alpha_\mu \le 1, \quad \text{for all } \mu \in J_k$$

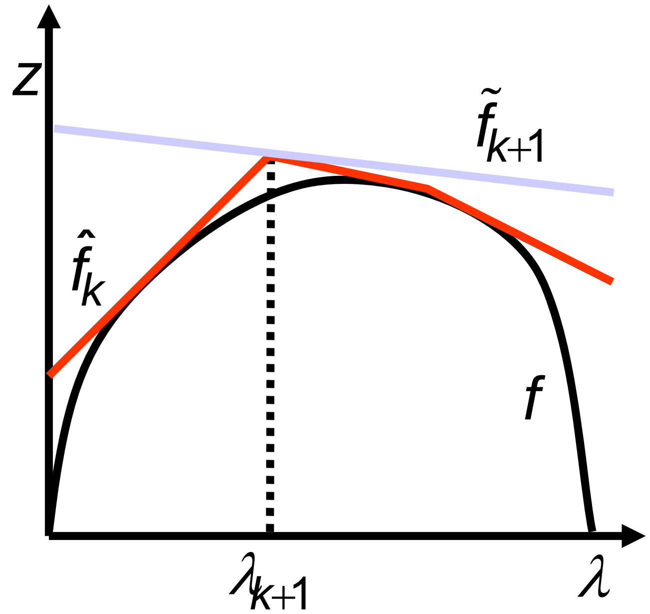
Ralf Borndörfer

Primal Approximation

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

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

$$\tilde{f}_k(\lambda) = c^T \tilde{x}_k + \lambda (b - A\tilde{x}_k)$$



- **Theorem**

$$\left\| b - A\tilde{x}_k \right\| \to 0 \ (k \to \infty)$$

$$\Rightarrow (\tilde{x}_k)_{k \in N} \text{ converges to a point } \bar{x} \in \{x : Ax = b, x \in X\}$$

Ralf Borndörfer

Where Bundle Wins

Ralf Borndörfer Andreas Löbel Steffen Weider

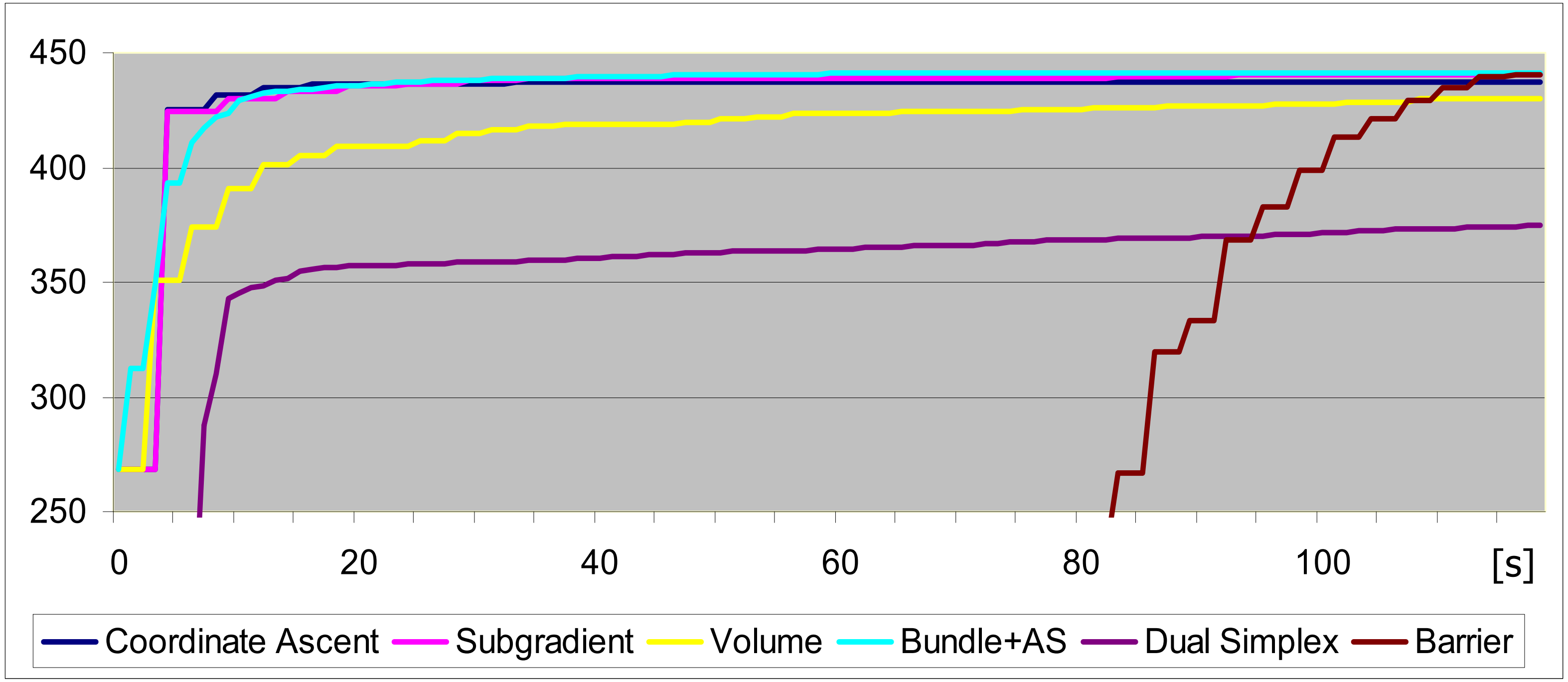
A Bundle Method for Integrated Multi-Depot Vehicle and Duty Scheduling in Public Transit

Coputational Results for a (Duty Scheduling) Set Partitioning Model

Duty Scheduling Problem Ivu41:

- 870 500 col
- 3 570 rows
- 10.5 non-zeroes per col

Coordinate Ascent:	Fast, low quality
Subgradient:	(Theoretical) Convergence
Volume:	Primal approximation
Bundle+AS:	Conv. + primal approx.
Dual Simplex:	Primal+dual optimal
Barrier:	Primal+dual optimal



Ralf Borndörfer

Contents

Martin Grötschel

Contents

Martin Grötschel

Some LP/MIP Solvers

Solver	Version	URL
IBM CPLEX	12.2	www.cplex.com
Gurobi	3.0	www.gurobi.com
FICO XPress-MP	7	www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx
…		
Lindo	6.1	www.lindo.com
Minto	3.1	coral.ie.lehigh.edu/~minto
SCIP	2.0	scip.zib.de
CBC	2.5	projects.coin-or.org/Cbc
Symphony	5.2	projects.coin-or.org/SYMPHONY
glpk	4.43	www.gnu.org/software/glpk/glpk.html
lp_solve	5.5	lpsolve.sourceforge.net
…		

OR / MS Today Surveys

OR/MS Today, June 2009

Linear Programming Survey
Table 3

Product	Platforms Supported											
	PC / Windows		PC / Linux		Unix			Other OS			Microprocessor Support	
	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit	Specify flavor of Unix	32-bit	64-bit	Specify	Shared Memory	Distributed Memory
AIMMS, the modeling system	y	y	y	y							Parallel Solver Sessions (Windows/Linux)	
AMPL	y	y	y	y	y	y	Solaris, Mac OS X, AIX, HP-UX, IRIX					
BendX Stochastic Solver	y	y	y	y	y	y	Sun Solaris, HP-UX, AIX (Unix platforms are (C/C++/Java only)					
C-WHIZ	y	y	y									
CBC	y	y	y	y	y	y	AIX, Solaris			Can be ported to most systems	Linux,Unix, Windows (needs pthreads)	
CLP	y	y	y	y	y	y	AIX, Solaris			Can be ported to most systems		
CoinMP	y	y	y	y			Solaris, Mac OS X					
DATAFORM	y	y	y									
FICO Xpress	y	y	y	y	y	y	Solaris, AIX, HP-UX				All	
flopc++	y	y	y	y	y	y						
Frontier Analyst	y											Task splitting on Windows XP threads

Martin Grötschel

Which LP solvers are used in practice?

Preview summary

- Fourier-Motzkin: hopeless
- Ellipsoid Method: total failure
- primal Simplex Method: good
- dual Simplex Method: better
- Barrier Method: for large LPs frequently better
- For LP relaxations of IPs: dual Simplex Method

Martin Grötschel

http://www.netlib.org/lp/index.html

lp

Click here to see the number of accesses to this library.

```
lib     data
for     a set of test problems in MPS format.

lib     generators
for     programs that generate linear programming test problems

lib     infeas
for     infeasible linear programming test problems
```

Martin
Grötschel

MIPLIB 1992/2010

MIPLIB - Mixed Integer Problem LIBrary

MIPLIB 2010

After its introduction, MIPLIB has become a standard test set used to compare the performance of mixed integer optimizers.

Since the first release in 1992 the MIPLIB has been updated several times. Now again 7 years have past since the last update in 2003. And again improvements in state-of-the-art optimizers, as well as improvements in computing machinery have made several instances too easy to be of further interest.

Last year a group of interested parties including participants from ASU, COIN, FICO, Gurobi, IBM, and MOSEK met at ZIB to discuss the guidelines for the 2010 release of the MIPLIB.

Involved people:

Tobias Achterberg (IBM)
Erling D. Andersen (Mosek)
Oliver Bastert (FICO)
Timo Berthold (ZIB, Matheon)
Robert Bixby (Gurobi)
Gerald Gamrath (ZIB)
Ambros Gleixner (ZIB)
Stefan Heinz (ZIB, Matheon)
Thorsten Koch (ZIB, Matheon)
Alexander Martin (TU Darmstadt)
Hans D. Mittelmann (Arizona State University)
Ted Ralphs (COIN-OR, Lehigh University)
Kati Wolter (ZIB)

We would be happy if you contribute to this library by sending us hard and/or real life instances. If you have any instances you would like to add to MIPLIB, please use the form below to submit it. The current deadline for instances is 10/1/2010!

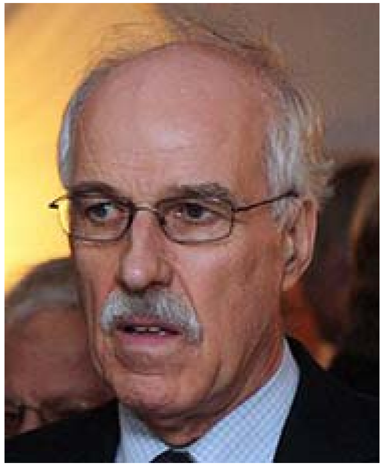
Independent Testing

Benchmarks for Optimization Software

by Hans Mittelmann (mittelmann at asu.edu)

The following are NEOS solvers we have installed.

BNBS, BPMPD, BPMPD-AMPL, Concorde, CONDOR, CSDP, DDSIP, FEASPUMP, FEASPUMP-AMPL, ICOS, NSIPS, PENBMI, PENSDP, QSOPT_EX, SCIP, SCIP-AMPL, SDPA, SDPLR, SDPT3, SeDuMi



http://plato.asu.edu/bench.html

LINEAR PROGRAMMING

- Benchmark of serial LP solvers (10-12-2010)
- Benchmark of parallel LP solvers (10-16-2010)
- Parallel CPLEX, GUROBI, and MOSEK on LP problems (7-18-2010)
- Large Network-LP Benchmark (commercial vs free) (10-16-2010)

MIXED INTEGER LINEAR PROGRAMMING

- MILP Benchmark - serial codes (10-15-2010)
- MILP Benchmark - parallel codes (10-14-2010)
- MILP cases that are difficult for some codes (10-8-2010)
- Feasibility Benchmark - Feaspump,CPLEX,SCIP,GUROBI (10-15-2010)
- Infeasibility Detection for MILP Problems (10-14-2010)

LP survey

Robert E. Bixby, Solving Real-World Linear Programs: A Decade and More of Progress. *Operations Research* 50 (2002)3-15.

Bob on September 27, 2010 at his 65th birthday party

Progress in LP: 1988—2004

(<u>Operations Research</u>, Jan 2002, pp. 3—15, updated in 2004)

- Algorithms (*machine independent*):

  Primal *versus* best of Primal/Dual/Barrier 3,300x
- Machines (workstations →PCs): 1,600x
- NET: Algorithm × Machine 5,300,000x

(2 months/5300000 ~= 1 second)

Courtesy Bob Bixby

Progress in LP: 1988—2004

- Where are we today?
  - The good news
    - "LP is a solved problem in practice"
  - But, …. a word of warning
    - 2% of MIP models are blocked by linear programming
    - Little progress in LP computation since 2004
    - LP could become a serious bottleneck in the future

Courtesy Bob Bixby

The latest computational study: Ed Rothberg (Gurobi)

- Rothberg slides

- LP state of the art - according to Gurobi: as of September 28, 2010 (Bixby's 65th birthday conference in Erlangen, Germany)
- All software producer do computational studies permanently but rarely make them publicly available.

What can we solve today? “strange examples”

Example: Primal > Barrier > Dual

Problem name	: patrick1
Optimal objective	: 28609090
Variables	: 2,666,441 [Boxed: 2,656,781, Nneg: 9,660]
Objective nonzeros	: 684,145
Linear constraints	: 44,886 [Less: 8,173, Equal: 36,713]
Nonzeros	: 7,991,889
RHS nonzeros	: 41,808

Dual Simplex	: 488,900 iterations in 10,009 s (not finished)
Barrier+crossover	: 349 iterations in 3,111 s
Primal Simplex	: 3,268,455 (895,004) iterations in 1,900 s

Martin Grötschel

What can we solve today? "strange examples"

Example: Barrier > Primal > Dual

Problem name	: aflow_2000_50
Optimal objective	: 4720.3225806
Variables	: 3,996,000 [Boxed: 1,998,000, Nneg: 1,998,000]
Objective nonzeros	: 1,958,437
Linear constraints	: 2,001,998 [Less: 1,998,000, Equal: 3,998]
Nonzeros	: 9,988,972
RHS nonzeros	: 3,998

Dual Simplex	: 1,049,300 iterations in 10,054 sec (not finished)
Primal Simplex	: 2,321,540 (28277) iterations in 6,752 sec
Barrier + crossover	: 40 iterations in 1,704 sec (total 1,938 sec)
8 threads	: 430.03 sec

Martin Grötschel

What can we solve today? "strange examples"

Example: Primal > Dual > Barrier

Problem name	: ts.log-bundle-060831-162253
Optimal objective	: 5.69997.52369
Variables	: 218,776 [Boxed: 218,776]
Objective nonzeros	: 124,060
Linear constraints	: 1,102,735 [Less: 970,339, Greater: 11,590, Equal: 120,806]
Nonzeros	: 2,554,196
RHS nonzeros	: 981,241

Presolve generated explicit dual

Dual Simplex	: 132854 in 163 sec
Primal Simplex	: 96397 (0) in 31 sec
Barrier	: 53 iterations in 10069 sec (not finished)

Martin Grötschel

ZIB Instances

	Variables	Constraints	Non-zeros	Description
1	12,471,400	5,887,041	49,877,768	Group Channel Routing on a 3D Grid Graph (Chip-Bus-Routing)
2	37,709,944	9,049,868	146,280,582	Group Channel Routing on a 3D Grid Graph (different model, infeasible)
3	29,128,799	19,731,970	104,422,573	Steiner-Tree-Packing on a 3D Grid Graph
4	37,423	7,433,543	69,004,977	Integrated WLAN Transmitter Selection and Channel Assignment
5	9,253,265	9,808	349,424,637	Duty Scheduling with base constraints

LP can still be difficult

- **We were not able to compute a feasible basis for zib03 so far.**
- After 10 h we still do not even have a primal feasible solution. Furthermore, experiments with smaller instances suggest the model is very unfavorable for the simplex method, especially regarding warm starts. Unfortunately, it is an IP.

Algorithm	Time [h]	Result	Memory [GB]	Resident [GB]
Primal Simplex	>300	Infeasibility 2189	24	18
Dual Simplex	>300	Lower bound 8335	24	18
Bundle	13	Lower bound 5951	55	18
Interior Point	103 (32 threads)	Optimal 1.2228.148	256	175
Crossover	>300	unfinished		

Summary

You should be surprised
if a linear program could not be solved

Martin
Grötschel

Contents

Martin Grötschel

Advertisement: http://zibopt.zib.de/

ZIB Optimization Suite

Konrad-Zuse-Zentrum für Informationstechnik Berlin
Division Scientific Computing
Department Optimization

The ZIB Optimization Suite is a tool for generating and solving mixed integer programs. It consists of the following parts

ZIMPL	a mixed integer programming modeling language
SoPlex	a linear programming solver
SCIP	a mixed integer programming solver and constraint programming framework.

The user can easily generate linear programs and mixed integer programs with the modeling language ZIMPL. The resulting model can directly be loaded into SCIP and solved. In the solution process SCIP may use SoPlex as underlying LP solver.

Since all three tools are available in source code and free for academic use, they are an ideal tool for academic research purposes and for teaching integer programming.

See ZIB licences for more information.

SoPlex Sequential object-oriented simplex

SoPlex is an implementation of the revised simplex algorithm. It features primal and dual solving routines for linear programs and is implemented as a C++ class library that can be used with other programs.

Roland Wunderling,
Paralleler und Objektorientierter Simplex-Algorithmus,
Dissertation, TU Berlin,1997

Zimpl

- Zimpl is a little language to translate the mathematical model of a problem into a linear or (mixed-) integer mathematical program expressed in .lp or .mps file format which can be read and (hopefully) solved by a LP or MIP solver.

- Thorsten Koch, *Rapid Mathematical Programming*, Dissertation, TU Berlin 2004 (awarded with the Dissertation Prize 2005 of the Gesellschaft für Operations Research)

SCIP http://scip.zib.de/

Tobias Achterberg, Tobias, *Constraint Integer Programming*, Dissertation, TU Berlin, 2007

- Dissertation Prize 2008 of the Gesellschaft für Operations Research (GOR)
- George B. Dantzig Dissertation Award 2008 of the Institute of Operations Research and the Management Sciences (INFORMS), 2nd prize)
- Beale-Orchard-Hays Prize 2009 of the Mathematical Optimization Society for the paper: Tobias Achterberg, "SCIP: Solving constraint integer programs", Mathematical Programming Computation, 1 (2009), pp. 1-41.

Contents

Martin Grötschel

Mathematics of Infrastructure Planning (ADM III)
Part II: Solving IP/MIP Problems

TU Berlin
Summer Semester 2012
First Lecture on April 12, 2012

Ralf Borndörer & Martin Grötschel

ZIB, TU, and MATHEON, Berlin

Martin Grötschel
- Institut für Mathematik, Technische Universität Berlin (TUB)
- DFG-Forschungszentrum "Mathematik für Schlüsseltechnologien" (MATHEON)
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

groetschel@zib.de http://www.zib.de/groetschel

Contents

Martin Grötschel

Contents

Martin Grötschel

typical optimization problems

$$\max\ f(x)\ \ or\ \ \min\ f(x)$$
$$g_i(x) = 0, \quad i = 1, 2, \ldots, k$$
$$h_j(x) \leq 0, \quad j = 1, 2, \ldots, m$$
$$x \in \mathbb{R}^n \text{ (and } x \in S)$$

„general“ (nonlinear) program NLP

$$\min c^T x$$
$$Ax = a$$
$$Bx \leq b$$
$$x \geq 0$$
$$(x \in \mathbb{R}^n)$$
$$(x \in \mathbb{k}^n)$$

linear program LP

$$\min c^T x$$
$$Ax = a$$
$$Bx \leq b$$
$$x \geq 0$$
$$\text{some } x_j \in \mathbb{Z}$$
$$(x \in \{0,1\}^n)$$

(linear) 0/1-mixed-integer program IP, MIP

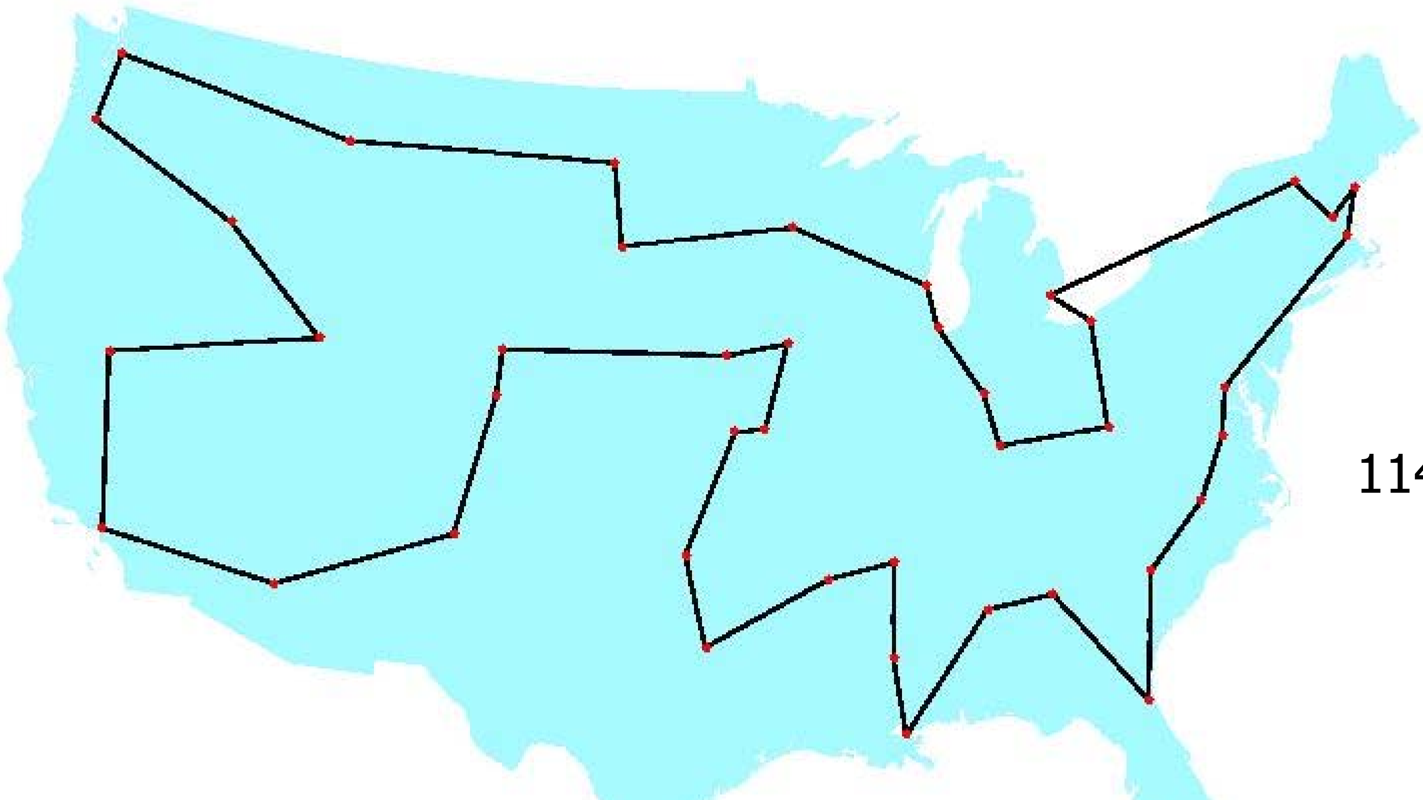
program = optimization problem

Martin Grötschel

Contents

Martin Grötschel

1954, the Beginning of IP
G. Dantzig, D.R. Fulkerson, S. Johnson



USA
49 cities
1146 variables

1954

George Dantzig's contributions to integer programming
Martin Grötschel and George L. Nemhauser
Discrete Optimization
Volume 5, Issue 2, May 2008, Pages 168-173

Martin Grötschel

A Milestone Paper

H.W. Kuhn, The Hungarian Method for the assignment problem,
Naval Research Logistic Quarterly, 2 (1955) 83-97.

In 2006, it was discovered that Carl Gustav Jacobi had solved the assignment problem. The paper (on diferential equations) was published posthumously in 1890 in Latin.

EGRES Technical Report No. 2004-14 1

On Kuhn's Hungarian Method – A tribute from Hungary

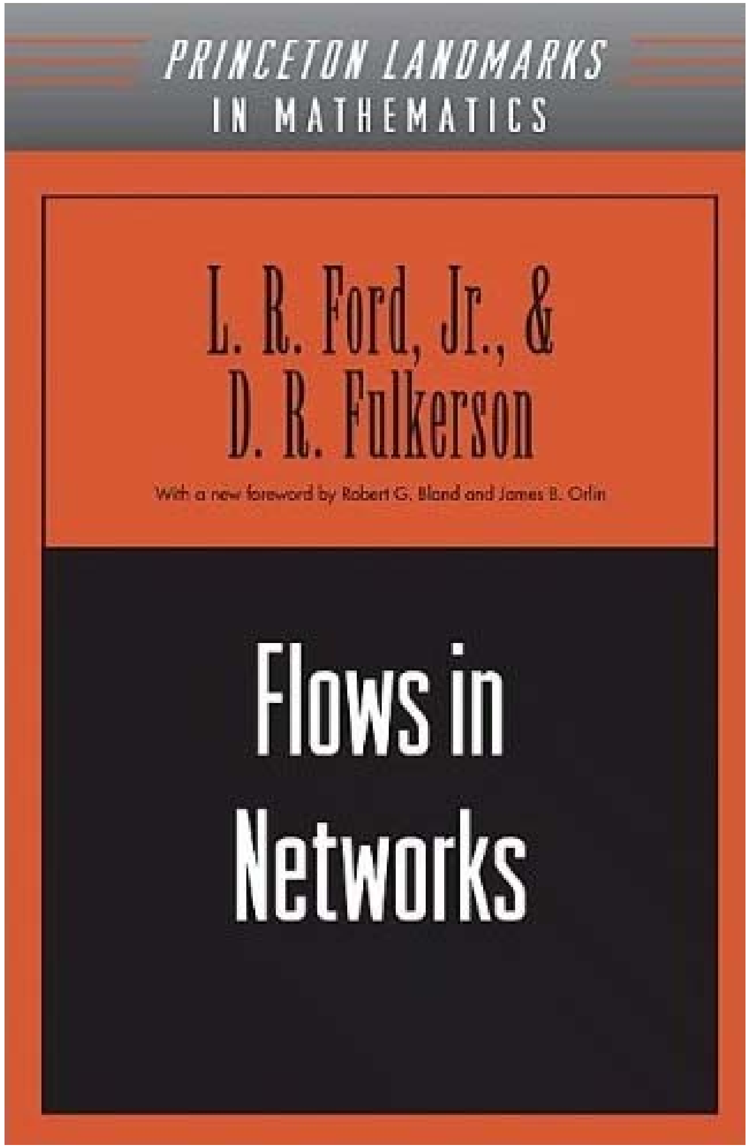
András Frank*

Harold W. Kuhn, in his celebrated paper entitled *The Hungarian Method for the assignment problem,* [Naval Research Logistic Quarterly, 2 (1955), pp. 83-97] described an algorithm for constructing a maximum weight perfect matching in a bipartite graph. In his delightful reminescences [18], Kuhn explained how the works (from 1931) of two Hungarian mathematicians, D. Kőnig and E. Egerváry, had contributed to the invention of his algorithm, the reason why he named it the Hungarian Method. (For citations from Kuhn's account as well as for other invaluable historical notes on the subject, see A. Schrijver's monumental book [20].)

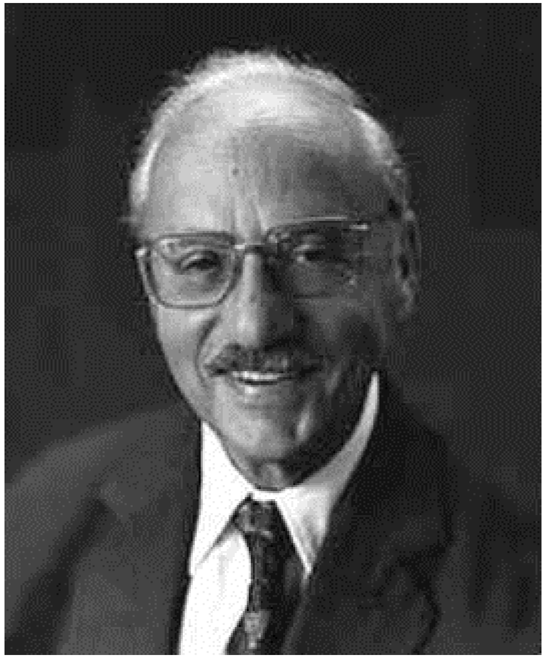
In this note I wish to pay tribute to Professor H.W. Kuhn by exhibiting the exact ralationship between his Hungarian Method and the achievements of Kőnig and Egerváry, and by outlining the fundamental influence of his algorithm on Combinatorial Optimization where it became the prototype of a great number of algorithms in areas such as network flows, matroids, and matching theory. And finally, as a Hungarian, I would also like to illustrate that not only did Kuhn make use of ideas of Hungarian mathematicians, but his extremely elegant method has had a great impact on the work of a next generation of Hungarian researchers.

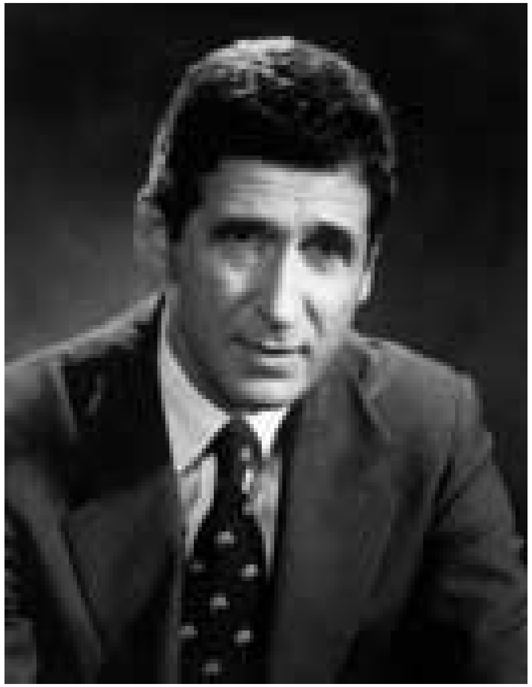
Martin Grötschel

A Milestone Book in IP



George Dantzig and Ralph Gomory





„founding fathers"

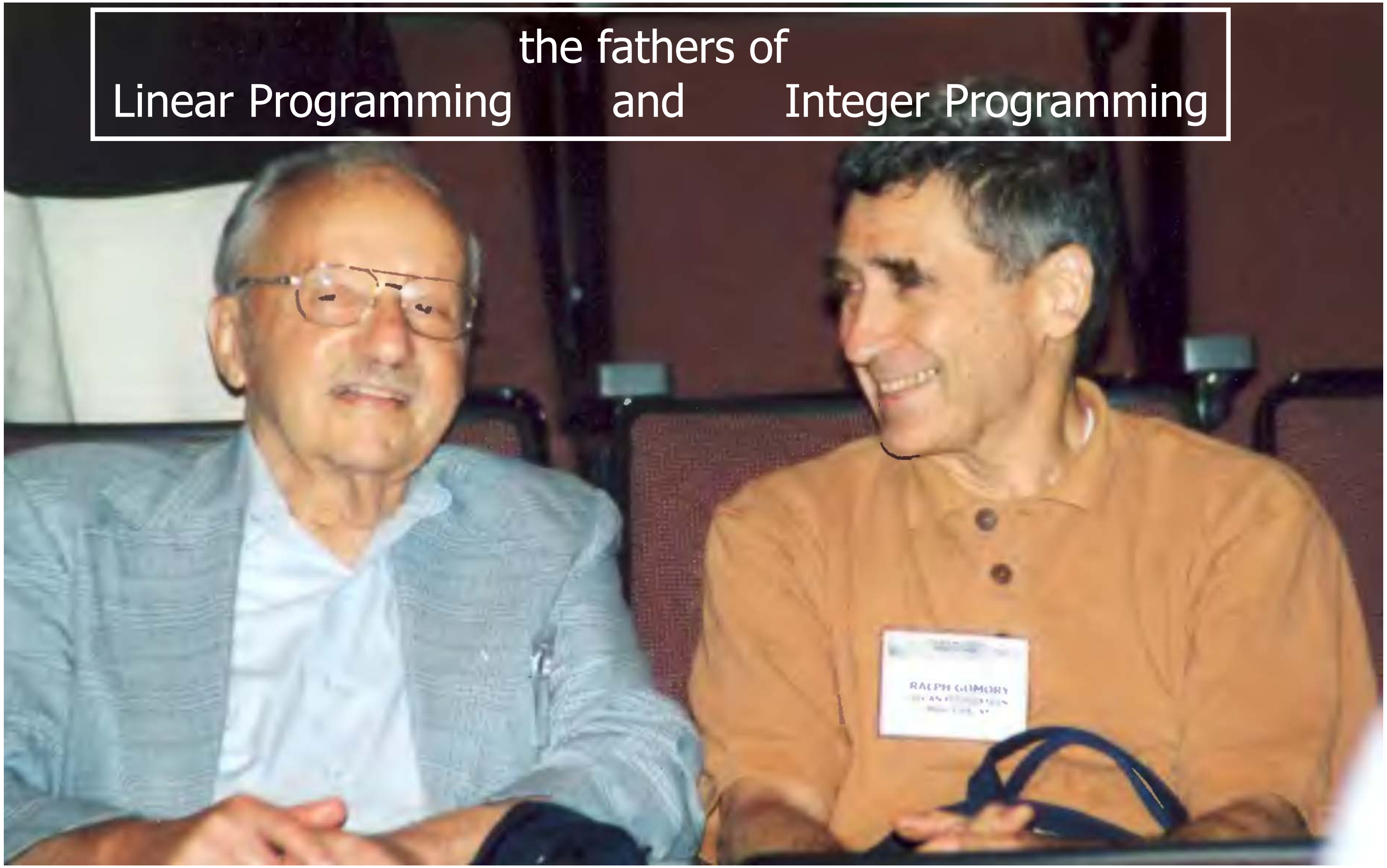
~1950
linear programming

~1960
integer programming

Martin Grötschel

George Dantzig and Ralph Gomory

ISMP Atlanta 2000



Martin Grötschel

Dantzig and Bixby

George Dantzig and
Bob Bixby (founder of CPLEX)

at the International Symposium on Mathematical Programming,

Atlanta, August 2000

Martin Grötschel

Contents

Martin Grötschel

Optimizers' dream: Duality theorems

- Max-Flow Min-Cut Theorem

  The value of a maximal (s,t)-flow in a capacitated network is equal to the minimal capacity of an (s,t)-cut.

- The Duality Theorem of Linear Programming

$$\begin{aligned} \max\ & c^T x \\ & Ax \le b \\ & x \ge 0 \end{aligned} \quad = \quad \begin{aligned} \min\ & y^T b \\ & y^T A \ge c^T \\ & y \ge 0 \end{aligned}$$

Martin Grötschel

Optimizers' dream: Duality theorems for integer programming

- The Max-Flow Min-Cut Theorem
  does not hold if several source-sink relations are given
  (multicommodity flow).
- The Duality Theorem of Linear Programming
  does not hold if integrality conditions are added

Important technique:

Use polyhedral theory to obtain "= ".

$$\max c^T x \quad \leq \quad \min y^T b$$

$$Ax \leq b \qquad\qquad y^T A \geq c^T$$

$$x \geq 0 \qquad << \qquad y \geq 0$$

$$x \in \mathbb{Z}^n \qquad\qquad y \in \mathbb{Z}^m$$

Martin Grötschel

IP Solvability

Theorem

Integer, 0/1, and mixed integer programming are NP-hard.

Consequences

- Investigation of special cases
- Exact problem specific special purpose algorithms
- Design of special purpose heuristics

Contents

Martin Grötschel

Public Transport Projects (ZIB and Matheon)

Busses (Berlin and elsewhere)

- Telebus (Transportation of disabled persons)
- Bus Circulation
- Bus driver Scheduling
- Integrated Vehicle and Driver Scheduling
- Timetable Exchange

Subways and Light Railways

- Subway Time Tabling
- Vehicle Scheduling

Infrastructure Planning

- Line Planing
- Network Planning (Potsdam)
- Fare Planing

Airlines

- Airline Crew Scheduling
- Tail Assignment: Robustness

Railways

- Railway Track Allocation
- ICE Circulation

Spin-Offs : LBW, Intranetz

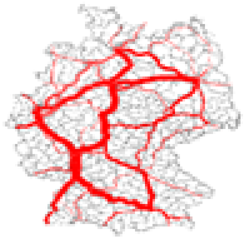
Current traffic / transport projects at ZIB

MATHEON-B22: Rolling Stock Roster Planning

Umlaufplanung im Schienenverkehr

» zur Projektdarstellung

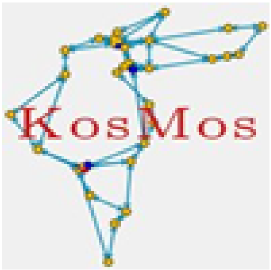
MATHEON-B15: Service Design in Public Transit



Angebotsplanung im Öffentlichen Nahverkehr

» zur Projektdarstellung

KosMos



Optimale Zugführung im Schienengüterverkehr

» zur Projektdarstellung

TollControlOpt

Optimierung des Mautkontrolldienstes

» zur Projektdarstellung

Vehicle Rotation Planning

DB BAHN

Fahrzeugumlaufplanung für die DB Fernverkehr AG

» zur Projektdarstellung

VPP

Vehicle Positioning Problem

» zur Projektdarstellung

Public Transport

Savings in Berlin public transport

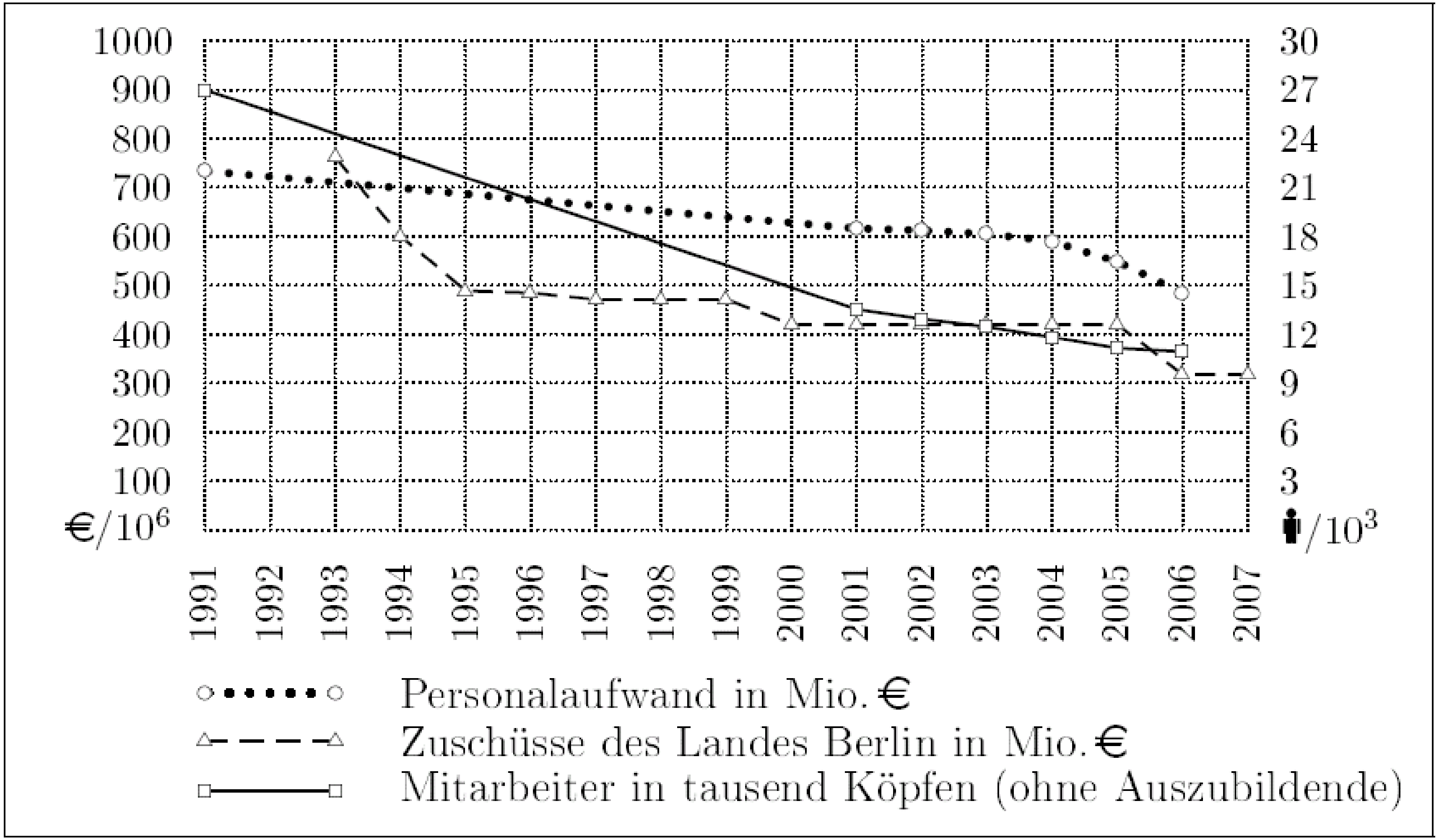
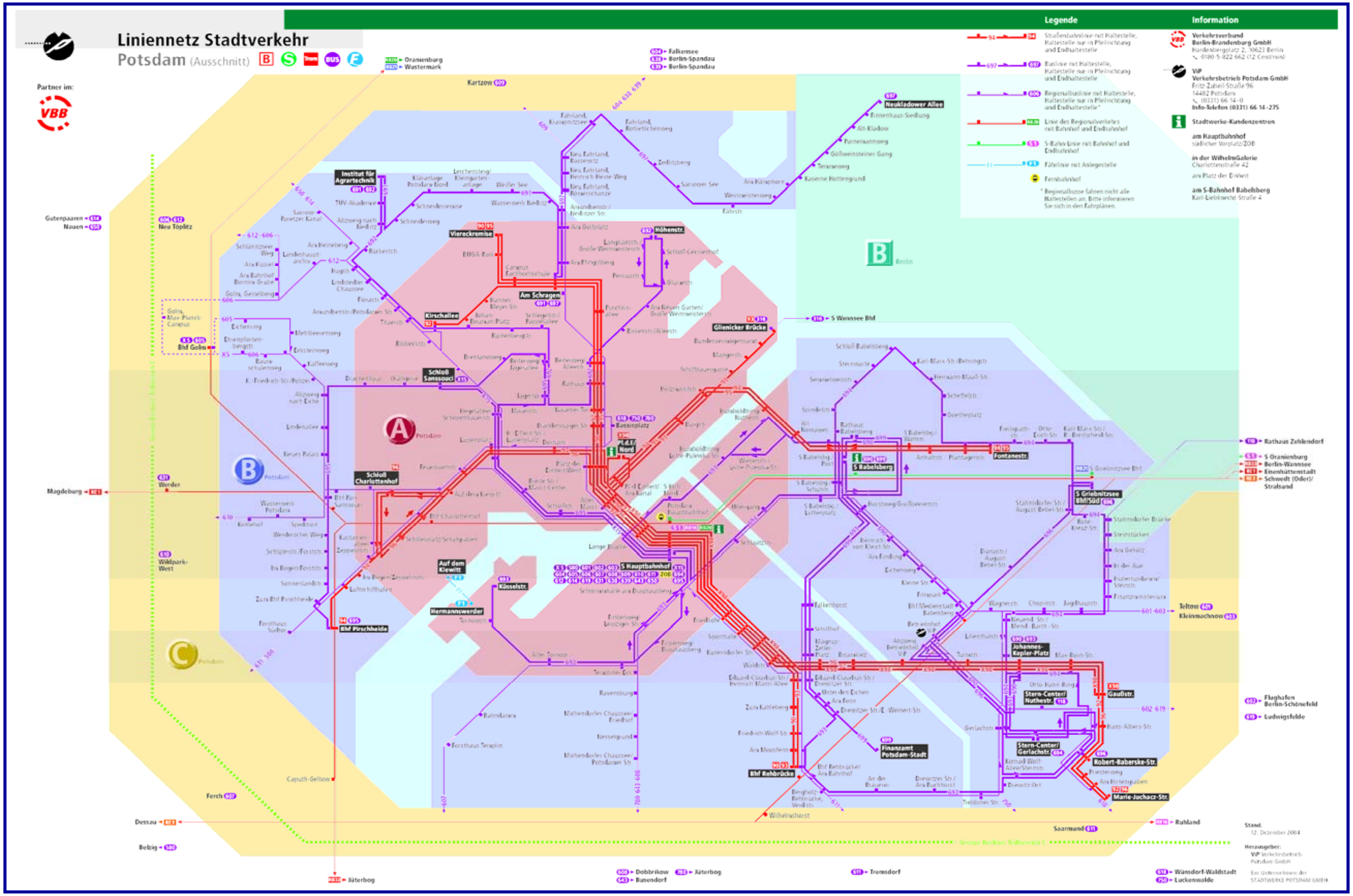


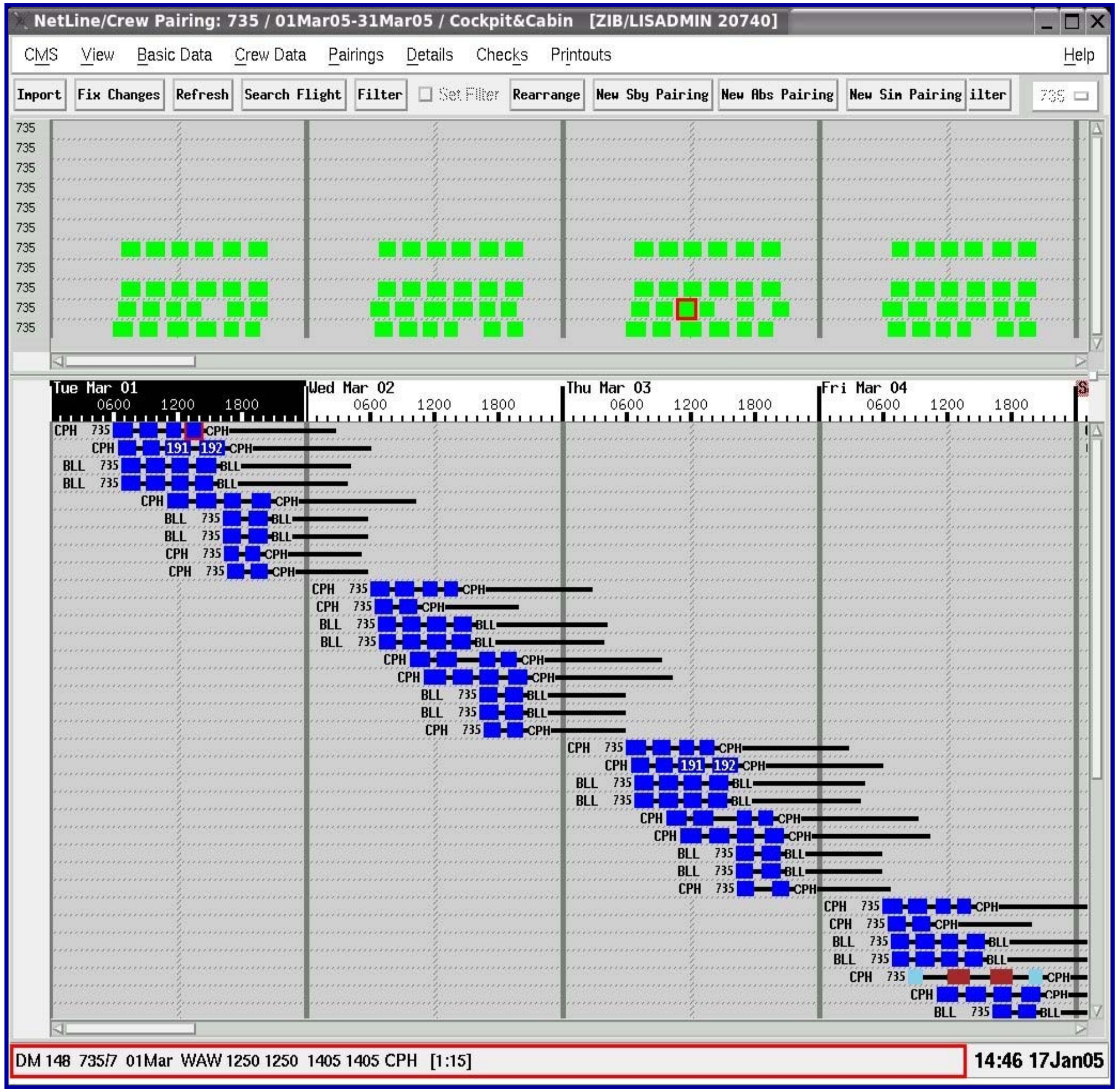
Abb. 3: Die Berliner Verkehrsbetriebe in Zahlen; Quelle: [36].

Network, Line and Fare Planning (Potsdam)



CS-OPT in NetLine/Crew

Lufthansa Systems



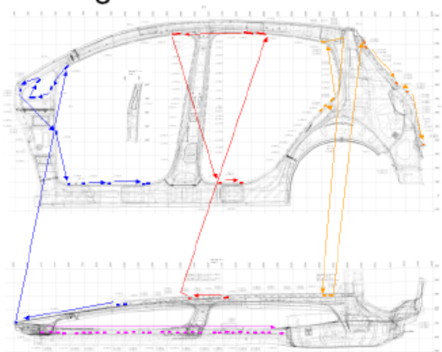
Martin Grötschel

Production und Logistics

- Optimization of a container terminal in Botany Bay, Sydney (TUB, UNSW, Uni Melbourne, Patrick Corp.)

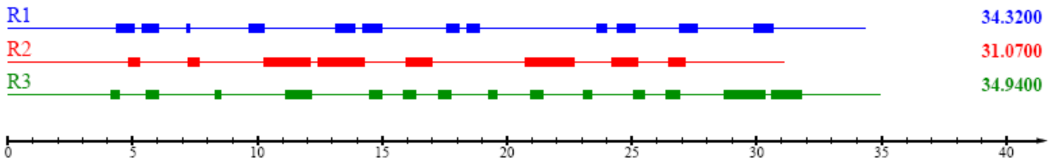
- Open-pit mine scheduling (BHP Billiton)
- Laser welding (Volkswagen AG)

Planung einer Schweißstation:



Vorgeben:

1. Menge von Schweißnähten
   - Positionen
   - Schweißzeiten
2. Umschaltzeit einer Laserquelle
3. Taktzeit



Martin Grötschel

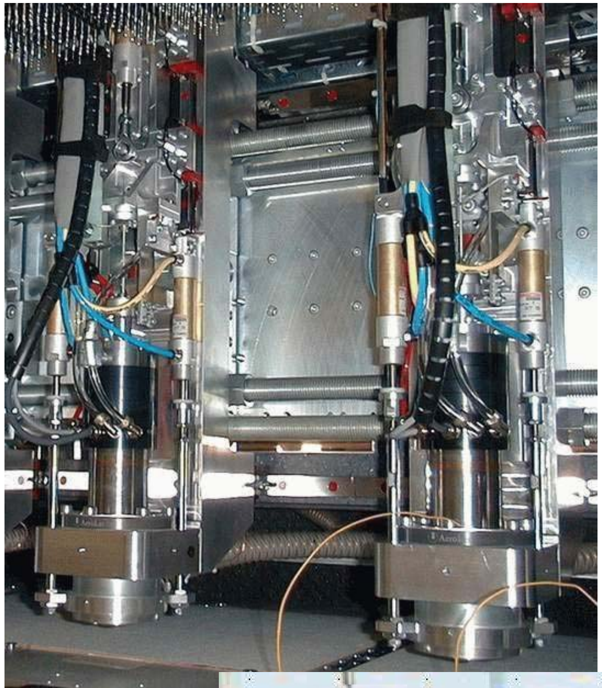
Herlitz at Falkensee (Berlin)

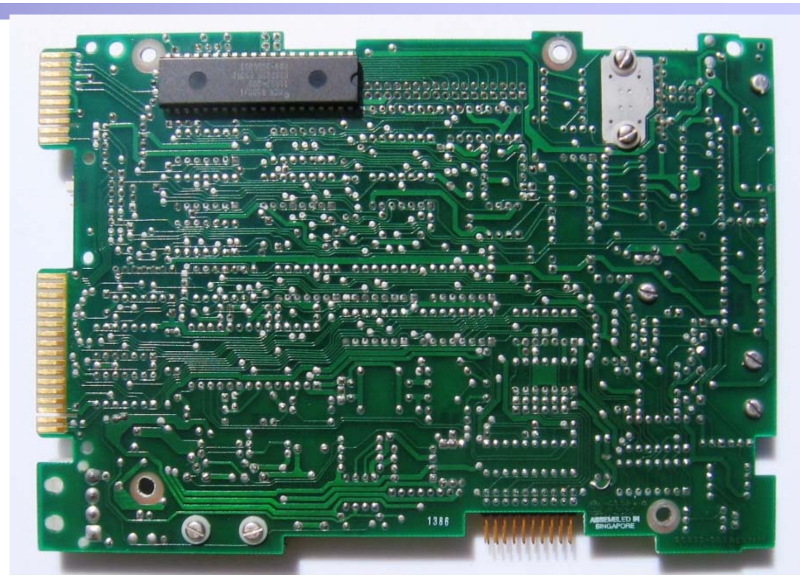
Martin Grötschel

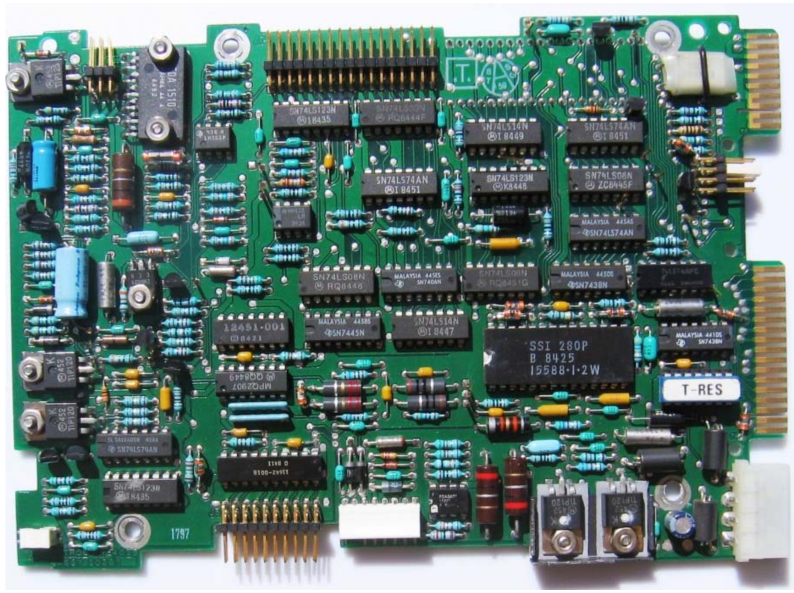
Optimization and control of transport devices
(such as elevators, stacker cranes) in factories

Herlitz, Falkensee

Printed Circuit Board Drilling and Assembly Machines







Martin Grötschel

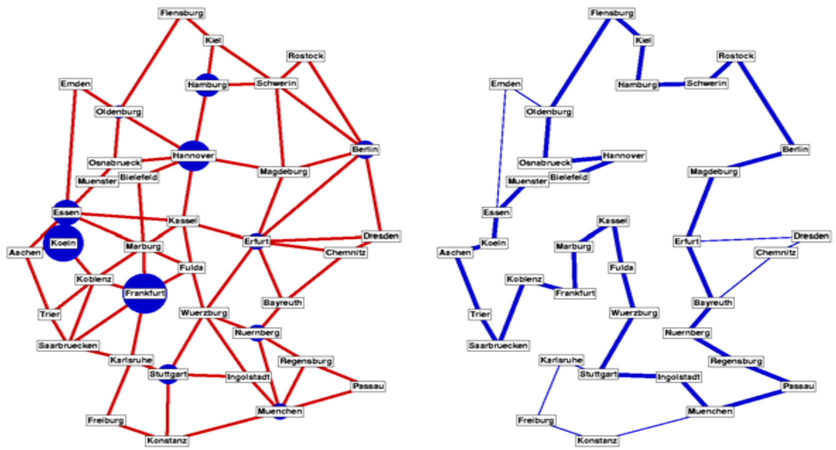
Telecommunication topics: Hardware and logistics

- Designing mobile phones
  - Task partitioning
  - Chip design (VLSI)
  - Component design
- Producing Mobile Phones
  - Production facility layout
  - Control of CNC machines
  - Control of robots
  - Cutting and welding
  - Printed Circuit Boards
    - Via minimization
    - Component Placement
    - Mounting Devices
    - Routing
  - Lot sizing
  - Scheduling
  - Logistics
- Marketing and Distributing Mobiles

Martin Grötschel

The Basic Question

Telecommunication network design:



Input:

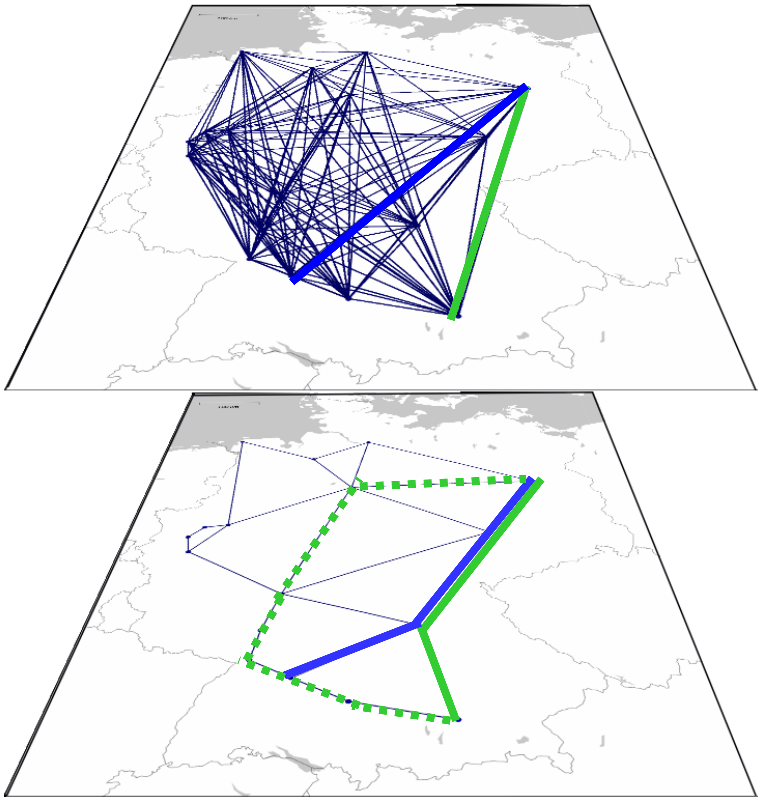
- ▷ potential network
- ▷ demands
- ▷ cost values
- ▷ various additional constraints

Output: cost-minimal

- ▷ subset of nodes and links
- ▷ discrete capacities
- ▷ survivable routing
- ▷ satisfying all constraints

Multi-layer telecommunication network design:

Logical layer



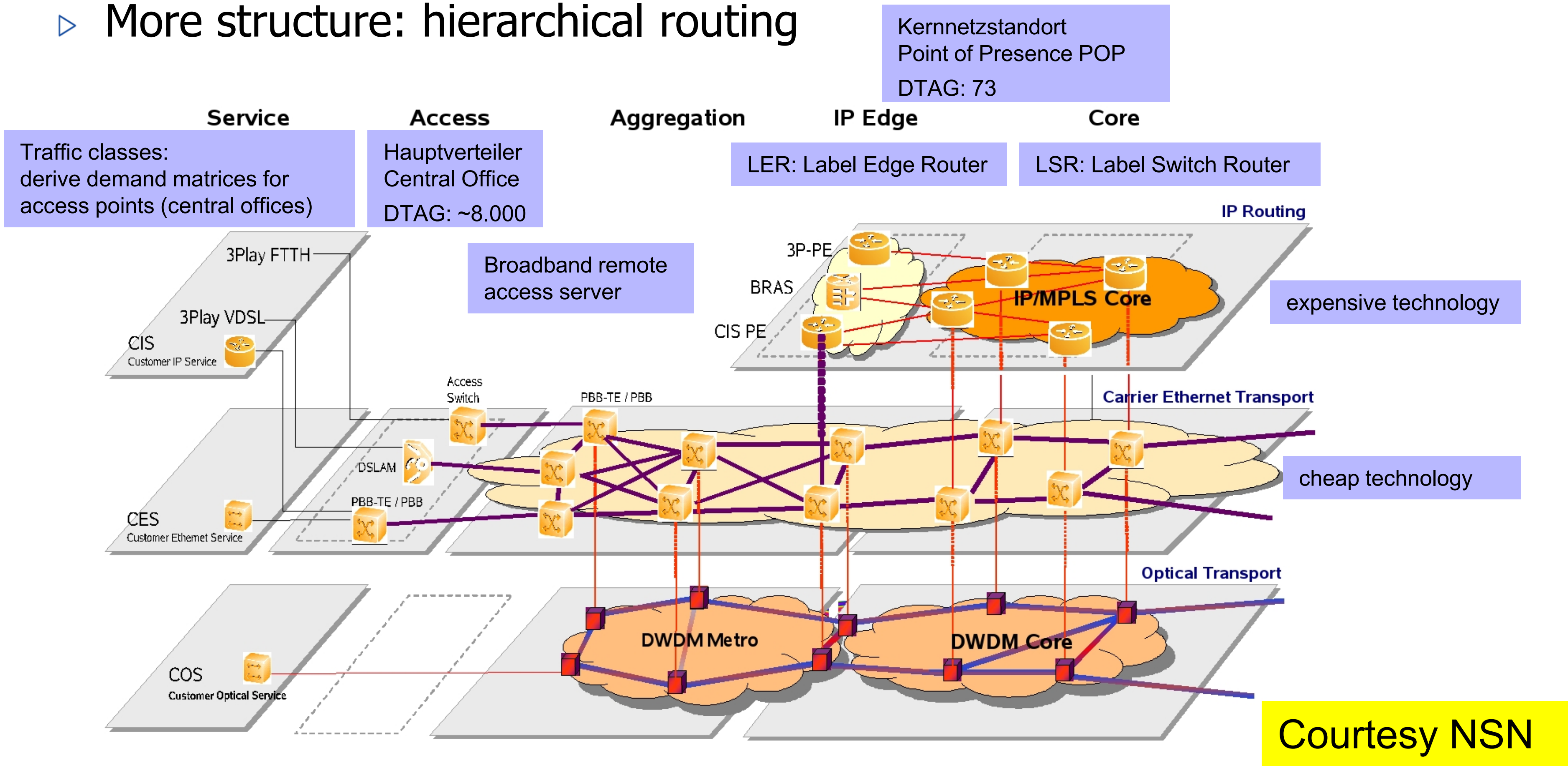
Physical layer

- **Challenges:**
  - "physical" and "logical" layer
  - logical links ≈ physical paths
  - complete "logical graph"
  - parallel "logical links"
  - multiple link failures
- Project goal, MATHEON B3 (Wessäly, Orlowski):
  - Integrated planning of several network layers

Multi-layer multi-level Planning

Goal: Integration of multi-layer backbone and regional networks

- Future networks: IP/Ethernet layer over shared optical fiber layer
- Huge networks (900 nodes), combine different services/technologies
- More structure: hierarchical routing



Integrated MIP model

- Integrated mixed-integer programming model: first model that is
  - Realistic: survivability, several bandwidths, node hardware
  - Suitable for computations

$$\min \sum_{i\in V}\sum_{m\in M_i} c_i^m x_i^m + \sum_{e\in E} c_e x_e + \sum_{\ell\in L}\sum_{m\in M_\ell} c_\ell^m y_\ell^m$$

$$\sum_{j\in V}\sum_{\ell\in L_{ij}} (f_{\ell,ij}^k - f_{\ell,ji}^k) = v_i^k \qquad i\in V,\ k\in K$$

$$\sum_{m\in M_\ell} C_\ell^m y_\ell^m - \sum_{k\in K} (f_{\ell,ij}^k + f_{\ell,ji}^k) \geq 0 \qquad \ell\in L$$

$$B_e x_e - \sum_{\ell\in L_e}\sum_{m\in M_\ell} y_\ell^m \geq 0 \qquad e\in E$$

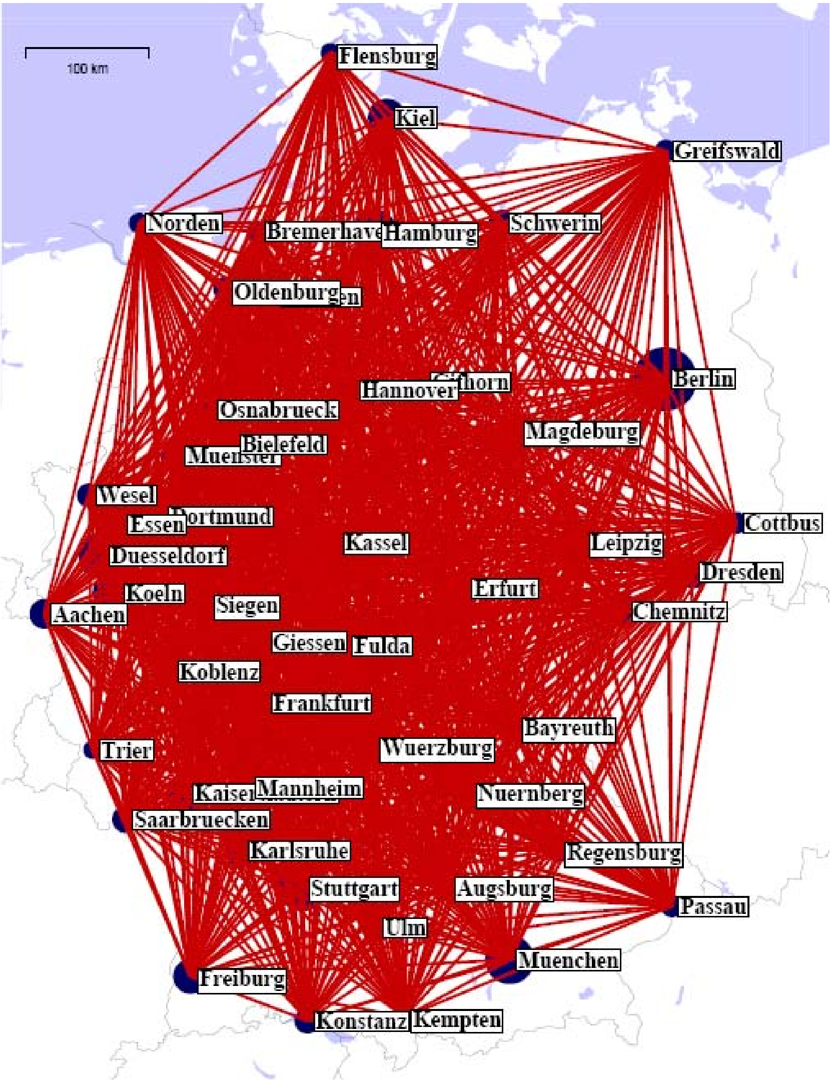
$$\sum_{m\in M_i} x_i^m \leq 1 \qquad i\in V$$

$$2\sum_{m\in M_i} C_i^m x_i^m - \sum_{\ell\in L_i}\sum_{m\in M_\ell} C_\ell^m y_\ell^m \geq v_i \qquad i\in V$$

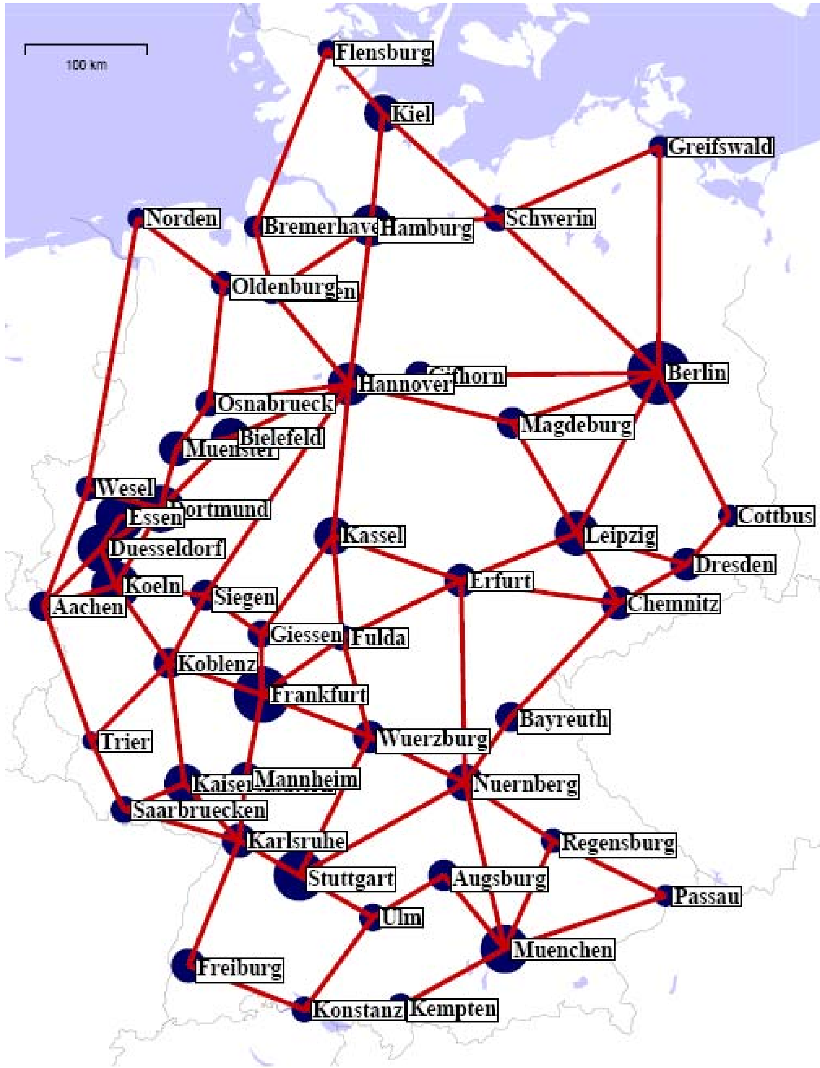
Too complex for standard tools, special algorithms needed

Telecommunication network design
MATHEON B4

Logical connections

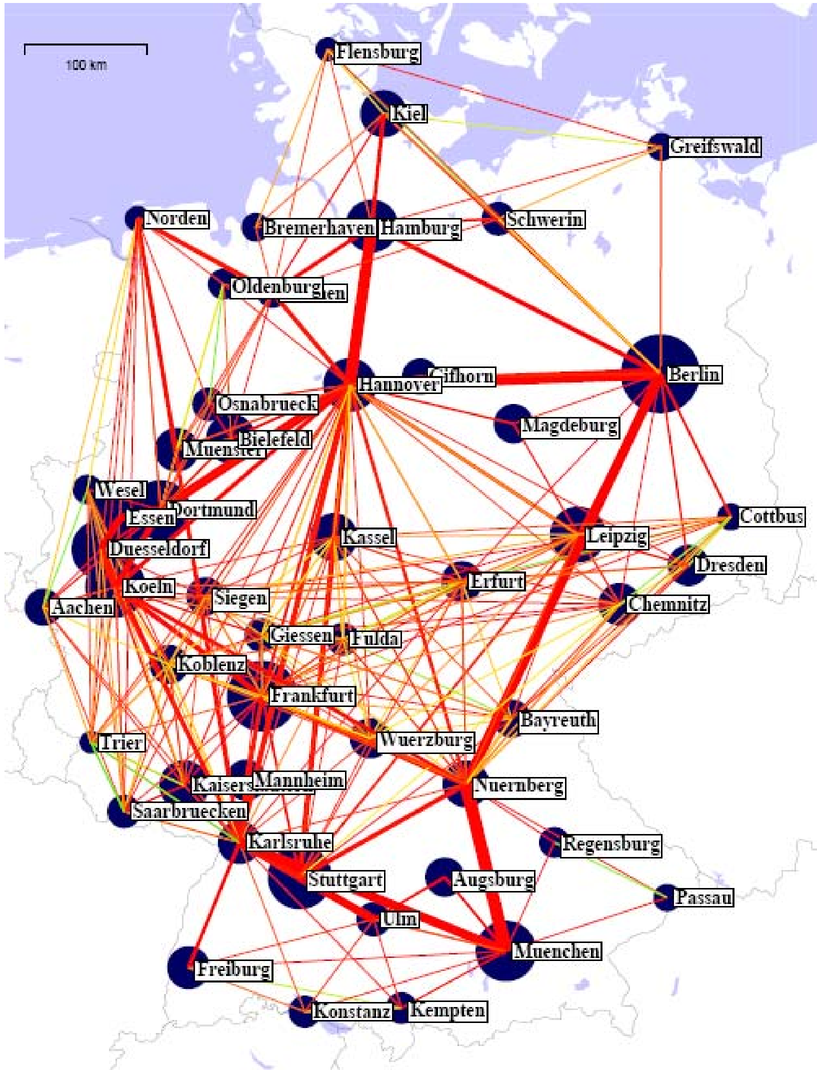


Physical connections

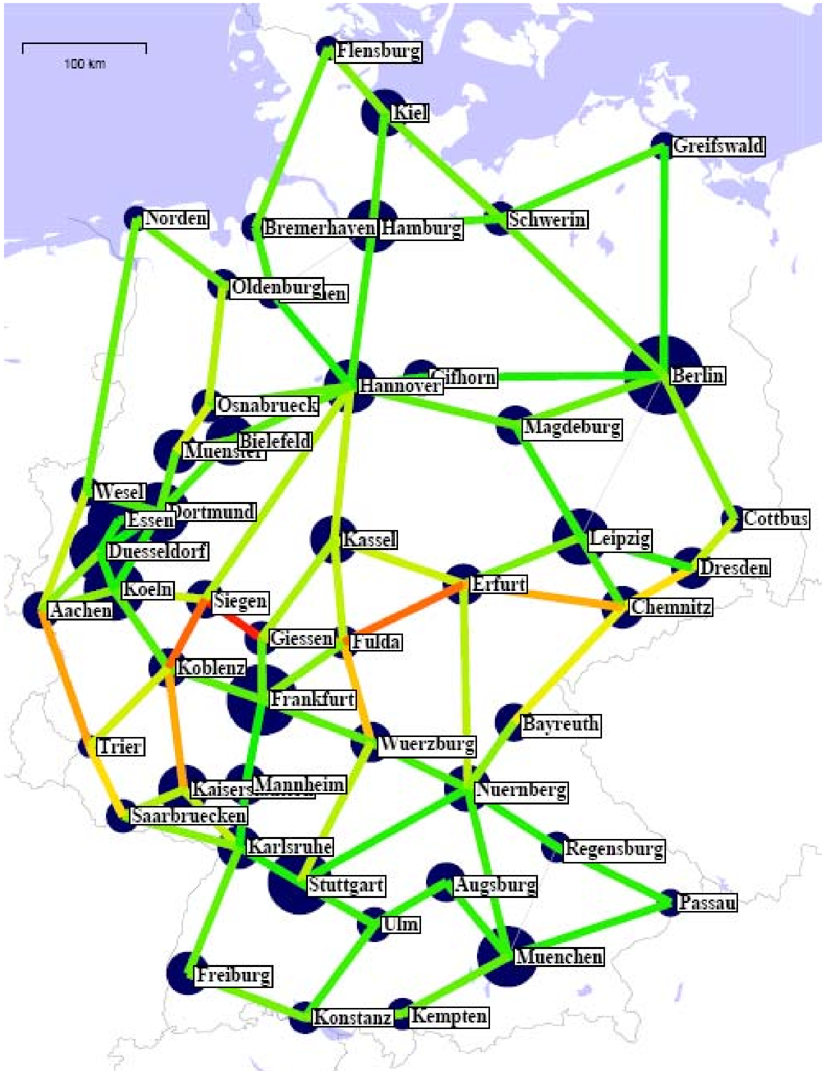


Network design
MATHEON B3

Logical connections: solution



Physical connections: solution



A success story: Deutsches Wissenschaftsnetz

- **Evolution of the Wissenschaftsnetz**
- 1990 S-WiN (Schmalband-WiN)
- 1996 B-WiN (Breitband-WiN)
- 2000 G-WiN (Gigabit-WiN)
- 2006 X-WiN

Optimization of the B-WiN, G-WiN, and X-WiN was carried out by Andreas Bley (with support of DFN, in particular, Marcus Pattloch)

- Example publication:

  A. Bley, M. Pattloch, *Modellierung und Optimierung der X-WiN Plattform,* Journal DFN-Mitteilungen, 67 (2005) 4-7

DFN-Verein X-WiN: German Science Network

- Project carried out by Andreas Bley (ZIB) in cooperation with DFN-Verein (Marcus Pattloch)

  PhD Thesis Andreas Bley:
  Routing and Capacity Optimization for IP Networks
  at TU Berlin
- GOR Dissertationspreis 2007 and
- INFORMS Doctoral Dissertation Award for Operations Research in Telecommunications 2008

Martin Grötschel

Data and a glimpse at the model

Gegebene Parameter

V	Menge der V-Standorte.
A	Menge der möglichen A-Standorte. Sie werden entweder A-Standort oder Anwenderstandort.
N	Menge der Anwenderstandorte
L	Menge aller möglichen Verbindungen zwischen Anwenderstandort und V- oder A-Standort. Für jede Anbindung wird jeweils nur die billigste Verbindung berücksichtigt, deren Kapazität mindestens so groß ist wie die Anschlussbandbreite des Anwenders.
P	Menge aller möglichen Ketten zur Anbindung von A-Standorten an die V-Standorte. Jede Kette hat die Form $(v_1, a_1, \ldots, a_m, v_2)$, d.h. sie bindet die A-Standorte $a_1, \ldots, a_m$ ausfallsicher an die beiden V-Standorte v_1 und v_2 an. Für jede Kombination von Kapazitäten auf den einzelnen Verbindungen gibt es eine eigene Kette p.
k_a^A	Kosten für das Einrichten des A-Standortes $a \in A$.
k_{ij}^L	Kosten der (billigsten) Zugangsleitung $ij \in L$ von Anwenderstandort i zu A- oder V-Standort j.
k_p^P	Kosten der Kette $p = (v_1, a_1, \ldots, a_m, v_2) \in P$ zur Anbindung der A-Standorte $a_1, \ldots, a_m$ an die V-Standorte v_1, v_2. Die Kosten einer Kette sind die Summe aller Einzelverbindungskosten.
c_p	Kapazität der Kette p. Sie ist die kleinste Kapazität aller Einzelverbindungen.

Entscheidungsvariablen

y_a	1 genau dann, wenn a zum A-Standort wird, 0 sonst.
x_{ij}	1 genau dann, wenn i ein Anwenderstandort ist oder wird und i an den A- oder V-Standort j angebunden wird, 0 sonst.
z_p	1 genau dann, wenn $a_1, \ldots, a_m$ zu A-Standorten werden und diese über die Kette $p = (v_1, a_1, \ldots, a_m, v_2)$ an die V-Standorte v_1, v_2 angebunden werden, 0 sonst.

Zielfunktion

Ziel ist die Minimierung der Gesamtkosten für das Einrichten der gewählten A-Standorte, für die Ketten zur Anbindung dieser A-Standorte an das V-Netz, sowie für die Zugangsleitungen zu den übrigen Anwenderstandorten:

$$\min \sum_{a \in A} k_a^A y_a + \sum_{p \in P} k_p^P z_p + \sum_{ij \in L} k_{ij}^L x_{ij}$$

Nebenbedingungen

Jeder Anwenderstandort wird an genau einen A- oder V-Standort angebunden:

$$\sum_{ij \in L} x_{ij} = 1 \quad \text{für alle } i \in N.$$

Wird ein möglicher A-Standort nicht eingerichtet, so wird dieser Standort als Anwenderstandort an einen anderen A- oder V-Standort angebunden:

$$\sum_{aj \in L} x_{aj} = 1 - y_a \quad \text{für alle } a \in A.$$

Ein Anwenderstandort kann nur dann an einen möglichen A-Standort angebunden werden, wenn dieser auch tatsächlich eingerichtet wird:

$$x_{ia} \leq y_a \quad \text{für alle } a \in A \text{ und } ia \in L.$$

Jeder eingerichtete A-Standort wird über genau eine Kette doppelt an das V-Netz angebunden:

$$\sum_{p \in P \text{ mit } a \in p} z_p = y_a \quad \text{für alle } a \in A.$$

Die Kapazität einer Kette muss mindestens so groß sein wie die Anschlussbandbreiten aller über sie angebundenen Standorte zusammen:

$$\sum_{a \in p} \Big(b_a + \sum_{ia \in L} b_i x_{ia} \Big) \leq c_p + \Big(\sum_{i \in A \cup N} b_i \Big) (1 - z_p) \quad \text{für alle } p \in P.$$

initial model:

- 1 billion variables

after reduction

- ~100.000 variables
- ~100.000 constraints

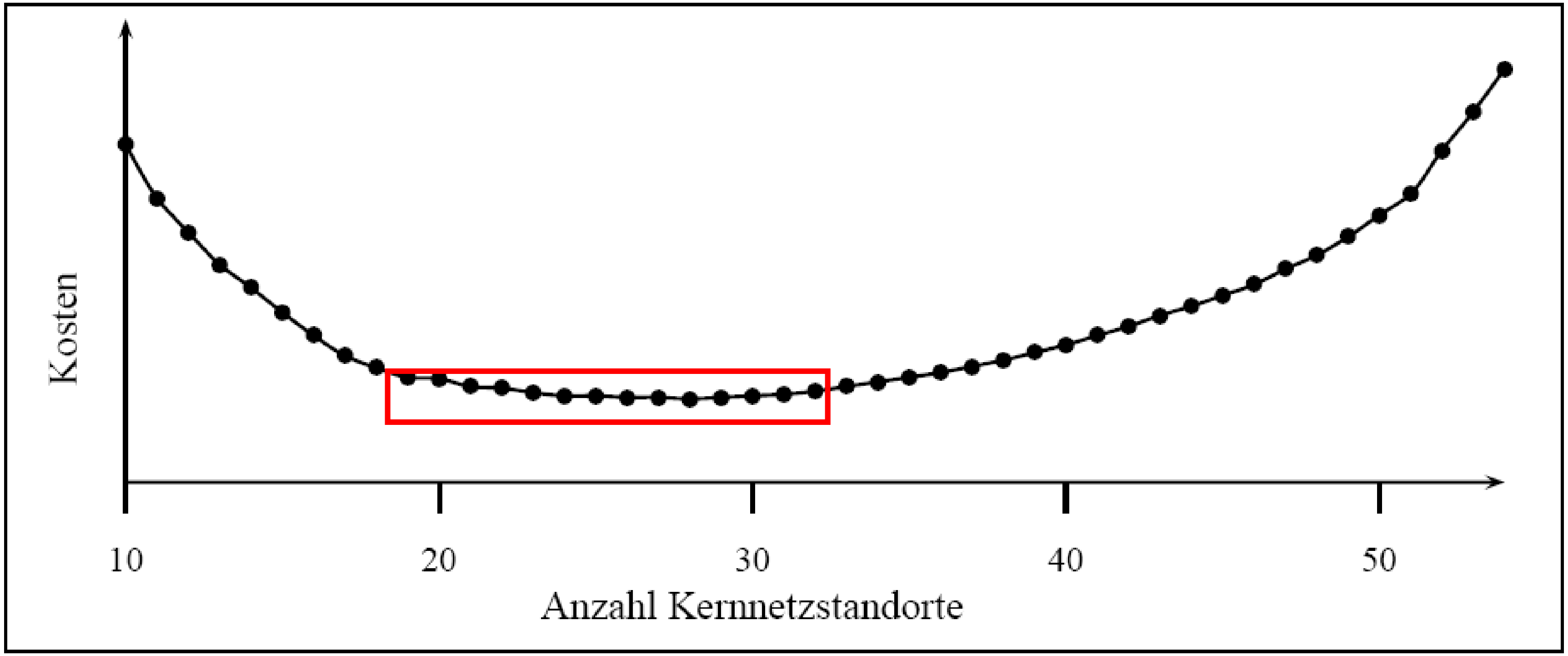
solved by ZIMPL/CPLEX

in a few minutes.

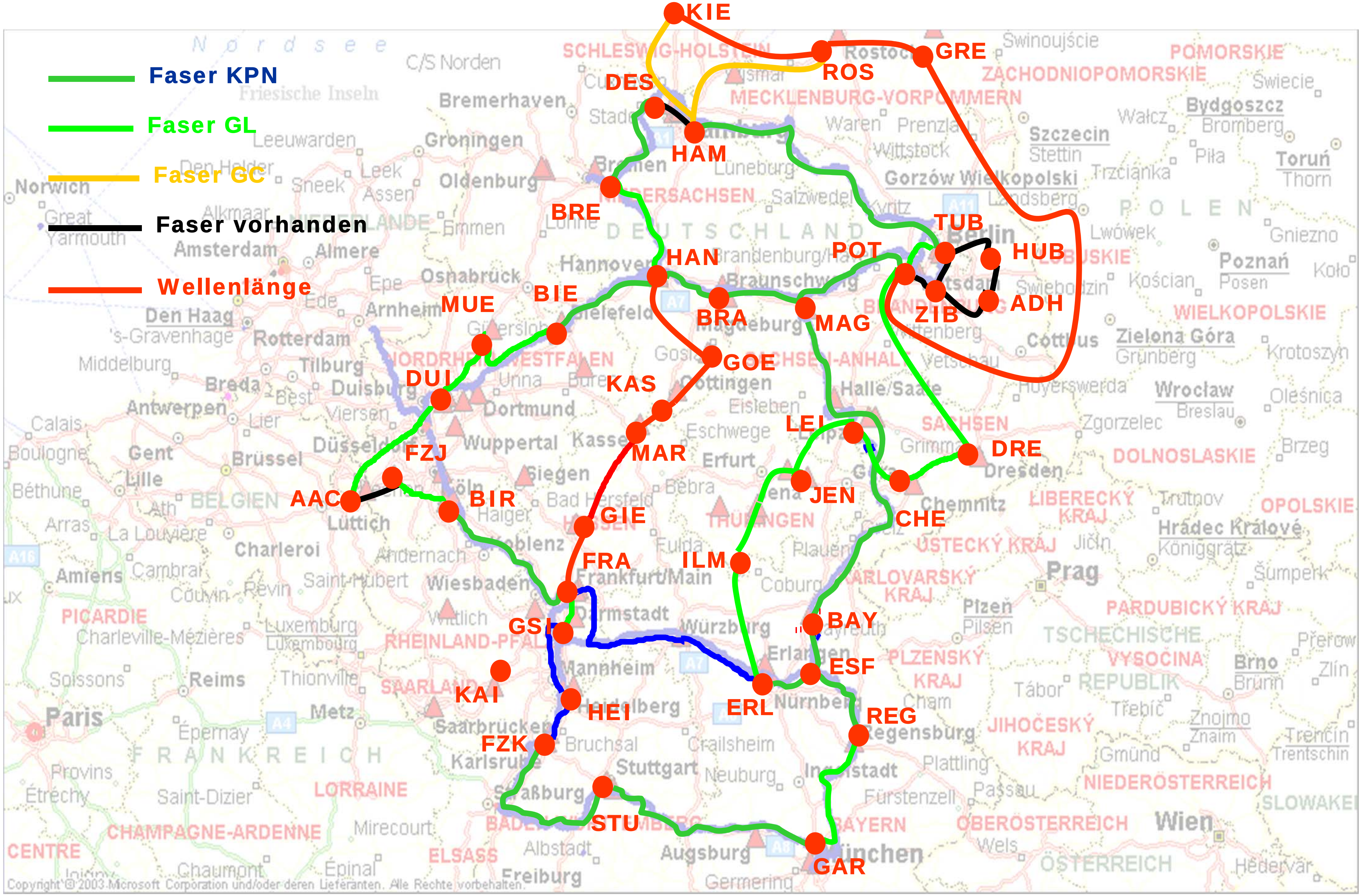
- 81 scenarios have been considered and solved – after lots of trials – for each choice of a reasonable number of core nodes.

Martin Grötschel

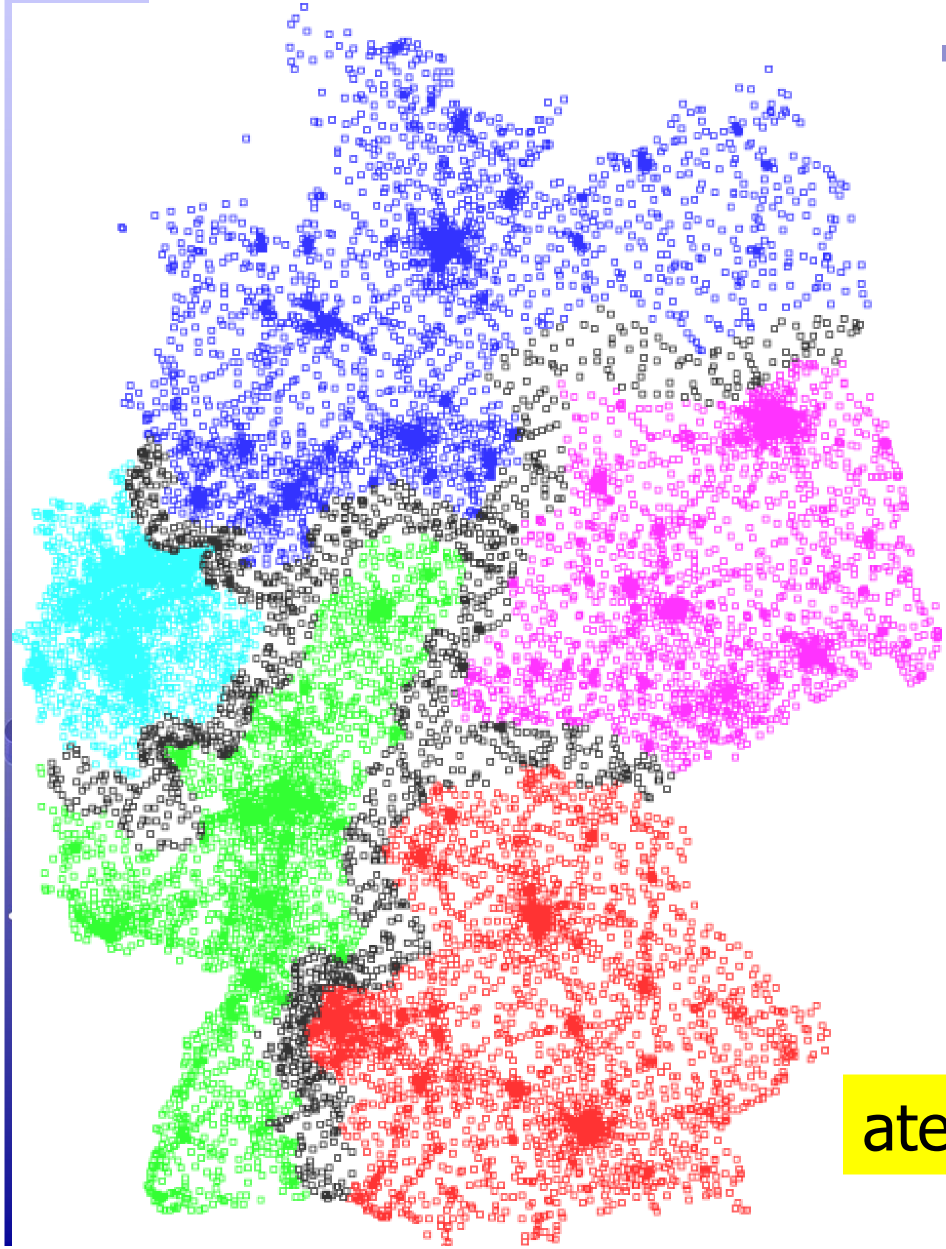
Number of Nodes in the Core Network



Location- and Network Topology Planing: solvable to optimality in practice



GSM 900-Optimization in Germany



1. Optimierung je Region aller
   - Standorte
   - Sektoren
   - Bänder
2. Zusammenführung der Ergebnisse aller Regionen
3. Optimierung eines Streifens entlang der Regionsgrenzen
4. Optimierung des 1800 MHz-Anteils von Dualband-Sektoren

atesio

Contents

Martin Grötschel

special „simple“ combinatorial optimization problems

Finding a

- minimum spanning tree
- shortest path
- maximum matching
- maximal flow through a network
- cost-minimal flow
- …

solvable in polynomial time by special purpose algorithms

Dijkstra algorithm for shortest paths

```
function Dijkstra(Graph, source):
    for each vertex v in Graph:            // Initializations
        dist[v] := infinity                // Unknown distance function from source to v
        previous[v] := undefined           // Previous node in optimal path from source
    dist[source] := 0                      // Distance from source to source
    Q := the set of all nodes in Graph
    // All nodes in the graph are unoptimized - thus are in Q
    while Q is not empty:                  // The main loop
        u := vertex in Q with smallest dist[]
        if dist[u] = infinity:
            break                          // all remaining vertices are inaccessible from source
        remove u from Q
        for each neighbor v of u:          // where v has not yet been removed from Q.
            alt := dist[u] + dist_between(u, v)
            if alt < dist[v]:              // Relax (u,v,a)
                dist[v] := alt
                previous[v] := u
    return previous[]
```

http://en.wikipedia.org/wiki/Dijkstra's_algorithm

Martin Grötschel

Special "hard" combinatorial optimization problems

- travelling salesman problem
- location und routing
- set-packing, partitioning, -covering
- max-cut
- linear ordering
- scheduling (with a few exceptions)
- node and edge colouring
- …

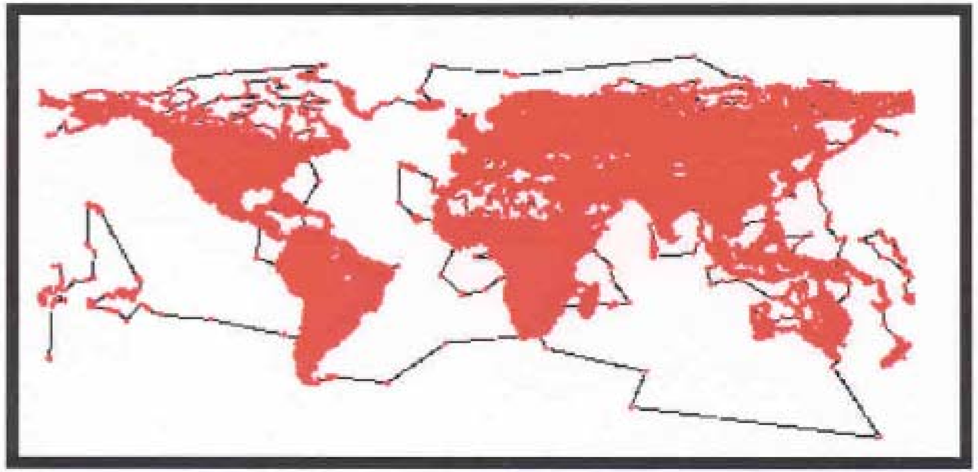
NP-hard (in the sense of complexity theory)

The most successful solution techniques employ linear programming. Lessons learned from these have no entered the general tools developed for general MIP solving.

Martin Grötschel

The importance of LP in IP solving
(slide from Bill Cook)

1,904,711–City World TSP, 2001



K	Optimality Gap
0	0.235%
8	0.190%
12	0.135%
14	0.111%
16	0.103%

Solution of LP Problems takes over 99% of CPU time

of variables = 1,813,961,044,405 = 1,8 trillion

Martin Grötschel

LP solvers in MIP solving

- A big computational issue for the simplex method is degeneracy.
- And LP relaxations of IPs/MIPs tend to be enormously degenerate.

Martin Grötschel

Mixed Integer Programming

$$
\begin{aligned}
&\min c^T x \\
&Ax = a \\
&Bx \le b \\
&\quad x \ge 0 \\
&\text{some } x_j \in \mathbb{Z} \\
&\text{some } x_k \in \{0,1\}
\end{aligned}
$$

(linear) 0/1- or
mixed-integer
program
IP, MIP

Contents

Martin Grötschel

Heuristics: A Survey

- Greedy Algorithms
- Exchange & Insertion Algorithms
- Neighborhood/Local Search
- Variable Neighborhood Search, Iterated Local Search
- Random sampling
- Simulated Annealing
- Taboo search
- Great Deluge Algorithms
- Simulated Tunneling
- Neural Networks
- Scatter Search
- Greedy Randomized Adaptive Search Procedures

Heuristics: A Survey

- Genetic, Evolutionary, and similar Methods
- DNA-Technology
- Ant and Swarm Systems
- (Multi-) Agents
- Population Heuristics
- Memetic Algorithms (Meme are the “missing links” gens and mind)
- Fuzzy Genetics-Based Machine Learning
- Fast and Frugal Method (Psychology)
- Method of Devine Intuition (Psychologist Thorndike)
- …..

An Unfortunate Development

- There is a marketing battle going on with unrealistic, or even ideological, claims about the quality of heuristics – just to catch attention
- Linguistic Overkill:
- Vodoo Approach

The state of heuristics in MIP is "not optimal"

Primal MIP heuristics, Examples:

- Rounding Heuristics
- Diving Heuristics
- Large Neighborhood Search
- Local Branching
- RINS
- Crossover
- DINS
- RENS (Relaxation Enforced Neighborhood Search)
- Feasibility Pump(s)
- Undercover

TIMO BERTHOLD

Heuristics of the Branch-Cut-and-Price-Framework SCIP

ZIB-Report 07-30 (October 2007)

Martin Grötschel

Contents

Martin Grötschel

Branching

- Branching is “without mathematical theory”.
- Implementation, however, is a major issue!
- New branching rules played an important role in the last years for the improvement of MIP codes.
- Their evaluation is based on “experimental analysis”.

Martin Grötschel

The Branch&Bound Technique: An Example

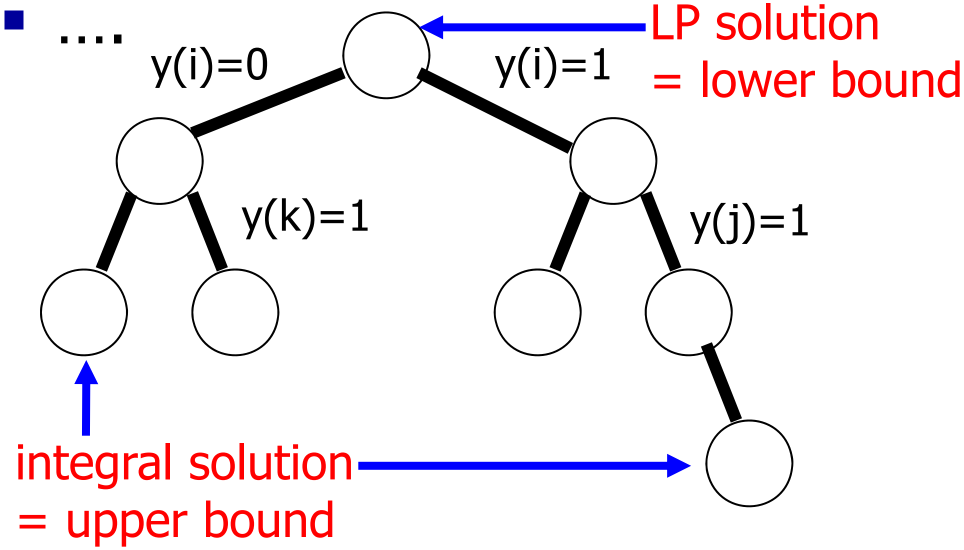
$$\begin{aligned} &\min c^T x \\ &Ax = a \\ &Bx \le b \\ &x \ge 0 \\ &x \in \{0,1\}^n \end{aligned}$$

0/1-program

$$\begin{aligned} &\min c^T x \\ &Ax = a \\ &Bx \le b \\ &x \ge 0 \\ &\sout{x \in \{0,1\}^n} \\ &x \le 1 \end{aligned}$$

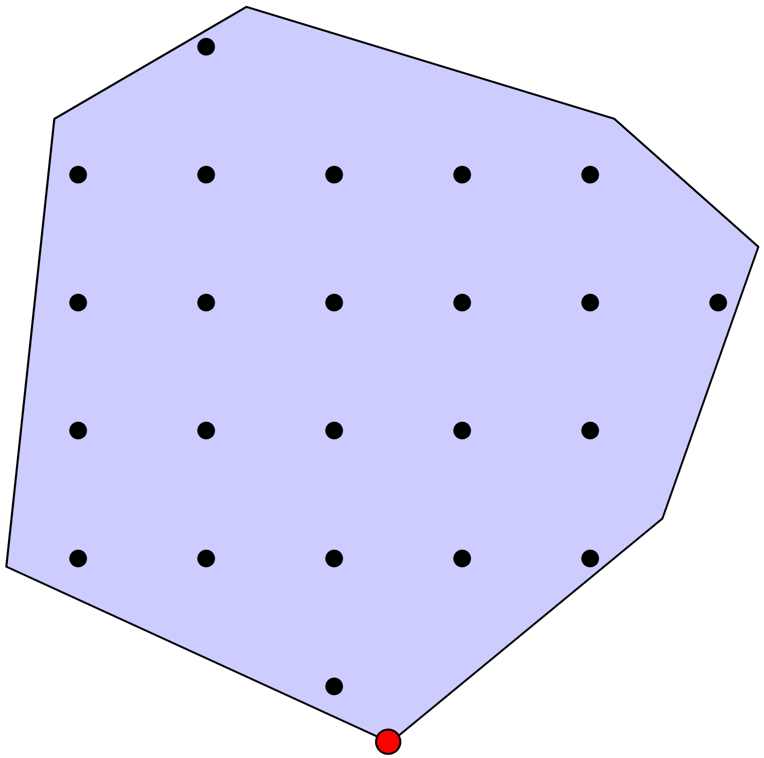
LP-relaxation

- Solve the LP-relaxation and get optimal solution y. (lower bound)
- If y integral, DONE!
- Otherwise pick fractional component y(i).
- Create two new subproblems by adding y(i)=1 and y(i)=0, resp.
- ....



Martin Grötschel

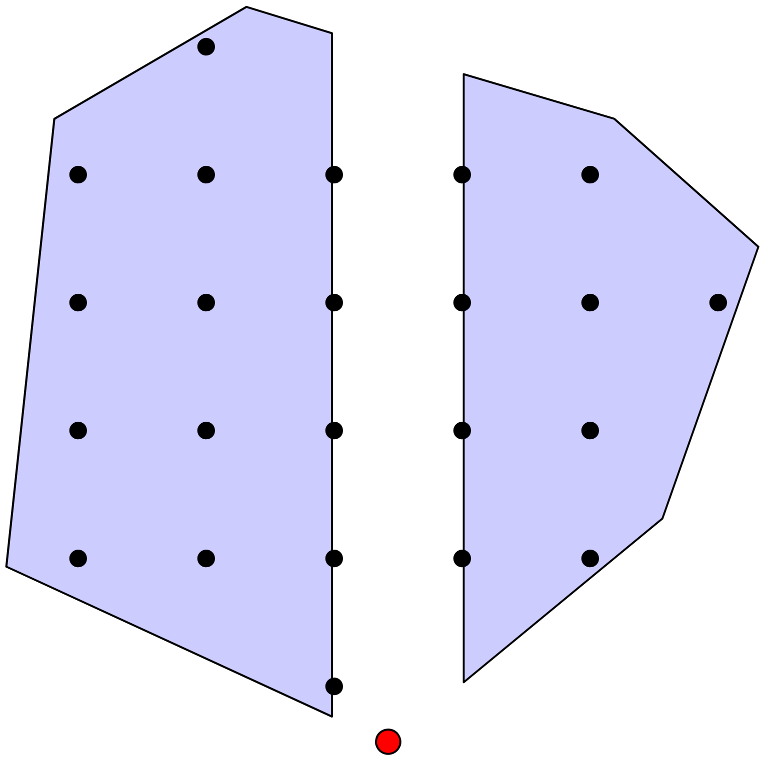
Branching (in general)



- Current solution is infeasible

Branching (in general)

- Rounding a fractional component up and down



- Decomposition into subproblems removes infeasible solution

Martin Grötschel

A Branching Tree

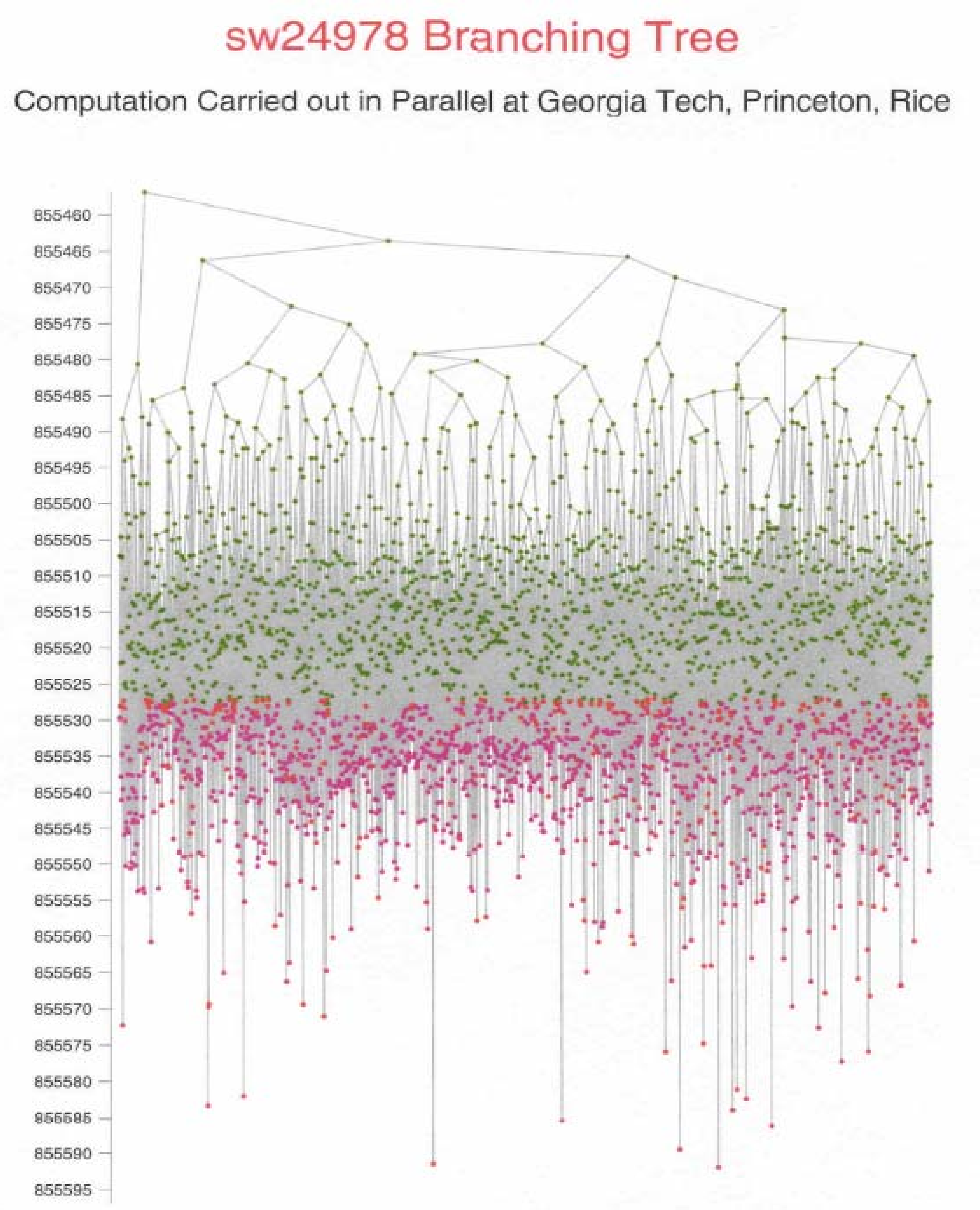
Applegate

Bixby

Chvátal

Cook

tree copied from
www1.ctt.dtu.dk/ROUTE2003/presentations/cook.pdf



Contents

Martin Grötschel

Cutting plane technique for integer and mixed-integer programming

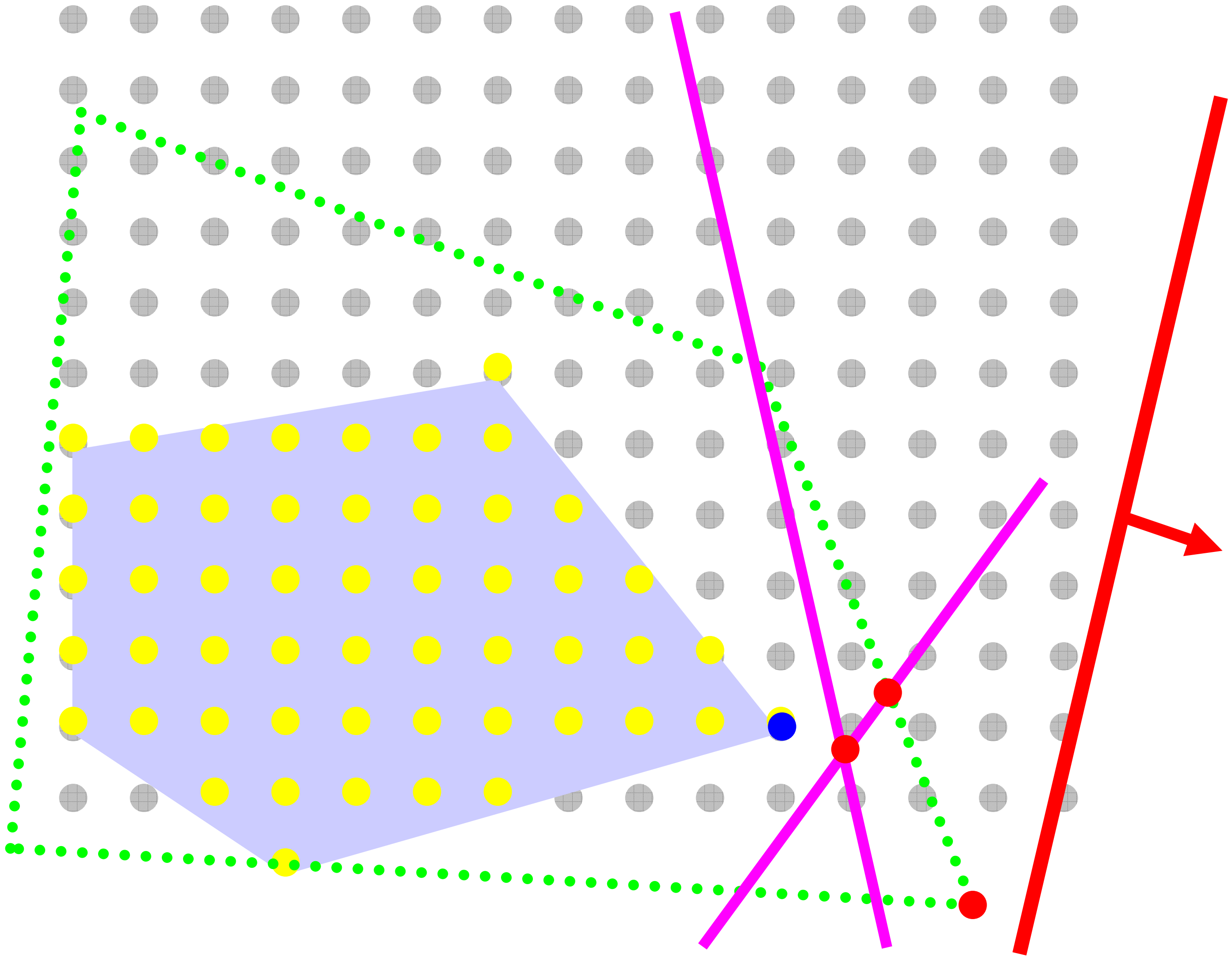
Feasible integer solutions

Objective function

Convex hull

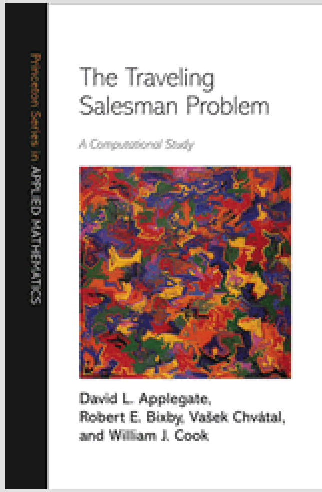
LP-based relaxation

Cutting planes



Martin Grötschel

Cutting Plane Theory & Practice



The Traveling Salesman Problem: A Computational Study
David L. Applegate, Robert E. Bixby, Vasek Chvátal & William J. Cook

Winner of the 2007 Lanchester Prize, Informs

Cloth | 2007 | **$70.00** / £48.95
606 pp. | 6 x 9 | 200 line illus.

Shopping Cart | **Reviews** | **Table of Contents**
Chapter 1 [PDF]

Gérard Cornuéjols *

Valid Inequalities for Mixed Integer Linear Programs

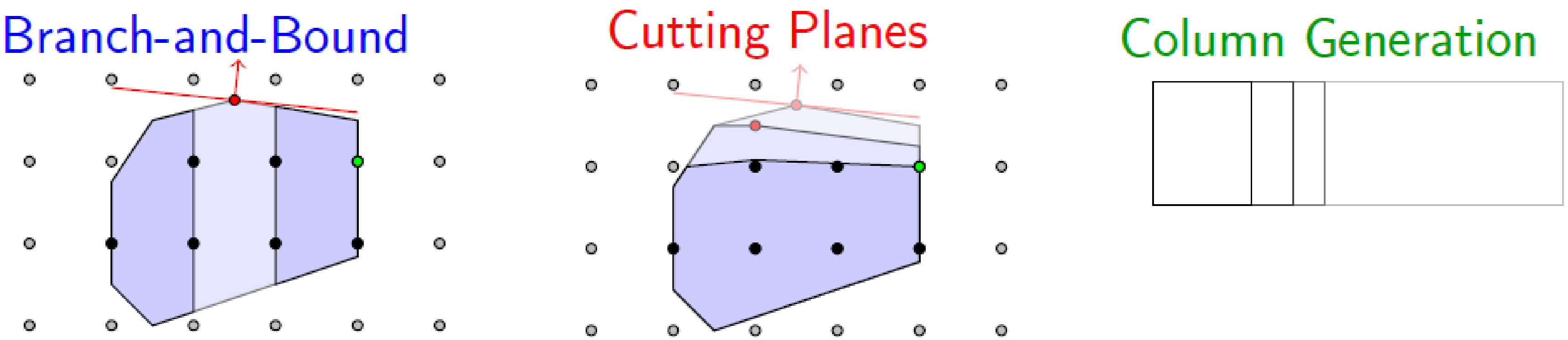
December 2005, revised July 2006

Abstract. This tutorial presents a theory of valid inequalities for mixed integer linear sets. It introduces the necessary tools from polyhedral theory and gives a geometric understanding of several classical families of valid inequalities such as lift-and-project cuts, Gomory mixed integer cuts, mixed integer rounding cuts, split cuts and intersection cuts, and it reveals the relationships between these families. The tutorial also discusses computational aspects of generating the cuts and their strength.

Mathematical Programming, Volume 112, 2008, pp 3-44

Other Names

- Branch & Cut
- Branch & Price
- Branch & Cut & Price
- etc.

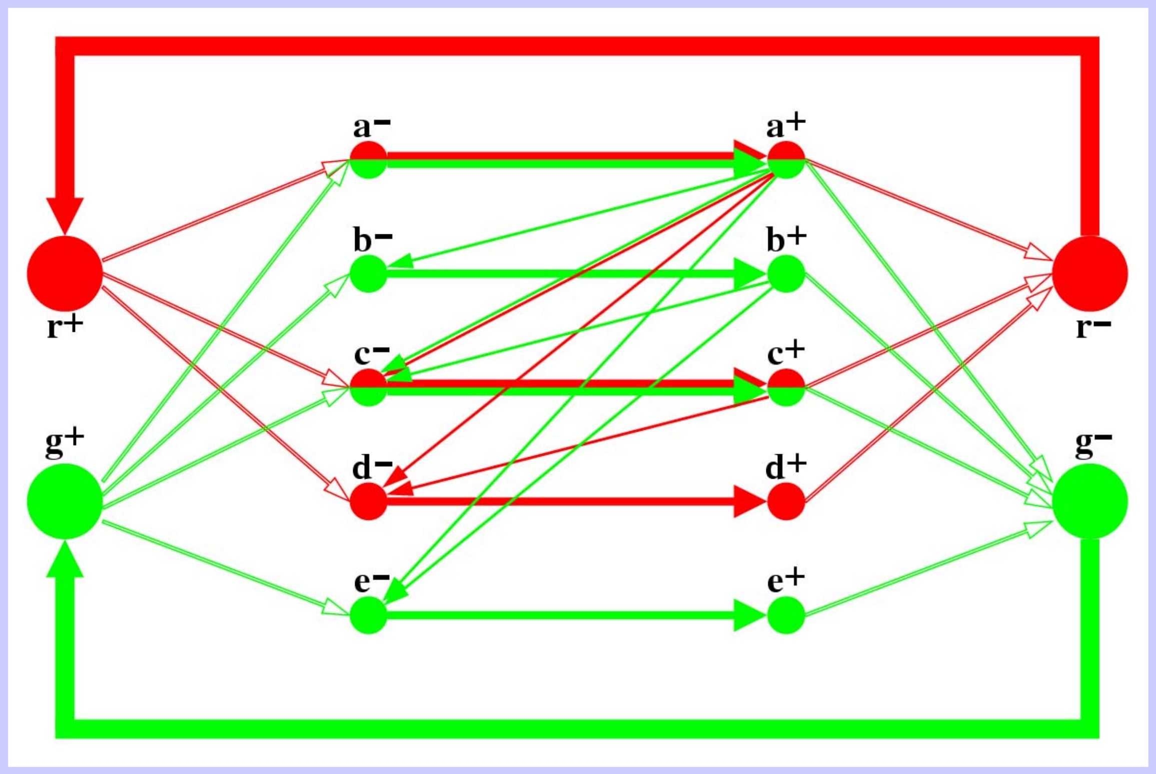


Contents

Martin Grötschel

Bus circulation:
Flow through a bus line network

Multicommodity flow with minimum cost



Urban Scenarios

	BVG	*HHA*	*VHH*
depots	10	14	10
vehicle types	44	40	19
timetabled trips	25 000	16 000	5 500
number of variables	70 000 000	15 100 000	10 000 000
cpu mins	200	50	28

SPEC CINT2006 Benchmarks - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe http://www.spec.org/cpu2006/CINT2006/

SPEC CINT2006 Benchmarks

spec

Standard Performance Evaluation Corporation

home | benchmarks | results | contact | site map | site search | help

Results

- Published Results
- Results Search
- OSG Fair Use Policy

Information

- CPU2006
- Documentation
  - Documentation Overview
  - Run & Reporting Rules
  - Readme1st
- FAQ

Press and Publications

- V1.0 Release
- V1.1 Release
- Related Publications

Order Benchmarks

- Order CPU2006

Resources

- Site Map
- Site Search
- Site Index
- Glossary
- Performance Links

CINT2006 (Integer Component of SPEC CPU2006):

Benchmark	Language	Application Area	Brief Description
400.perlbench	C	Programming Language	Derived from Perl V5.8.7. The workload includes SpamAssassin, MHonArc (an email indexer), and specdiff (SPEC's tool that checks benchmark outputs).
401.bzip2	C	Compression	Julian Seward's bzip2 version 1.0.3, modified to do most work in memory, rather than doing I/O.
403.gcc	C	C Compiler	Based on gcc Version 3.2, generates code for Opteron.
429.mcf	C	Combinatorial Optimization	Vehicle scheduling. Uses a network simplex algorithm (which is also used in commercial products) to schedule public transport.
445.gobmk	C	Artificial Intelligence: Go	Plays the game of Go, a simply described but deeply complex game.
456.hmmer	C	Search Gene Sequence	Protein sequence analysis using profile hidden Markov models (profile HMMs)
458.sjeng	C	Artificial Intelligence: chess	A highly-ranked chess program that also plays several chess variants.
462.libquantum	C	Physics / Quantum Computing	Simulates a quantum computer, running Shor's polynomial-time factorization algorithm.
464.h264ref	C	Video Compression	A reference implementation of H.264/AVC, encodes a videostream using 2 parameter sets. The H.264/AVC standard is expected to replace MPEG2
471.omnetpp	C++	Discrete Event Simulation	Uses the OMNet++ discrete event simulator to model a large Ethernet campus network.
473.astar	C++	Path-finding Algorithms	Pathfinding library for 2D maps, including the well known A* algorithm.
483.xalancbmk	C++	XML Processing	A modified version of Xalan-C++, which transforms XML documents to other document types.

Home - Contact - Site Map - Privacy - About SPEC

webmaster@spec.org
Last updated: *Thu Aug 24 00:44:00 EDT 2006*

URL: *http://www.spec.org/cpu2006/CINT2006/index.html*

Suchen: bench Abwärts Aufwärts Hervorheben Groß-/Kleinschreibung

http://www.spec.org/cpu2006/Docs/429.mcf.html Now: Stark bewölkt, 5° C Mi: 7° C Do: 8° C

MCF Literature

(on Löbel's implementation of the Min-Cost Flow algorithm)

- Marty Itzkowitz, Brian J. N. Wylie, Christopher Aoki, and Nicolai Kosche: Memory Profiling using Hardware Counters
- ARCTiC Labs: 181.mcf - Datasets profile vs. Reference Dataset
- Joshua J. Yi, Resit Sendag, and David J. Lilja: Increasing Instruction-Level Parallelism with Instruction Precomputation
- Jinwoo Kim, Weng-Fai Wong, and Drishna V. Palem: Data Prefetching using Off-line Learning
- Resit Sendag, Peng-fei Chuang, and David J. Lilja: Address Correlation: Exceeding the Limits of Locality
- Kim M. Hazelwood, Mark C. Toburen, and Thomas M. Conte: A Case for Exploiting Memory-Access Persistence
- Ian R. Bratt, Alex Settle, and Daniel A. Connors: Predicate-Based Transformations to Eliminate Control and Data-Irrelevant Cache Misses
- Andreas Stiller: Hammer, Nägel und Köpfe: Das Microprocessor Forum 2001, c't 23/2001, S. 28

Contents

Martin Grötschel

Column Generation

Column Generation can be viewed as a procedure dual to the cutting plane method. The basic principle:

1. Select a small number of variables and solve the linear program (or LP relaxation) using only these.
2. Find an unused variable (or several) which, if included, would (most) improve the objective value, or determine that there is none, i.e., the linear program has been solved: stop.

   Solve the **column generation subproblem (Pricing).**

   Model this as an optimization problem, works best if this is an easy IP (shortest path, dynamic program, etc.)
3. Include the variable(s) in the linear program, re-solve it, and go to step 2.

Successful Applications: Examples

1960 Cutting stock problems (Gilmore, Gomory)

Air crew scheduling

Aircraft fleeting and routing

Crew rostering

Vehicle routing

Driver assignment

Global shipping

Multi-item lot-sizing

Telecommunications network design

Cancer radiation treatment using IMRT

Column Generation

- This method is particularly important if lots of rules have to be satisfied that are not easy to model, blow up the IP enormously and are subject to frequent changes, such as the work rules for drivers in public transport or pilots and crew in the airline business.
  (I can give a full hour lecture on "break rules".)
- The "difficult side constraints" are treated in the column generation subroutine(s).

Martin Grötschel

Column Generation

- Typical application for column generation: set partitioning

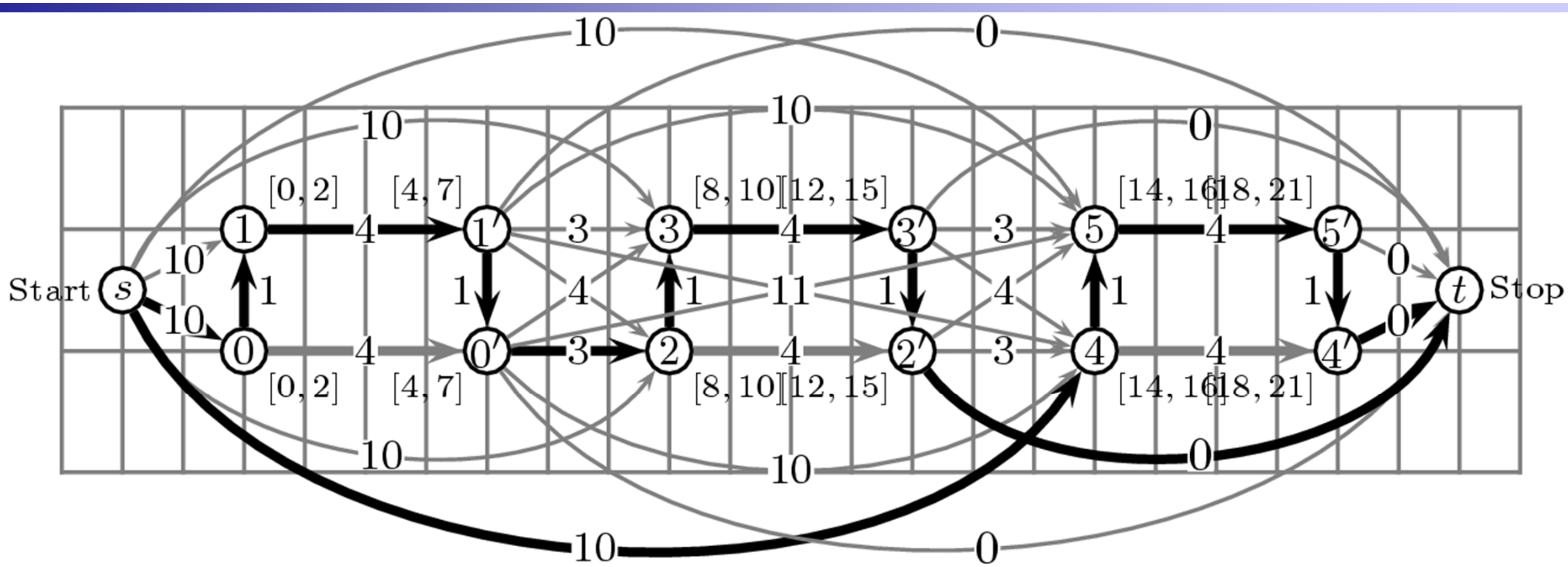
$$\min c^T x$$
$$Ax = 1$$
$$x \geq 0$$
$$x \in \{0,1\}^n$$
where A is a 0/1-matrix

We have used this in many of the projects mentioned.

Martin Grötschel

Telebus
(transportation of disabled persons)

Graph Theoretical Model
IP: Set Partitioning Model



	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	c_{20}	c_{21}	c_{22}	c_{23}	c_{24}
	14	14	14	14	14	14	16	21	22	28	29	22	21	29	28	16	21	22	22	21	16	23	23	30	30
i	$a_{.0}$	$a_{.1}$	$a_{.2}$	$a_{.3}$	$a_{.4}$	$a_{.5}$	$a_{.6}$	$a_{.7}$	$a_{.8}$	$a_{.9}$	$a_{.10}$	$a_{.11}$	$a_{.12}$	$a_{.13}$	$a_{.14}$	$a_{.15}$	$a_{.16}$	$a_{.17}$	$a_{.18}$	$a_{.19}$	$a_{.20}$	$a_{.21}$	$a_{.22}$	$a_{.23}$	$a_{.24}$
0	1	.	.	.	.	.	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	1	1	1	1
1	.	1	.	.	.	.	1	.	.	.	.	1	1	1	1	.	.	.	.	.	.	1	1	1	1
2	.	.	1	.	.	.	.	1	.	.	.	1	.	.	.	1	1	1	.	.	.	1	.	.	.
3	.	.	.	1	.	.	.	.	1	.	.	.	1	.	.	1	.	.	1	1	.	.	1	.	.
4	.	.	.	.	1	.	.	.	.	1	.	.	.	1	.	.	1	.	1	.	1	.	.	1	.
5	.	.	.	.	.	1	.	.	.	.	1	.	.	.	1	.	.	1	.	1	1	.	.	.	1
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

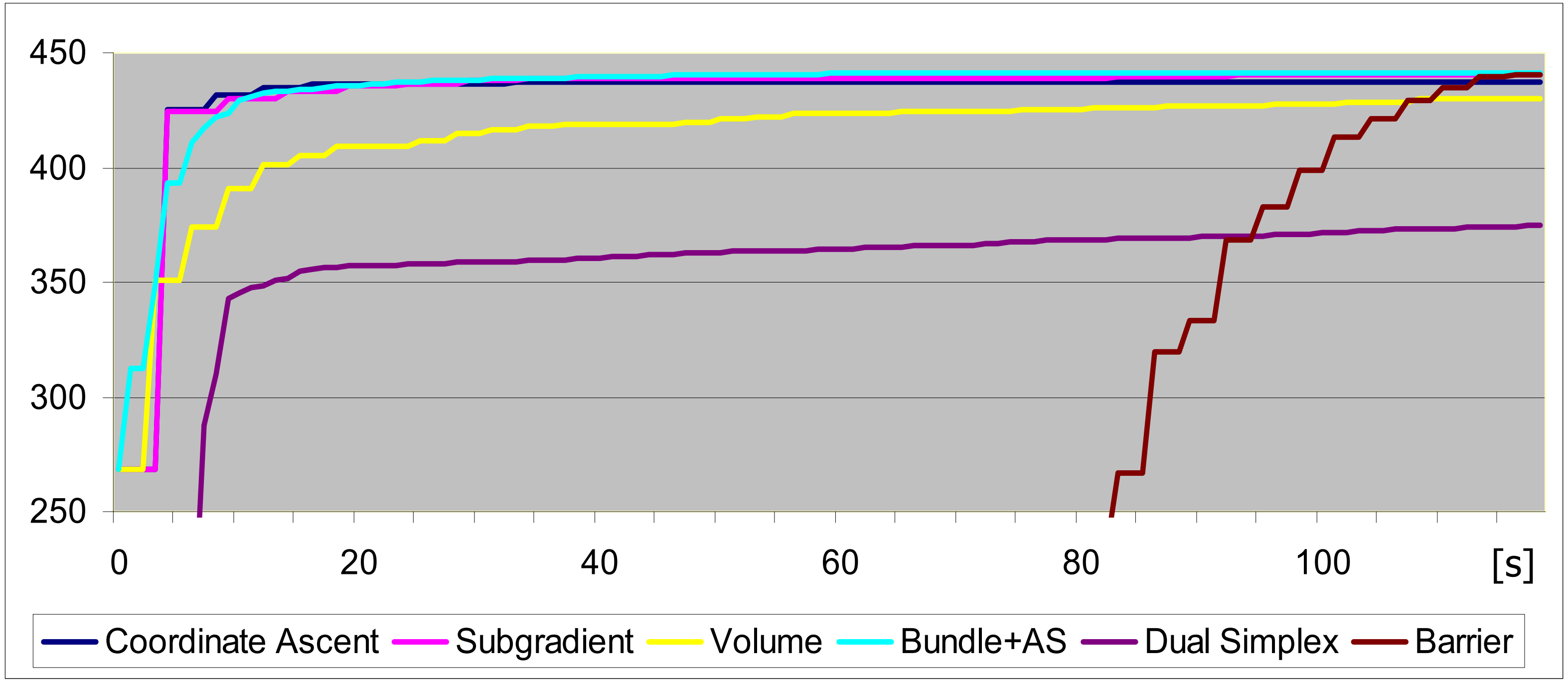
	c_{25}	c_{26}	c_{27}	c_{28}	c_{29}	c_{30}	c_{31}	c_{32}	c_{33}	c_{34}	c_{35}	c_{36}	c_{37}	c_{38}	c_{39}	c_{40}	c_{41}	c_{42}	c_{43}	c_{44}	c_{45}	c_{46}	c_{47}	c_{48}
	23	28	29	30	29	30	23	29	30	29	28	30	23	23	25	30	31	31	30	32	30	31	31	30
i	$a_{.25}$	$a_{.26}$	$a_{.27}$	$a_{.28}$	$a_{.29}$	$a_{.30}$	$a_{.31}$	$a_{.32}$	$a_{.33}$	$a_{.34}$	$a_{.35}$	$a_{.36}$	$a_{.37}$	$a_{.38}$	$a_{.39}$	$a_{.40}$	$a_{.41}$	$a_{.42}$	$a_{.43}$	$a_{.44}$	$a_{.45}$	$a_{.46}$	$a_{.47}$	$a_{.48}$
0	1	1	1	1	1	1	.	.	.	.	.	.	.	.	1	1	1	1	1	1	1	1	.	.
1	.	.	.	.	.	.	1	1	1	1	1	1	.	.	1	1	1	1	1	1	.	.	1	1
2	1	1	1	.	.	.	1	1	1	.	.	.	1	.	1	1	1	.	.	.	1	.	1	.
3	1	.	.	1	1	.	1	.	.	1	1	.	.	1	1	.	.	1	1	.	.	1	.	1
4	.	1	.	1	.	1	.	1	.	1	.	1	1	1	.	1	.	1	.	1	1	1	1	1
5	.	.	1	.	1	1	.	.	1	.	1	1	1	1	.	.	1	.	1	1	1	1	1	1
	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}	x_{39}	x_{40}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}

Coputational Results for a (Duty Scheduling) Set Partitioning Model

Duty Scheduling Problem Ivu41:

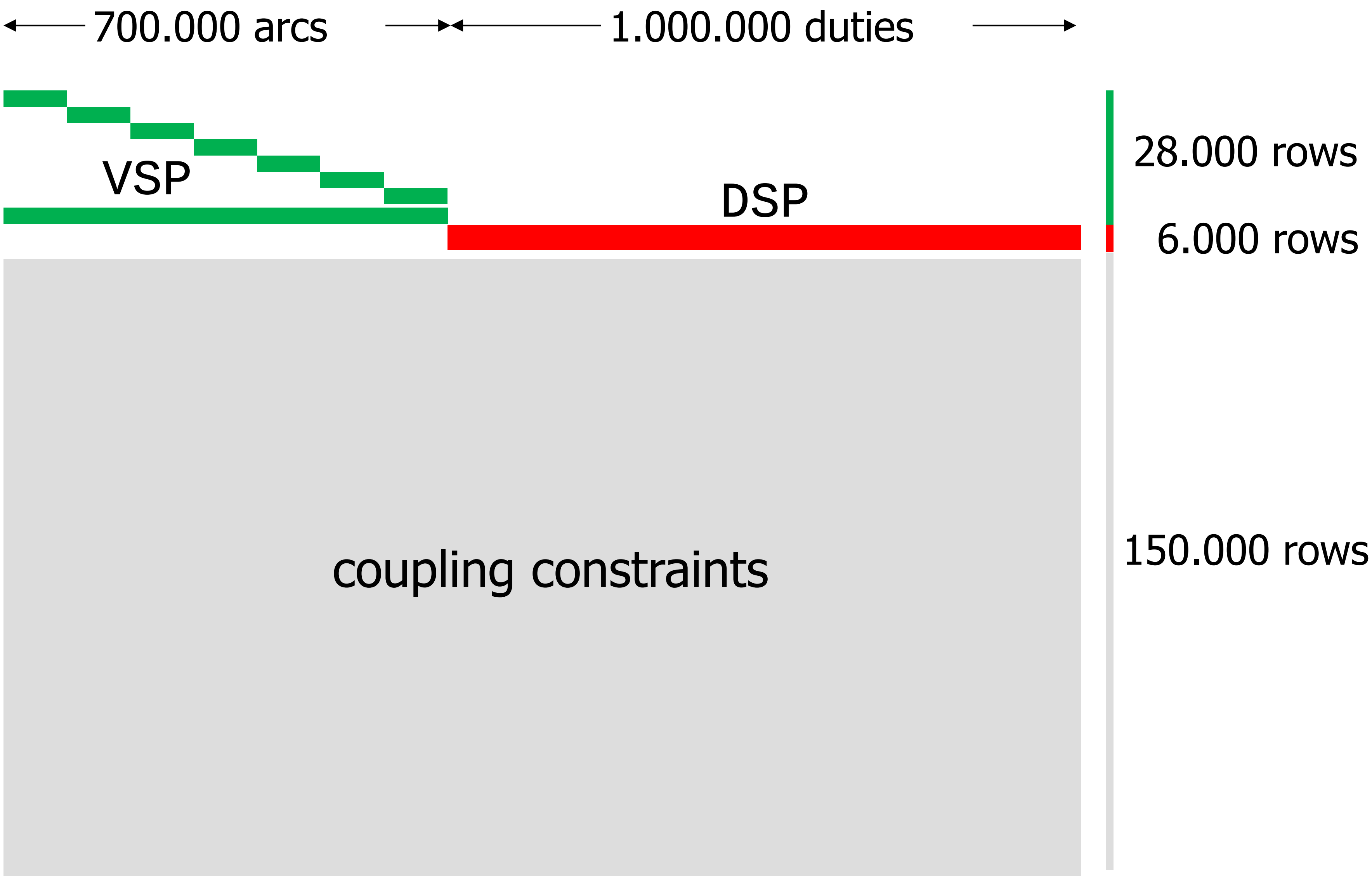
- 870 500 col
- 3 570 rows
- 10.5 non-zeroes per col

Coordinate Ascent:	Fast, low quality
Subgradient:	(Theoretical) Convergence
Volume:	Primal approximation
Bundle+AS:	Conv. + primal approx.
Dual Simplex:	Primal+dual optimal
Barrier:	Primal+dual optimal



Ralf Borndörfer

Integrated Bus and Driver Scheduling: Model Structure



Contents

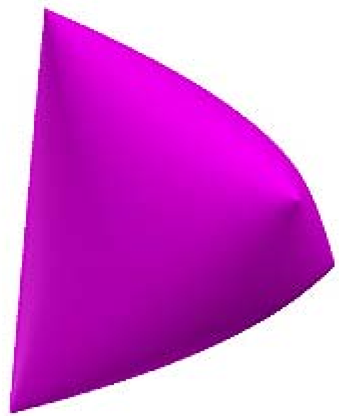
Martin Grötschel

Other Methods

- Dantzig-Wolfe decomposition
- Benders' decomposition
- Algebraic approach, lattice point methods, Gröbner basis techniques, test sets

Smooth relaxations

- Instead of an LP relaxation, one can consider "smooth relaxations" such as "semidefinite relaxations" or polynomial equations/inequalities.
- Theoretical success: stable sets in perfect graphs
- Some practical success in max-cut algorithms



elliptope formed from all positive semidefinite 3x3 matrices having 1's on the main diagonal

- Some theoretical/practical success in mathematical proofs.
- In general: Possible success in special cases, not a general tool yet.

Martin Grötschel

Contents

Martin Grötschel

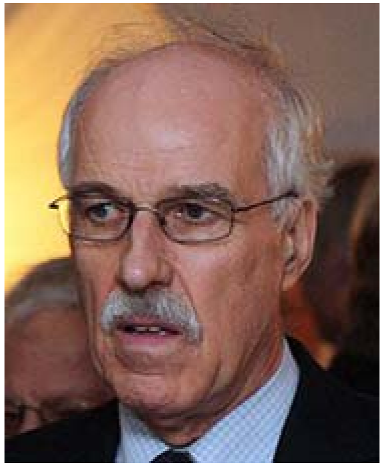
Independent Testing

Benchmarks for Optimization Software

by Hans Mittelmann (mittelmann at asu.edu)

The following are NEOS solvers we have installed.

BNBS, BPMPD, BPMPD-AMPL, Concorde, CONDOR, CSDP, DDSIP, FEASPUMP, FEASPUMP-AMPL, ICOS, NSIPS, PENBMI, PENSDP, QSOPT_EX, SCIP, SCIP-AMPL, SDPA, SDPLR, SDPT3, SeDuMi



http://plato.asu.edu/bench.html

LINEAR PROGRAMMING

- Benchmark of serial LP solvers (10-12-2010)
- Benchmark of parallel LP solvers (10-16-2010)
- Parallel CPLEX, GUROBI, and MOSEK on LP problems (7-18-2010)
- Large Network-LP Benchmark (commercial vs free) (10-16-2010)

MIXED INTEGER LINEAR PROGRAMMING

- MILP Benchmark - serial codes (10-15-2010)
- MILP Benchmark - parallel codes (10-14-2010)
- MILP cases that are difficult for some codes (10-8-2010)
- Feasibility Benchmark - Feaspump,CPLEX,SCIP,GUROBI (10-15-2010)
- Infeasibility Detection for MILP Problems (10-14-2010)

http://miplib.zib.de/

MIPLIB - Mixed Integer Problem Library

In response to the needs of researchers for access to real-world mixed integer programs a group of researchers Robert E. Bixby, E.A. Boyd and R.R. Indovina created in 1992 the MIPLIB, an electronically available library of both pure and mixed integer programs. This was updated in 1996 by Robert E. Bixby, Sebastian Ceria, Cassandra M. McZeal, and Martin W.P. Savelsbergh.

Since its introduction, MIPLIB has become a standard test set used to compare the performance of mixed integer optimizers. Its availability has provided an important stimulus for researchers in this very active area.

MIPLIB 2010

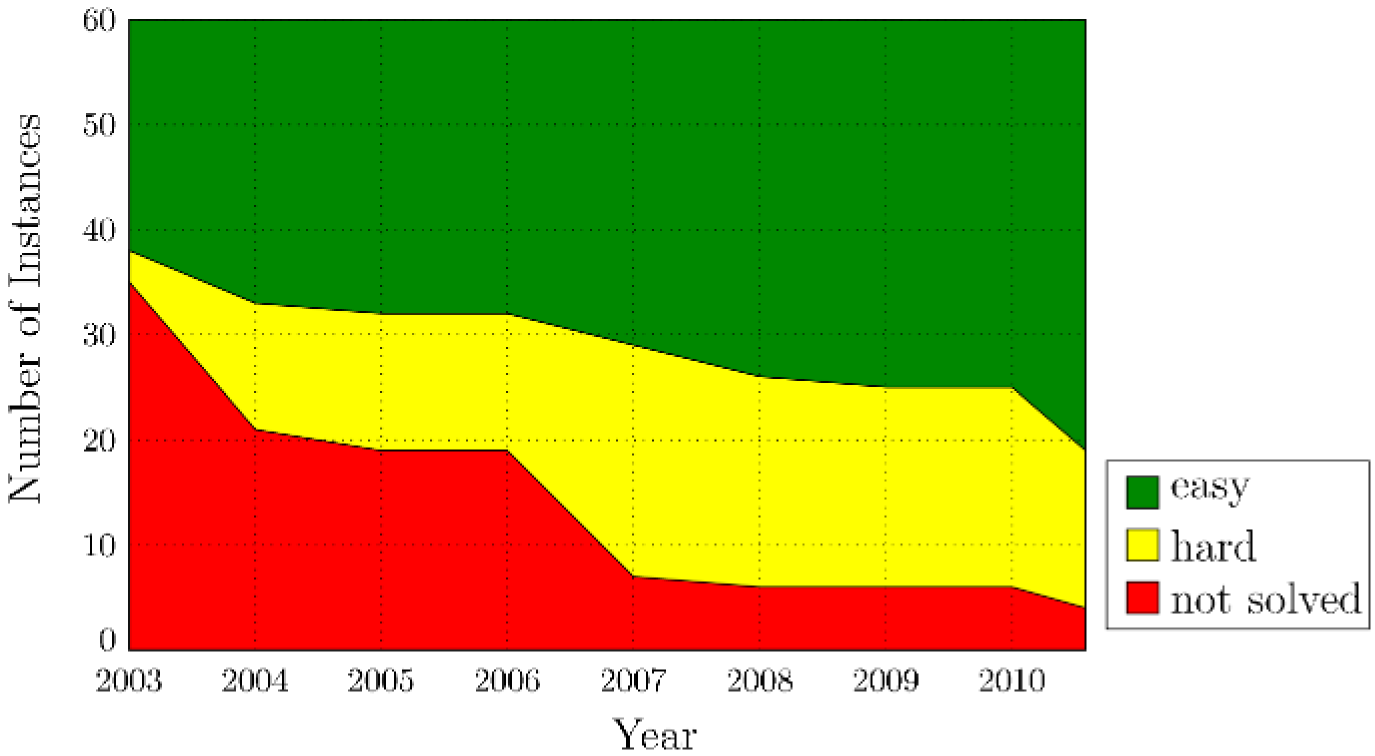
The collection of instances for the next version of the MIPLIB is finished (we had about 60 submitters).

These instances were evaluated and put together into some test sets, including a general benchmark set, a challenge set with hard and unsolved instances, and specialized sets that focus on problems with a specific property. This includes testsets with huge, infeasible, and numerically unstable problems, as well as testsets where finding the optimal primal solution is the major issue, where the LP resolve takes long at each node, and where a large enumeration tree is created during the search.

An updated beta version of MIPLIB 2010 can be downloaded here. The final version of MIPLIB 2010 will be released within the next weeks, the web page will then also be updated, presenting contributors, background and statistics about the instances of the final MIPLIB 2010.

http: / / miplib.zib.de /

Development over the years



Comparison of the number of solved MIPLIB 2003 instances at the beginning of each year.
'Easy' means, that the instance could be solved within one hour using a commercial MIP-solver, 'hard' stands for instances, that were solved, but not in the previous conditions.

Martin Grötschel

http://miplib.zib.de/

- fifth version of the Mixed Integer Programming Library, first established in 1992
- program committee: people from Cbc, Cplex, Gurobi, Mosek, Scip, Xpress
- 1000 instances submitted by 60 contributors
- data-mining the public domain gave another 1000 instances
- final benchmark set: 90 instances
- special test sets:
  - infeasible: instances which dot not have any integral solution
  - challenge: unsolved and very hard instances
  - XXL: millions of variables, constraints or nonzeros.
  - largeTree: millions of nodes in branch-and-bound tree
  - hardLP: many iterations per LP resolve
  - hardPrimal: root LP optimum = IP optimum, "lucky guess" suffices
  - numerics: unstable behaviour, large conditon number
- `http://miplib.zib.de`

Martin Grötschel

Contents

Martin Grötschel

How does a Branch-and-Cut MIP Solver work?

Mixed Integer Program (MIP)

Characteristics

Objective function

- linear function

Feasible region

- described by linear constraints

Variable domains

- real or integer values

$$\min c^T x$$

$$Ax = a$$

$$Bx \leq b$$

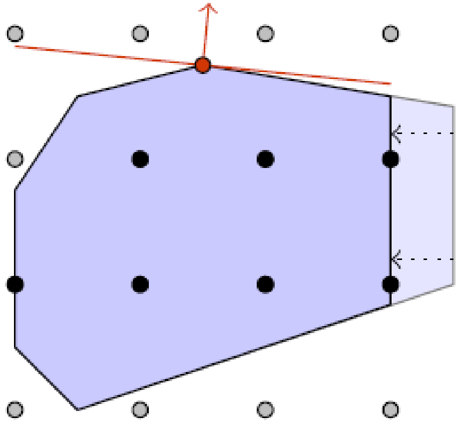
$$x \geq 0$$

$$\text{some } x_j \in \mathbb{Z}$$

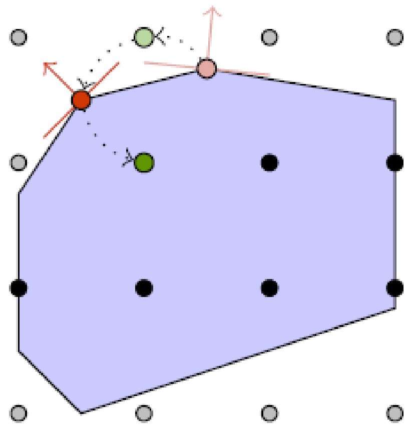
$$\text{some } x_k \in \{0,1\}$$

MIP Solver Techniques

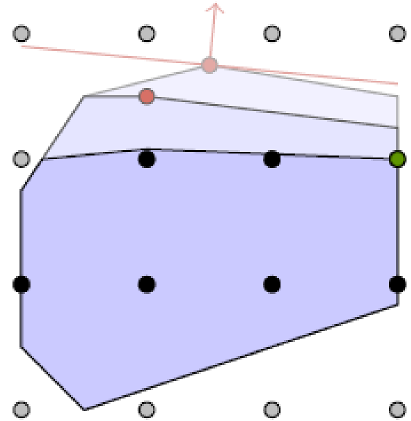
Presolving



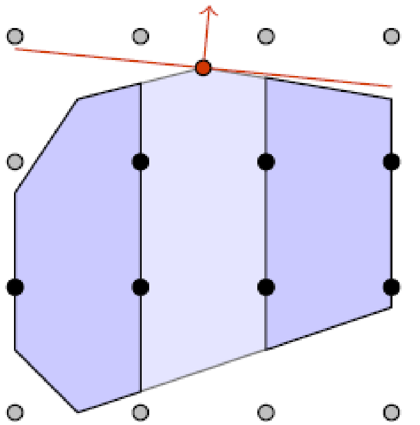
Primal Heuristics



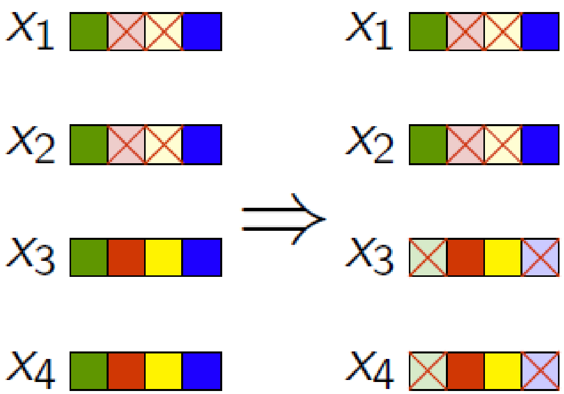
Cutting Planes



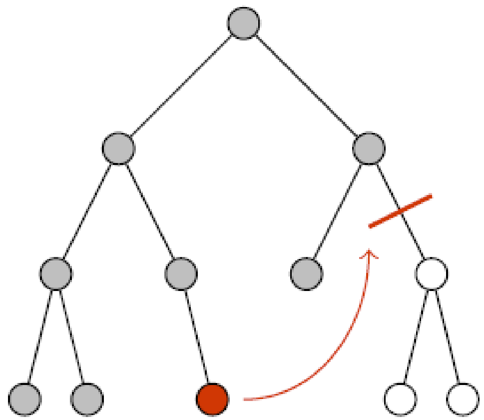
Branch & Bound



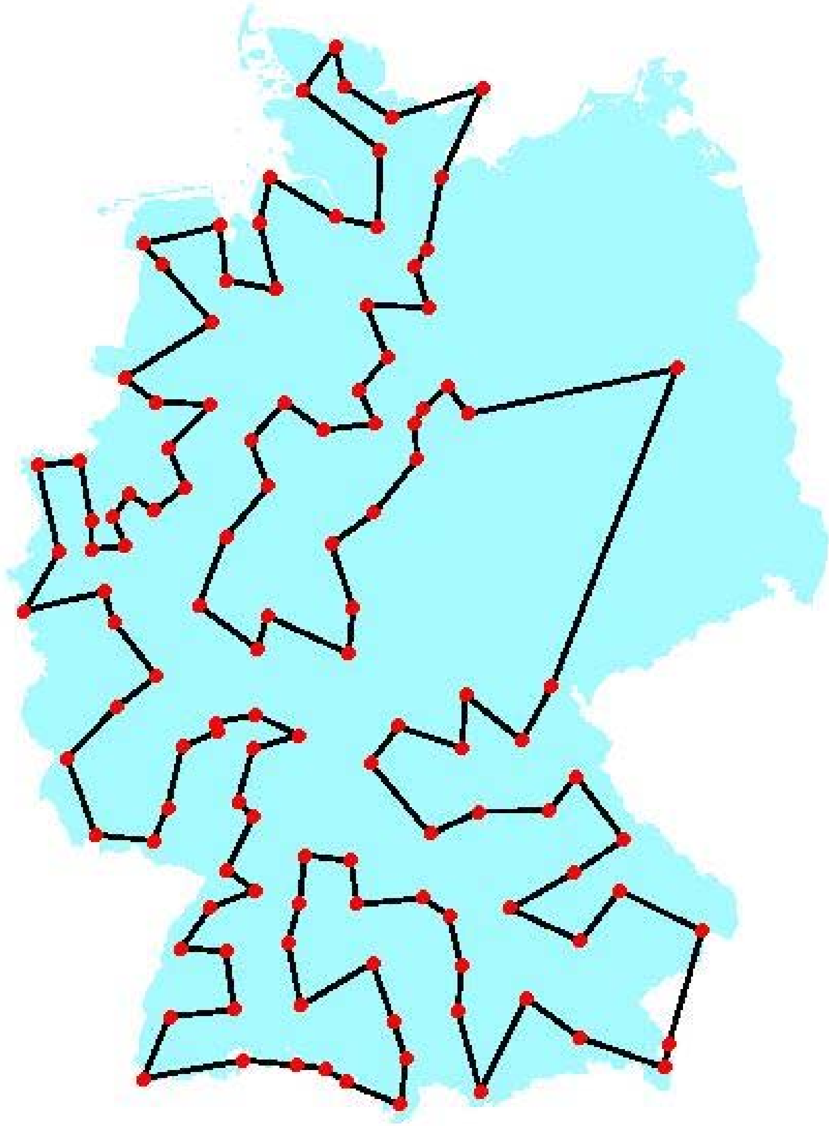
Domain Propagation



Conflict Analysis



West-Germany and Berlin



120 cities

7140 variables

1975

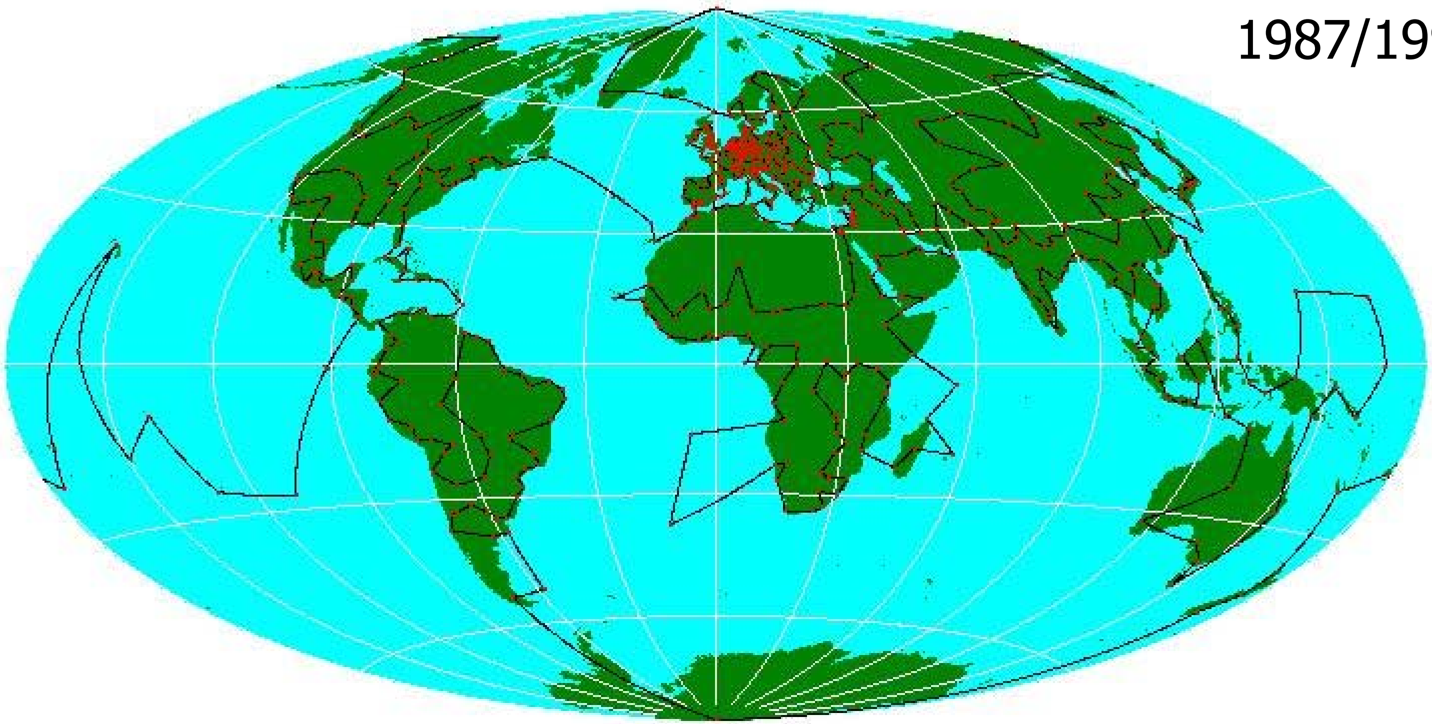
M. Grötschel

Martin Grötschel

A trip around the world

666 cities
221,445 variables

1987/1991



M. Grötschel, O. Holland

Some TSP World Records

2006
pla 85,900
solved

3,646,412,050
variables

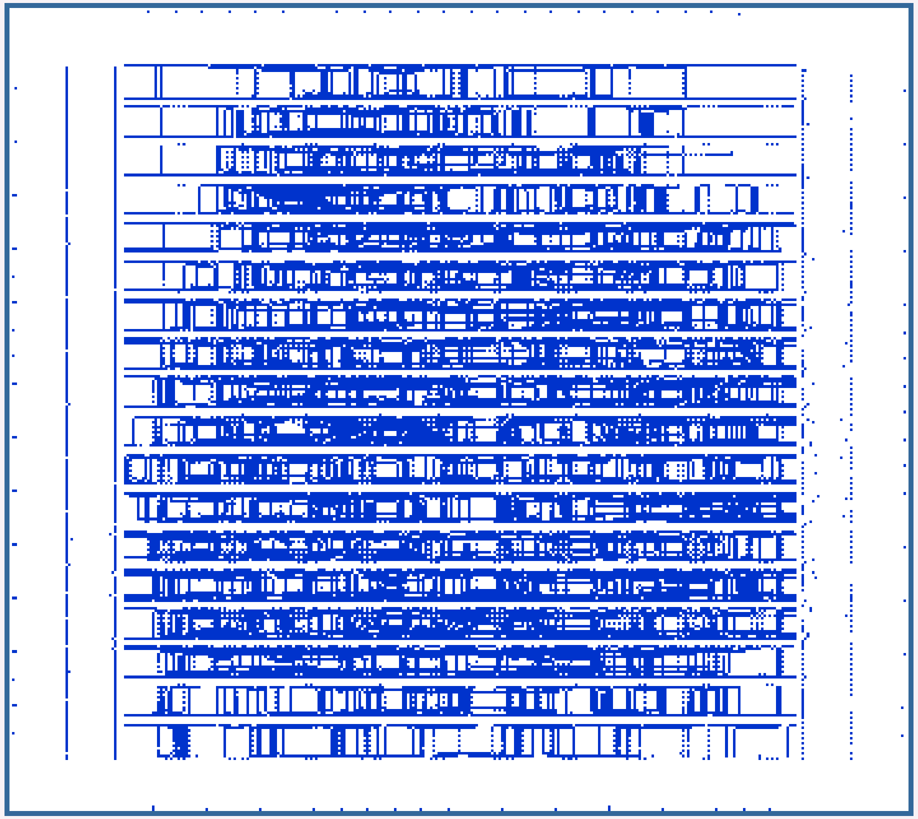
number of cities
2000x
increase

4,000,000
times
problem size
increase

in 52
years

year	authors	# cities	# variables
1954	DFJ	42/49	820/1146
1977	G	120	7140
1987	PR	532	141,246
1988	GH	666	221,445
1991	PR	2,392	2,859,636
1992	ABCC	3,038	4,613,203
1994	ABCC	7,397	27,354,106
1998	ABCC	13,509	91,239,786
2001	ABCC	15,112	114,178,716
2004	ABCC	24,978	311,937,753
2005	W. Cook, D. Epsinoza, M. Goycoolea	33,810	571,541,145

The current TSP world record



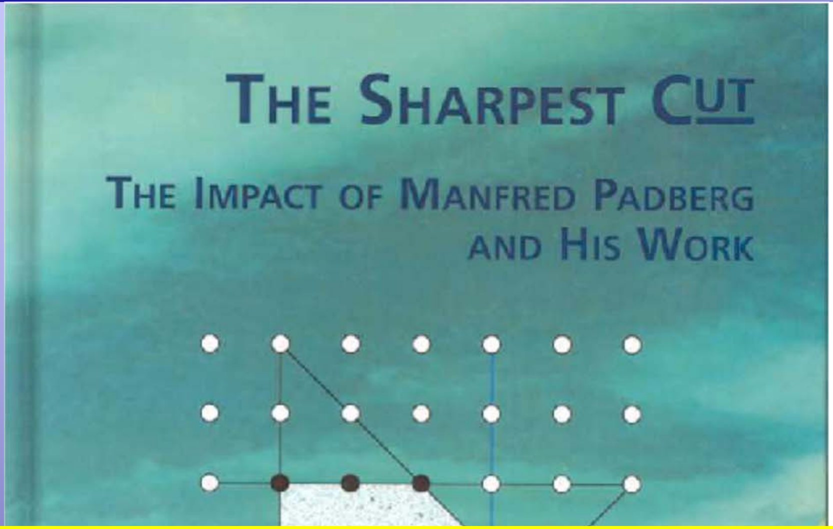
85,900 Locations in a VLSI Application
Solved in 2006

http://www.tsp.gatech.edu/optimal/index.html

http://www.tsp.gatech.edu/pla85900/index.html

Martin Grötschel

Information about computational Mixed Integer Programming, see, e.g.,



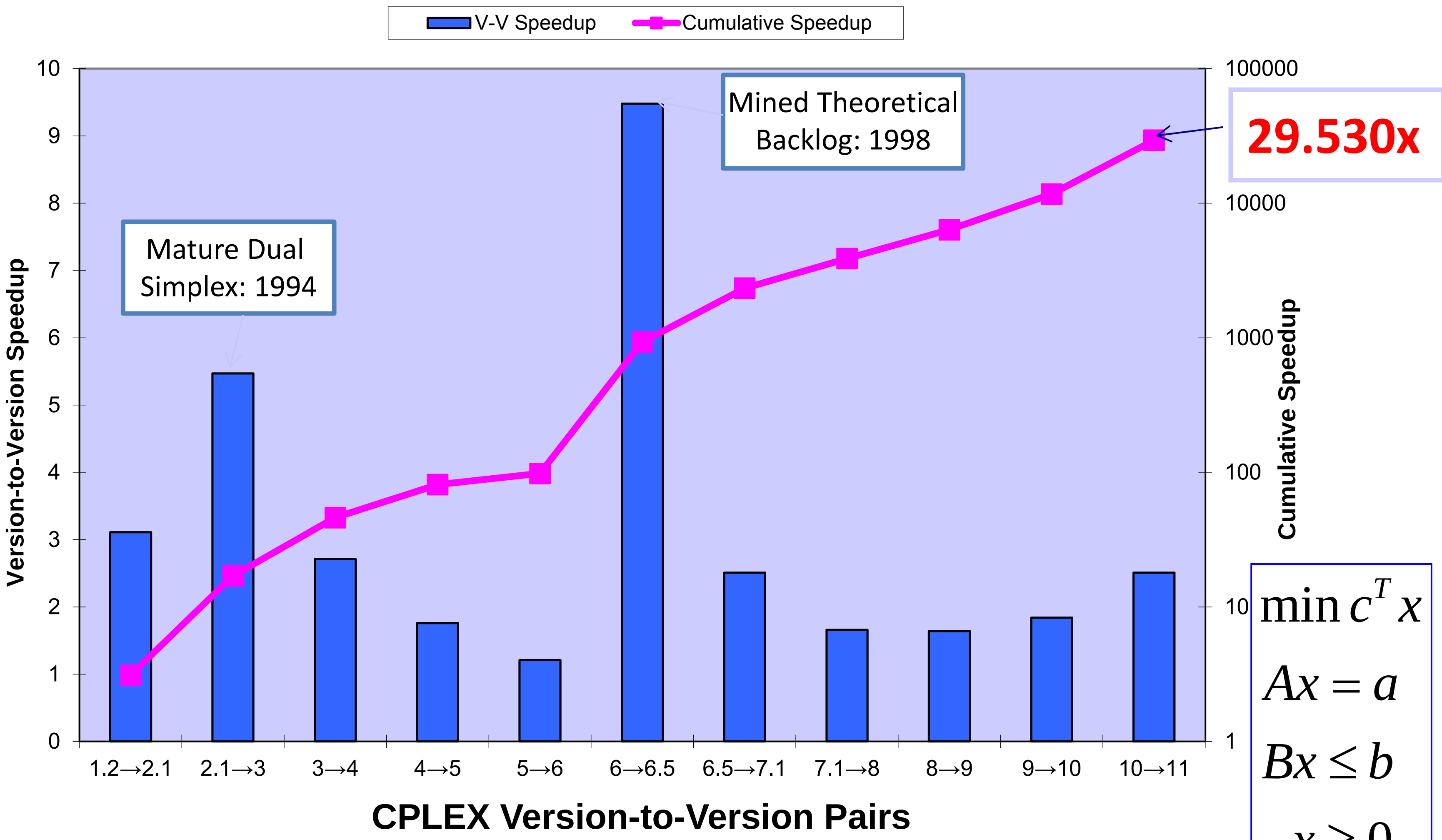
This book appeared in MPS-SIAM "Series on Optimization", 2004,
One particularly interesting article on MIP:

Mixed-Integer Programming: A Progress Report

Robert E. Bixby, Mary Fenelon, Zonghao Gu, Ed Rothberg, and Roland Wunderling **309**

Courtesy Bob Bixby

MIP Speedups 1991 – 2008



1½ years in 1991 ~ 1 second in 2008

$$\min c^T x$$
$$Ax = a$$
$$Bx \leq b$$
$$x \geq 0$$
$$x \in \mathbb{Z}^n$$

Gurobi MIP Solver

- New parallel design taking full advantage of multi-core architecture
- Leverage over existing cutting planes
  - Constraint aggregations: network cut
  - Different solutions: submip cut
- New heuristics and better designed and balanced for parallel
- MIP domination
  - Symmetry breaking: e.g. orbit branching
- A bag of new tricks
  - New presolve reductions

Slide of Zonghao Gu (Gurobi), 9/2010

MIP Performance

- Internal MIP set
- Gurobi V1.1 -> V2.0

Time	#Models	Speedup
> 1s	650	1.7x
> 10s	410	1.9x
>100s	210	2.2x
>1000s	59	3.9x

- Gurobi V2.0 -> V3.0

Time	#Models	Speedup
> 1s	794	1.6x
> 10s	521	2.0x
> 100s	295	2.9x
>1000s	144	6.7x

Slide of Zonghao Gu (Gurobi), 9/2010

Mittelmann MIP Benchmark

- 2.72X faster since 2007
  - CPLEX 11.1 (best at 2007)
  - Gurobi 1.1 is 1.6X faster than CPLEX 11.2 on P4
  - Gurobi 3.0 is 1.7X faster than Gurobi 1.1 on P4
  - CPLEX 12.2 is also much faster

Progress: MIP

Example 1: LP still can be HARD

SGM: Schedule Generation Model

157323 rows, 182812 columns, 6348437 nzs

- **LP relaxation at root node:**
  - **18 hours**
- **Branch-and-bound**
  - **1710 nodes, first feasible**
  - **3.7% gap**
  - **Time: 92 days!!**
- **MIP does not appear to be difficult: *LP is a roadblock*** **(but 1000x LP improvement would make "solvable" in 2 hours!)**

Example: bell3a (MIPLIB 3)

123 constraints 133 variables (39 binary, 32 integer)

Solution time line (3.0 GHz Xeon 5160):

- 1995 (CPLEX 3.0.7): 1.74 seconds
- 1996 (CPLEX 4.0.9): 1.13 seconds
- 1998 (CPLEX 6.0.1): 1.18 seconds
- 1999 (CPLEX 6.5.3): 3.11 seconds
- 2006 (CPLEX 10.0.1): 1.89 seconds
- 2007 (CPLEX 11.0.0): 1.42 seconds
- 2010 (CPLEX 12.2.0): 1.84 seconds

Mined theoretical backlog

Speedup: <1x

Example: magicsquare

89 constraints 552 variables (529 binary)

Solution time line (3.0 GHz Xeon 5160):

- 1995 (CPLEX 3.0.7): 13.51 seconds
- 1996 (CPLEX 4.0.9): 8.10 seconds
- 1998 (CPLEX 6.0.1): 0.03 seconds
- 1999 (CPLEX 6.5.3): 159.16 seconds
- 2006 (CPLEX 10.0.1): 19.7 minutes
- 2007 (CPLEX 11.0.0): 1.5 hours

Mined theoretical backlog

Dynamic search introd.

Slowdown: 179463x

A nasty example: bmf24mar

44 constraints   51 variables (51 integer)

Solution time line (3.0 GHz Xeon 5160):

- 1995 (CPLEX 3.0.7): cannot solve
- 1996 (CPLEX 4.0.9): cannot solve
- 1998 (CPLEX 6.0.1): cannot solve
- 1999 (CPLEX 6.5.3): cannot solve
- 2006 (CPLEX 10.0.1): cannot solve
- 2007 (CPLEX 11.0.0): cannot solve
- 2010 (CPLEX 12.2.0): cannot solve

Speedup: what is this?

Martin Grötschel

A three variables MIP that can't be solved by CPLEX

```
Welcome to CPLEX Interactive Optimizer 12.1.0
CPLEX is a registered trademark of IBM Corp.
CPLEX> read check/IP/Bugs/Kaibel/ggt3.lp
CPLEX> optimize
Presolve time =    0.00 sec.
MIP search method: dynamic search.
Parallel mode: none, using 1 thread.
Root relaxation solution time =    0.00 sec.

   Node     Left    Objective  IInf  Best Integer  Best Node       Gap

        0       0      1.0000     1                   1.0000
*       0+      0                          3.0000      1.0000    66.67%
        0       2      1.0000     1        3.0000      1.0000    66.67%
...
```

Contents

Martin Grötschel

http://zibopt.zib.de/

ZIB Optimization Suite

Konrad-Zuse-Zentrum für Informationstechnik Berlin
Division Scientific Computing
Department Optimization

The ZIB Optimization Suite is a tool for generating and solving mixed integer programs. It consists of the following parts

ZIMPL	a mixed integer programming modeling language
SoPlex	a linear programming solver
SCIP	a mixed integer programming solver and constraint programming framework.

The user can easily generate linear programs and mixed integer programs with the modeling language ZIMPL. The resulting model can directly be loaded into SCIP and solved. In the solution process SCIP may use SoPlex as underlying LP solver.

Since all three tools are available in source code and free for academic use, they are an ideal tool for academic research purposes and for teaching integer programming.

See ZIB licences for more information.

SCIP Design

MIP

- LP relaxation
- cutting planes

CP

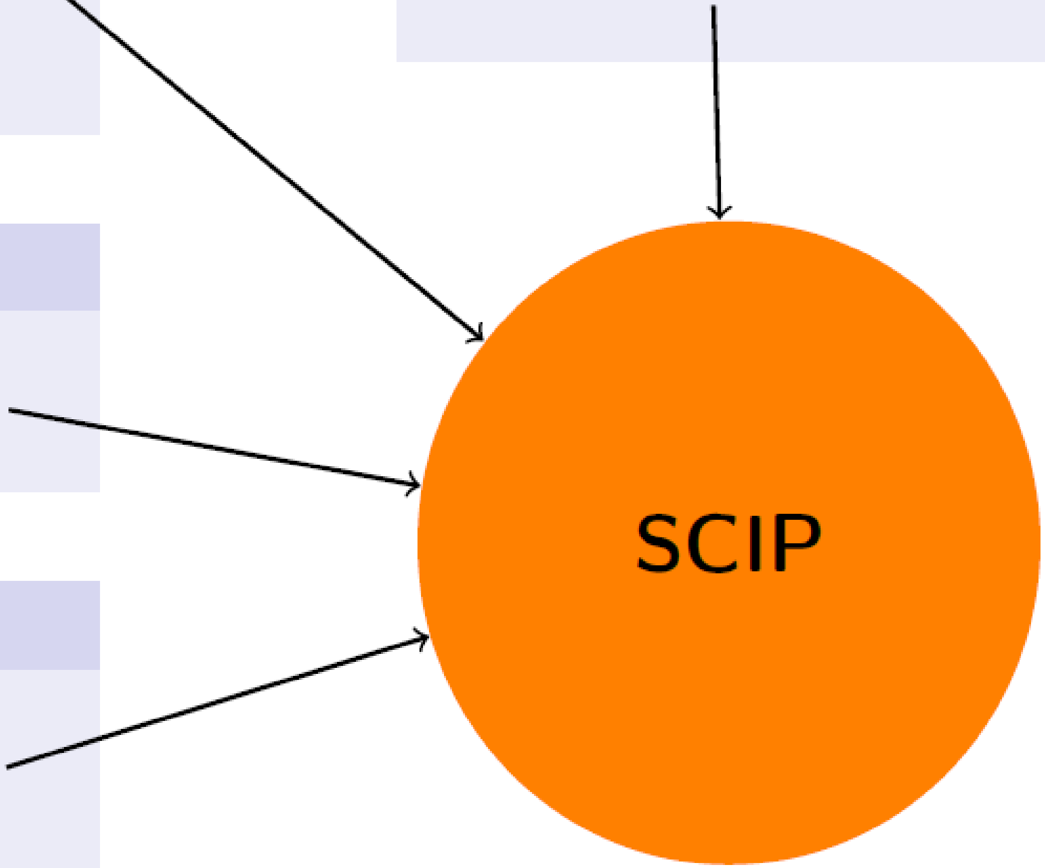
- domain propagation

SAT

- conflict analysis
- periodic restarts

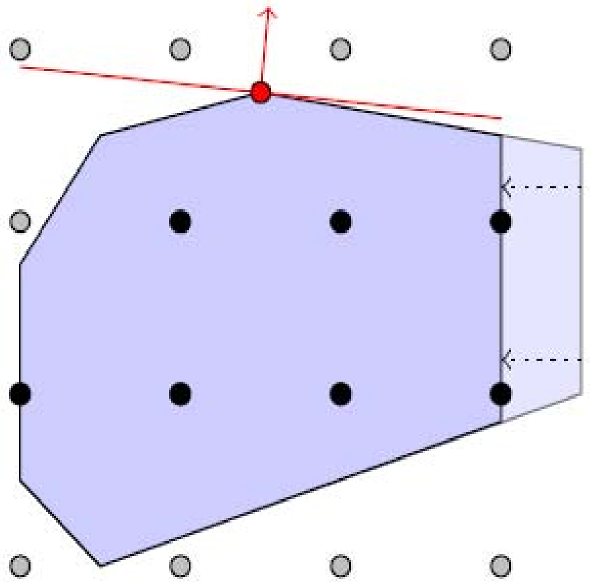
MIP, CP, and SAT

- branch-and-bound



Impact: -48 % solving time

SCIP Presolving



Task

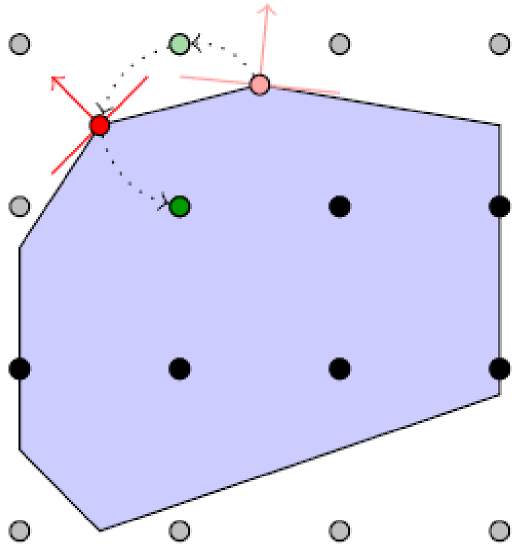
- ▷ Simplify model
- ▷ Strenghten formulation
- ▷ Extract information

Techniques

- ▷ Variables:
  - dual fixing
  - probing
  - bound strengthening
  - type changes
- ▷ Constraints:
  - coefficient tightening
  - dominance
  - upgrading
- ▷ Restarts:
  - abort search
  - reapply global presolving

Impact: -34% solving time

Primal Heuristics



Task

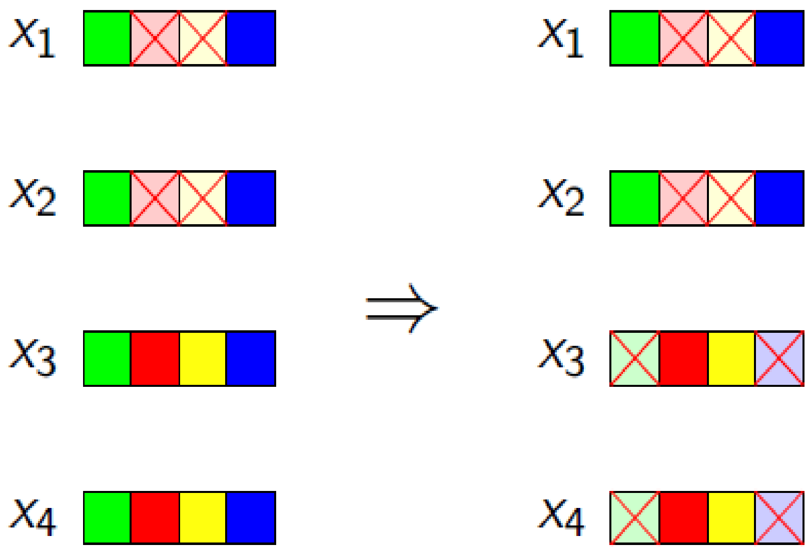
- ▷ Improve primal bound
- ▷ Incomplete methods
- ▷ Effective on average
- ▷ Guide remaining search

Techniques

- ▷ Rounding
  - Change fractional to integral values
- ▷ Diving
  - simulate DFS in the B&B tree using some special branching rule
- ▷ Objective diving
  - manipulate objective function (instead of bounds)
- ▷ Large Neighborhood Search
  - solve some sub-MIP
- ▷ Combinatorial
  - use special polyhedral properties

Impact: -20 % solving time

Domain Propagation



Task

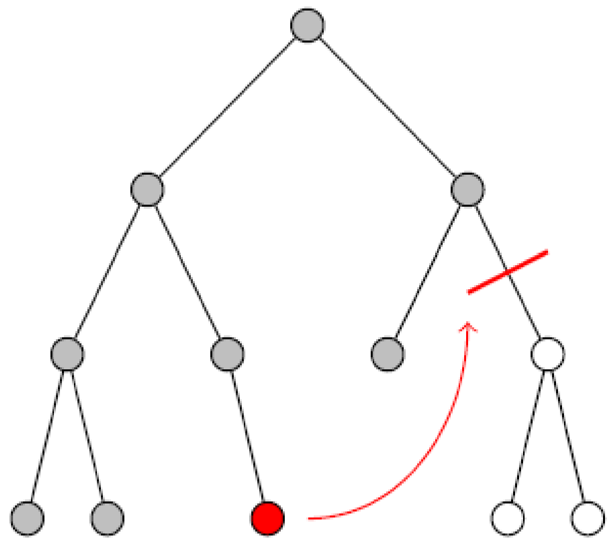
- ▷ Simplify model locally
- ▷ Improve local dual bound
- ▷ Detect infeasibility

Techniques

- ▷ Constraint specific
  - Each constraint handler may provide a propagation routine
  - Reduced presolving (usually)
- ▷ Dual propagation:
  - Root reduced cost strengthening
  - Objective function
- ▷ Reverse propagation:
  - Reconstruct a propagation
  - Necessary for conflict analysis

Impact: -12 % solving time

Conflict Analysis



Task

- ▷ Analyze infeasibility
- ▷ Derive valid constraints
- ▷ Help to prune other nodes

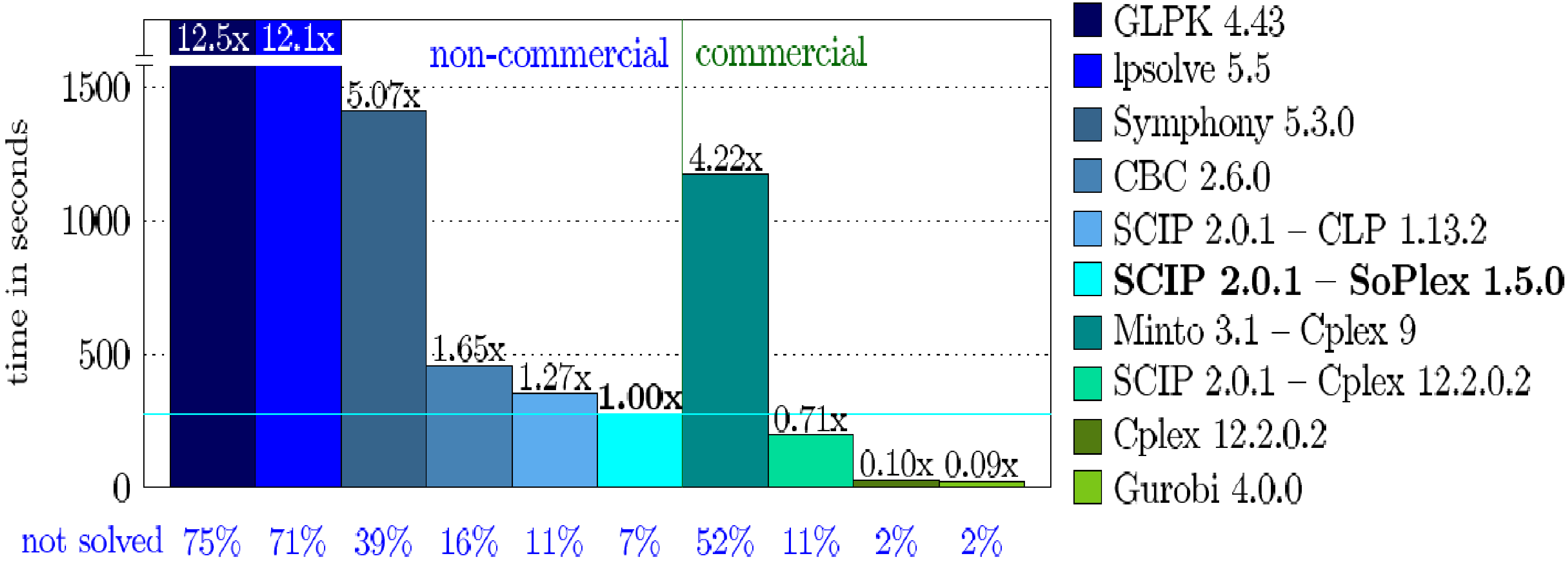
Techniques

- ▷ Analyze:
  - Propagation conflicts
  - Infeasible LPs
  - Bound-exceeding LPs
  - Strong branching conflicts
- ▷ Detection:
  - Cut in conflict graph
  - LP: Dual ray heuristic
- ▷ Use conflicts:
  - Only for propagation
  - As cutting planes

Martin Grötschel

SCIP: Solving Constraint Integer Programs

Newest Hans Mittelmann test, 55 instances



SCIP is currently one of the fastest non-commercial mixed integer programming solver. It is also a framework for Constraint Integer Programming and branch-cut-and-price.

SCIP allows total control of the solution process and the access of detailed information down to the guts of the solver.

Martin Grötschel

SCIP
Solving Constraint Integer Programs

SCIP is developed at ZIB in cooperation with

- TU Braunschweig, Institute for Mathematical Optimization
- University of Erlangen-Nürnberg, Chair of EDOM
- Siemens AG, Corporate Technology
- SAP
- Google (new supporter)

- Last release: September 30, 2010

The initial version of SCIP was developed in the PhD thesis of Tobias Achterberg (now IBM/CPLEX):

Martin Grötschel

Variability of SCIP solve runs: just permute rows and columns

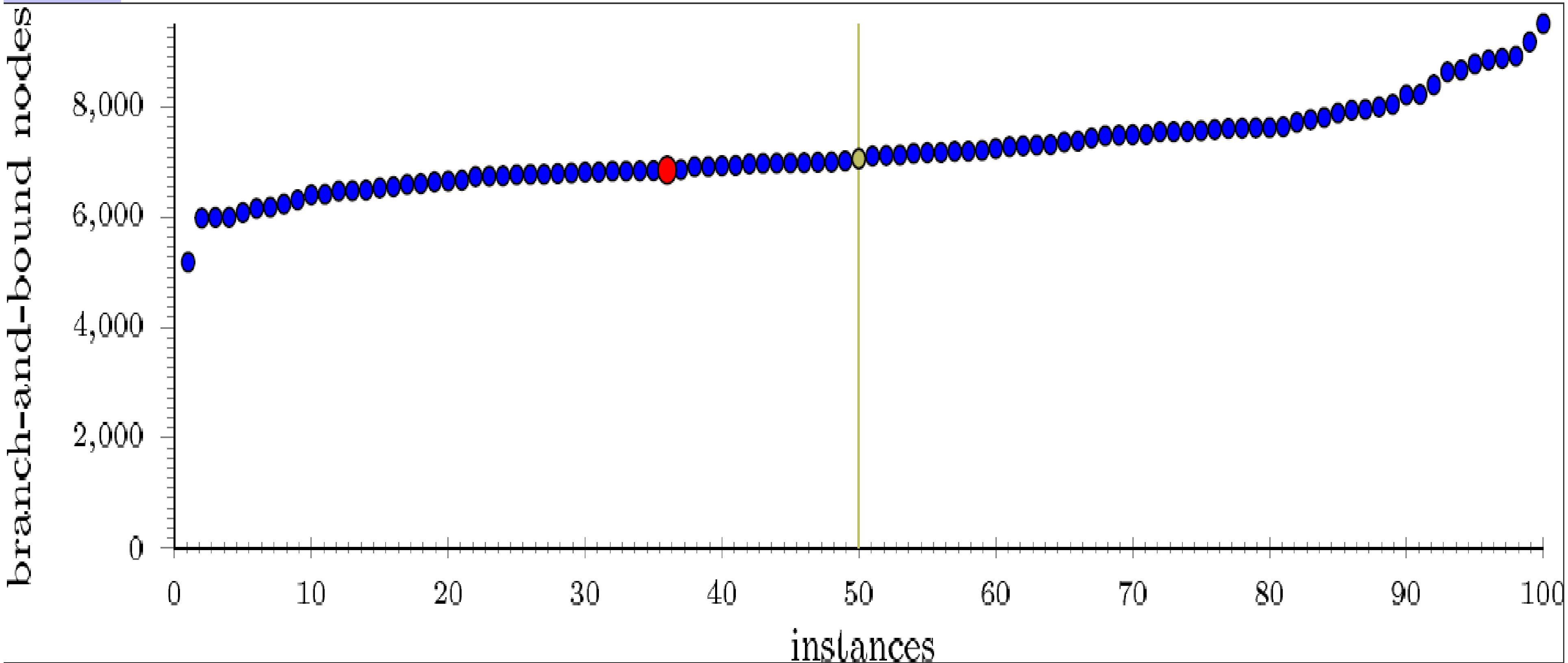
- Pick a “representative standard problem”.
- Permute rows and columns randomly 100 times.
- Run SCIP and analyze the runs.
- Graphical representation of number branch&bound nodes follows.

Running Time:

- We have seen examples where the longest running time was 60 times larger than the shortest one.

So, be careful when you claim that your code is faster than some other code!

Variability of SCIP solve runs, problem 1: just permute rows and columns



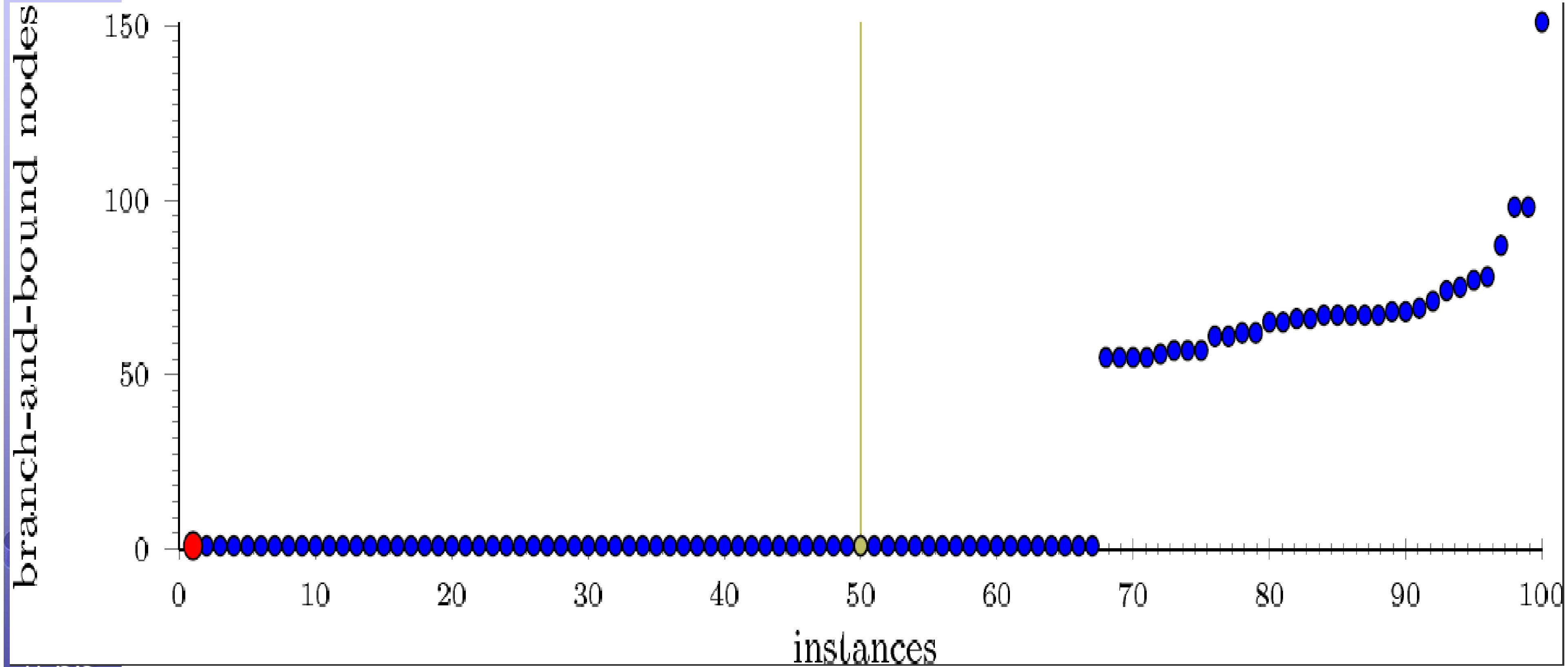
Ordered by increasing # of B&B nodes

green node: Median

red node: default variant

Martin Grötschel

Variability of SCIP solve runs, problem 2: just permute rows and columns



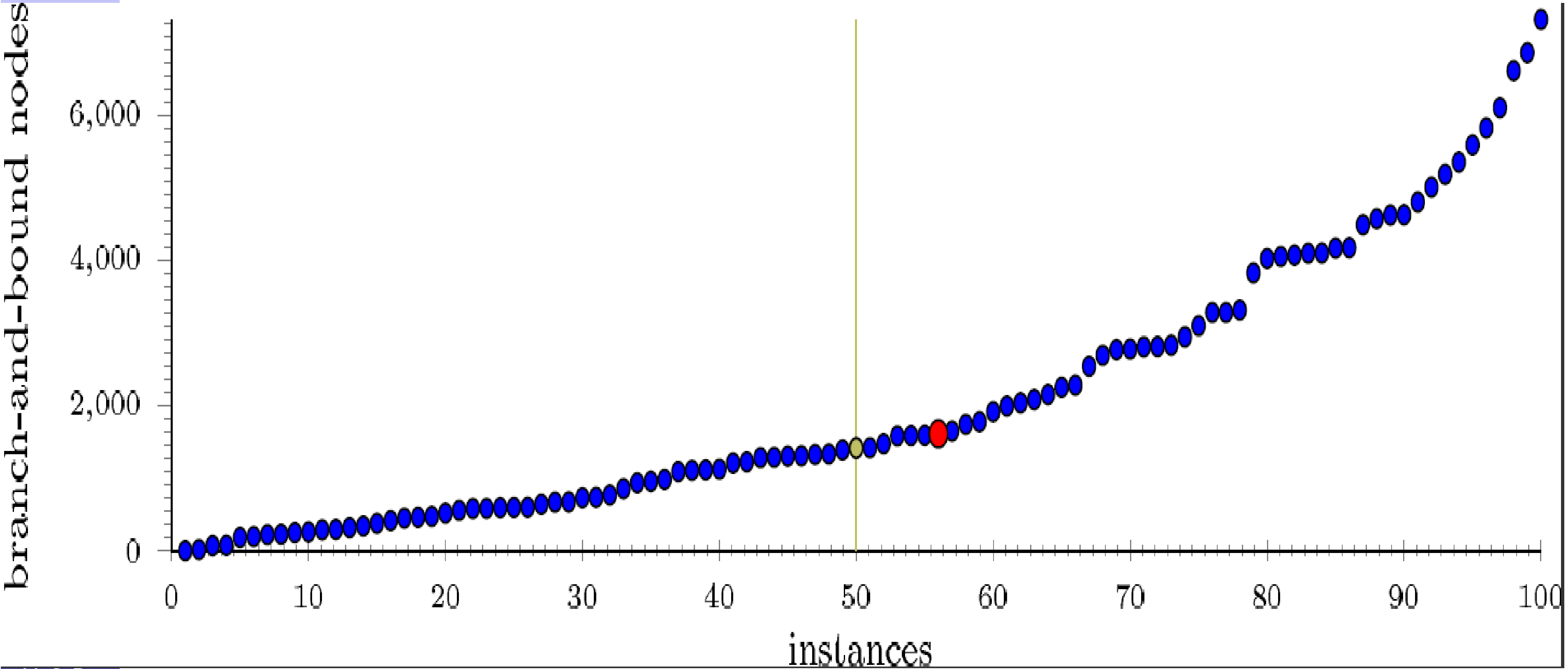
Ordered by increasing # of B&B nodes

green node: Median

red node: default variant

Martin Grötschel

Variability of SCIP solve runs, problem 3: just permute rows and columns



Ordered by increasing # of B&B nodes

green node: Median

red node: default variant

Martin Grötschel

Contents

Martin Grötschel

General Summary

- Tremendous improvements in the last 10-15 years.
- We are getting close to solving the really exciting and economically relevant cases.
- Industry, once a special problem has been solved, immediately asks for more.

Martin Grötschel

Mathematics of Infrastructure Planning (ADM III)

Thanks for your attention

ZIB, TU, and MATHEON, Berlin

Martin Grötschel
- Institut für Mathematik, Technische Universität Berlin (TUB)
- DFG-Forschungszentrum "Mathematik für Schlüsseltechnologien" (MATHEON)
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

groetschel@zib.de http://www.zib.de/groetschel