Mathematisches Forschungsinstitut Oberwolfach

Report No. 50/2005

Combinatorial Optimization

Organised by
Rainer E. Burkard (Graz)
David Shmoys (Ithaca)
Uwe Zimmermann (Braunschweig)

November 6th – November 12th, 2005

Abstract. This report summarizes the meeting on Combinatorial Optimization where new and promising developments in the field were discussed. The lectures show the many connections to other branches in Mathematics, like Combinatorics, Graph Theory, Geometry and Integer Programming. Furthermore, there are important connections to Theoretical Computer Science, Operations Research and an ever-growing number of application areas. Differing research directions in Combinatorial Optimization, current hot topics as well as classical streams were present in the talks. As can be seen from the table of contents, we strived for a balanced mixture of basic theoretical advancements and some selected practical applications.

Mathematics Subject Classification (2000): 90C27, 90C10, 90C11, 90C22, 90C06.

Introduction by the Organisers

For more than 30 years, meetings on Combinatorial Optimization have established a long and successful tradition at Oberwolfach. In fact, Combinatorial Optimization is a particularly active research area with links to many other areas in mathematics, e.g., to Combinatorics, Graph Theory, Geometry and Integer Programming. Furthermore, there are important connections to Theoretical Computer Science, Operations Research and many application areas. Therefore, it is not surprising that each of the meetings had its own format reflecting the most important recent developments within this scope and focusing on differing topics chosen by the respective organizers. In order to encourage such changes, a consensus emerged in the community that the organizers should vary from meeting to meeting.

Conceptually, we followed the outline of the last meeting in 2002, organized by Tom Liebling, Rolf Möhring and Uwe Zimmermann. Oberwolfach meetings are

planned a long time ahead and this one was no exception. In six focus talks, the state of the art in selected areas of high interest was demonstrated. For these presentations some internationally leading experts were approached about two months before the meeting. As was our intention, we put together a mix of senior scientists with broad experience as well as young scientists successfully pushing lines of core research with fresh ideas. In additional, we organized a series of sessions containing short talks not exceeding 25 minutes, which presented recent results. Here, we particularly encouraged young researchers to take advantage of this opportunity to present and discuss ongoing work with a broad international audience. Of course, the total number of talks had to be severely limited in order make room for the many vivid discussions within the sessions and between the sessions. We are very grateful to those who volunteered to pass on this opportunity to give a presentation. Their understanding was of great help in organizing the sessions. Furthermore, everybody had the opportunity to place a current research abstract at an appropriate message board in order to stimulate discussions.

The extended abstracts included in this report show that many differing research directions in Combinatorial Optimization, current hot topics as well as classical streams were present in the talks. In particular, the balanced mixture of basic theoretical advancements and some selected practical applications was very much welcomed by the participants.

At this point we wish to emphasize the special value which this meeting has for the Combinatorial Optimization community. The number of international conferences on Combinatorial Optimization and related topics is growing from year to year. Very attractive locations all around the world compete for such meetings. On the other hand, Oberwolfach offers the single and very well known opportunity to meet at a place optimally prepared for exchange of most recent results, for discussions of ongoing work and for joint work between the sessions. Combined with the friendly and relaxed atmosphere, these features are the basis for the success of the Combinatorial Optimization meetings at Oberwolfach. The long list of participants of this meeting and its predecessors reads as a guide to the international community of researchers in Combinatorial Optimization. Therefore, it is no surprise that many breakthrough results were initially spread at one of these meetings. Many participants arrive at the meeting well-prepared to discuss important steps in joint work with other participants. We are convinced that this meeting is one of the most important international meetings in Combinatorial Optimization.

In particular, for this particular meeting, we had a rather hard time to reduce the number of participants to the Oberwolfach meeting size. Moreover, there were surprisingly few rejections of the invitations. Nevertheless, with the help of Oberwolfach and supported by the European community, we succeeded in inviting a particularly large number of young researchers.

In our own view, and as expressed by many of the participants, the Oberwolfach workshop on "Combinatorial Optimization" was indeed a great success. This is mainly due to the excellent lectures prepared at a very high standard as well as to the many spontaneous questions and remarks discussed within the sessions. In

part, this may be seen from the included extended abstracts. We are very happy that we had the opportunity to organize this workshop at Oberwolfach. We think that Combinatorial Optimization as a central, lively research area should and will continue to be present at Oberwolfach.

Workshop: Combinatorial Optimization

Table of Contents

Abstracts

Exploiting symmetry in optimization

Alexander Schrijver

1. Introduction

We investigate how symmetry can be exploited to reduce the size of optimization problems, in particular, semidefinite programming problems. It results in new bounds for the maximum size of error-correcting codes ([3]) and for the crossing number of complete bipartite graphs (joint work with E. de Klerk and D.V. Pasechnik [2]).

Consider optimization problems of type

$$\max\{\mathrm{tr}(CX) \mid X \in \mathcal{F}\},$$

where C is some $N \times N$ matrix and $\mathcal{F}$ is a convex collection of $N \times N$ matrices.

Suppose that there is a group $\mathcal{G}$ of $N \times N$ permutation matrices such that $MCM^T = C$ and $M\mathcal{F}M^T = \mathcal{F}$ for each $M \in \mathcal{G}$. Then the maximum is attained by a matrix in

$$C_{\mathcal{G}} := \{X \in \mathbb{R}^{N \times N} \mid MXM^T = X \text{ for each } M \in \mathcal{G}\},$$

the *centralizer algebra* of $\mathcal{G}$.

Often, $\dim(C_{\mathcal{G}})$ is much smaller than N^2, and we might reduce the dimension of the search space. The set $C_{\mathcal{G}}$ is a matrix $*$-algebra, that is, it is closed under addition, scalar and matrix multiplication, and taking the conjugate transpose. We describe two reduction methods: block diagonalization and the regular representation. First we give two examples.

2. Coding

Given n and d, the number $A(n,d)$ is defined as the maximum number of words in $\{0,1\}^n$ at mutual Hamming distances $\geq d$. (The *Hamming distance* of two vectors is the number of coordinates in which they differ.)

Then the following holds: $A(n,d) \leq \max\{\mathrm{tr}(JX) \mid X \in \mathbb{R}_+^{\{0,1\}^n \times \{0,1\}^n}, X$ positive semidefinite, $\mathrm{tr}(X) = 1, X_{u,v} = 0$ if $0 < d(u,v) < d$ for all $u,v \in \{0,1\}^n$, $X_{u,v} = X_{u,u+v}$ for all $u,v \in \{0,1\}^n$, $\widetilde{X} \geq 0$, $\widetilde{X}$ PSD$\}$.

Here J denotes the all-1 matrix, and

$$\widetilde{X}_{u,v} := X_{u+v,u+v} - X_{u,v}.$$

This bound generalizes the well-known Delsarte linear programming bound, which is $\max\{\mathrm{tr}(JX) \mid X \in \mathbb{R}_+^{\{0,1\}^n \times \{0,1\}^n}, X$ positive semidefinite, $\mathrm{tr}(X) = 1, X_{u,v} = 0$ if $0 < d(u,v) < d$ for all $u,v \in \{0,1\}^n\}$.

Proof of the bound: If $C \subseteq \{0,1\}^n$ has minimum distance $\geq d$, define

$$X := |C|^{-2} \sum_{u \in C} \chi^{u+C} \left(\chi^{u+C}\right)^T,$$

where χ^C denotes the incidence vector of C in $\mathbb{R}^{\{0,1\}^n}$, considered as *column* vector. Then X is a feasible solution and $\mathrm{tr}(JX) = |C|$.

Group acting on the problem: For each $\pi \in S_n$, let M_π be the $\{0,1\}^n \times \{0,1\}^n$ permutation matrix corresponding to the permutation

$$(u_1, \ldots, u_n) \mapsto (u_{\pi(1)}, \ldots, u_{\pi(n)}).$$

for $(u_1, \ldots, u_n) \in \{0,1\}^n$. Let $\mathcal{G} := \{M_\pi \mid \pi \in S_n\}$. Then $\mathcal{G}$ satisfies the conditions.

3. Crossing number

Let $\mathrm{cr}(K_{m,n})$ be the *crossing number* of $K_{m,n}$ (the complete bipartite graph with colour classes of size m and n). *'Zarankiewicz conjecture'* asserts:

$$\mathrm{cr}(K_{m,n}) = \lfloor \tfrac{1}{4}(m-1)^2 \rfloor \lfloor \tfrac{1}{4}(n-1)^2 \rfloor.$$

Here $\leq$ follows by putting $\lfloor m/2 \rfloor$ points on the positive x-axis, $\lceil m/2 \rceil$ points on the negative x-axis, $\lfloor n/2 \rfloor$ points on the positive y-axis, $\lceil n/2 \rceil$ points on the negative y-axis, and next connecting each point on the x-axis with each point on the y-axis by a straight line segment.

The following upper bound was proved by de Klerk, Maharry, Pasechnik, Richter, and Salazar:

$$\mathrm{cr}(K_{m,n}) \geq \tfrac{1}{2} m^2 \alpha_n - \tfrac{1}{2} m \lfloor \tfrac{1}{4}(n-1)^2 \rfloor,$$

where

$$\alpha_n := \min\{\mathrm{tr}(CX) \mid X \in \mathbb{R}_+^{Z_n \times Z_n} \text{ positive semidefinite, } \mathrm{tr}(JX) = 1\}.$$

and where Z_n denotes the collection of cyclic permutations in S_n and where C denotes the matrix in $\mathbb{R}^{Z_n \times Z_n}$ with, for $\sigma, \tau \in Z_n$: $C_{\sigma,\tau} :=$ the minimum number of crossings of $K_{2,n}$ such that the edges leaving the two n-degree vertices in clockwise order go to $\sigma(1), \ldots, \sigma(n)$ and to $(\tau(1), \ldots, \tau(n)$ respectively. (Here we indicate the 2-degree vertices by $1, \ldots, n$.)

Proof of the bound: Let $K_{m,n}$ be embedded with a minimum number of crossing. For each $\sigma \in Z_n$, let d_σ be the number of n-degree vertices such that the edges leaving it go in clockwise order to $\sigma(1), \ldots, \sigma(n)$. Then $X := m^{-2} dd^T$ is feasible and

$$\mathrm{cr}(K_{m,n}) \geq \tfrac{1}{2} m^2 \mathrm{tr}(CX) - \tfrac{1}{2} m \lfloor \tfrac{1}{4}(n-1)^2 \rfloor.$$

Group acting on the problem: For each $\pi \in S_n$, let M_π be the $Z_n \times Z_n$ permutation matrix corresponding to the permutation

$$\sigma \mapsto \pi^{-1} \sigma \pi$$

for $\sigma \in Z_n$. Let

$$\mathcal{G} := \{M_\pi \mid \pi \in S_n\}.$$

4. Block diagonalization

Recall that a complex matrix U is called *unitary* if $U^*U = I$.

Theorem (Wedderburn): *Let $\mathcal{A}$ be a matrix $*$-algebra containing I. Then there exist a unitary matrix U and integers $p_1, q_1, \ldots, p_m, q_m > 0$ such that $U^*\mathcal{A}U$ is equal to the collection of* all *matrices*

$$\begin{pmatrix} B_1 & & & & & & \\ & B_1 & & & & & \\ & & B_2 & & & & \\ & & & B_2 & & & \\ & & & & & & \\ & & & & & B_m & \\ & & & & & & B_m \end{pmatrix}$$

where B_i is any *matrix in $\mathbb{C}^{p_i \times p_i}$, which is repeated q_i times.*

So

$$\dim(\mathcal{A}) = p_1^2 + \cdots + p_m^2$$

and

$$N = p_1 q_1 + \cdots + p_m q_m.$$

Moreover, $A \in \mathcal{A}$ is positive semidefinite $\iff$ each block B_i of U^*AU is PSD.

In the coding example, $C_{\mathcal{G}}$ is the *Terwilliger algebra:*

$$\mathcal{T}_n := \{\sum_{k,i,j} x_{k,i,j} E_{k,i,j} \mid x_{k,i,j} \in \mathbb{C}\},$$

where

$$(E_{k,i,j})_{u,v} := \begin{cases} 1 & \text{if } d(u,v) = k, |u| = i, |v| = j, \\ 0 & \text{else.} \end{cases}$$

Here $d(u,v)$ denotes the Hamming distance of u and v, and $|u|$ denotes the Hamming weight of u O(the number of nonzero entries).

It turns out that $\mathcal{T}_n$ has blocks $B_0, \ldots, B_{\lfloor \frac{1}{2}n \rfloor}$, such that block B_t of

$$U^* \left(\sum_{k,i,j} x_{k,i,j} E_{k,i,j} \right) U$$

is:

$$B_t := \left(\sum_k \gamma_{k,i,j}^{(t)} x_{k,i,j} \right)_{i,j=t}^{n-t}$$

where

$$\gamma_{k,i,j}^{(t)} := \sum_u (-1)^{u - \frac{1}{2}(i+j-k)} \tbinom{u}{\frac{1}{2}(i+j-k)} \tbinom{n-2t}{u-t} \tbinom{n-t-u}{i-u} \tbinom{n-t-u}{j-u}.$$

This leads to the following new upper bounds on $A(n,d)$:

n	d	best lower bound known	new upper bound	best upper bound previously known	Delsarte bound
19	6	1024	1280	1288	1289
23	6	8192	13766	13774	13775
25	6	16384	47998	48148	48148
19	8	128	142	144	145
20	8	256	274	279	290
25	8	4096	5477	5557	6474
27	8	8192	17768	17804	18189
28	8	16384	32151	32204	32206
22	10	64	87	88	95
25	10	192	503	549	551
26	10	384	886	989	1040

In a similar way, new bounds on constant-weight codes and on nonbinary codes were obtained.

5. The regular $*$-representation

For any group $\mathcal{G}$ of $N \times N$ permutation matrices, there exist nonzero $0,1$ matrices $E_1, \ldots, E_d$ such that the centralizer algebra satisfies

$$C_{\mathcal{G}} = \{\sum_{i=1}^{d} x_i E_i \mid x_1, \ldots, x_d \in \mathbb{C}\}$$

and such that

$$E_1 + \cdots + E_d = J \text{ (the all-one matrix).}$$

So $d = \dim(C_{\mathcal{G}})$. Suppose now that we can identify these matrices, and that we also can determine the *multiplication parameters*
$\mu_{i,j}^k$ (for $i,j,k = 1, \ldots, k$), defined by:

$$E_i E_j = \sum_{k=1}^{d} \mu_{i,j}^k E_k.$$

Then define, for $k = 1, \ldots, d$, the $d \times d$ matrix L_k by

$$(L_k)_{i,j} := \frac{\mathrm{tr}(E_i E_i^T)^{1/2}}{\mathrm{tr}(E_j E_j^T)^{1/2}} \mu_{k,j}^i$$

for $i,j = 1, \ldots, d$. Let

$$\mathcal{L} := \{\sum_{i=1}^{d} x_i L_i \mid x_1, \ldots, x_d \in \mathbb{R}\}.$$

Then $\mathcal{L}$ is a matrix $*$-algebra, and $\phi : C_{\mathcal{G}} \to \mathcal{L}$ defined by

$$\phi\left(\sum_i x_i E_i\right) := \sum_i x_i L_i$$

is an algebra $*$-isomorphism. That is, ϕ is a bijection satisfying $\phi(X+Y) = \phi(X) + \phi(Y)$, $\phi(\lambda X) = \lambda\phi(X)$, $\phi(XY) = \phi(X)\phi(Y)$, $\phi(X^*) = \phi(X)^*$ for all $X, Y \in C_{\mathcal{G}}$ and $\lambda \in \mathbb{C}$.

This implies:

$$\sum_i x_i E_i \text{ is positive semidefinite, } \iff \sum_i x_i L_i \text{ is positive semidefinite.}$$

It gives a reduction of the the order of the matrices in the matrix $*$-algebra to the dimension of the algebra. $\mathcal{L}$ corresponds to the *regular $*$-representation* of $C_{\mathcal{G}}$.

Application to the crossing number gives:

$$\alpha_9 = 7.7352126\ldots.$$

This implies, for each fixed $n \geq 9$:

$$\lim_{m\to\infty} \frac{\mathrm{cr}(K_{m,n})}{\lfloor \frac{1}{4}(m-1)^2 \rfloor \lfloor \frac{1}{4}(n-1)^2 \rfloor} \geq 0.8303.$$

References

[1] D.C. Gijswijt, A. Schrijver, H. Tanaka, New upper bounds for nonbinary codes, preprimt.

[2] E. de Klerk, D.V. Pasechnik, A. Schrijver, Reduction of symmetric semidefinite programs using the regular $*$-representation, Mathematical Programming, to appear.

[3] A. Schrijver, New code upper bounds from the Terwilliger algebra and semidefinite programming, IEEE Transactions on Information Theory 51 (2005) 2859–2866.

Optimizing the algebraic connectivity of a graph

Christoph Helmberg

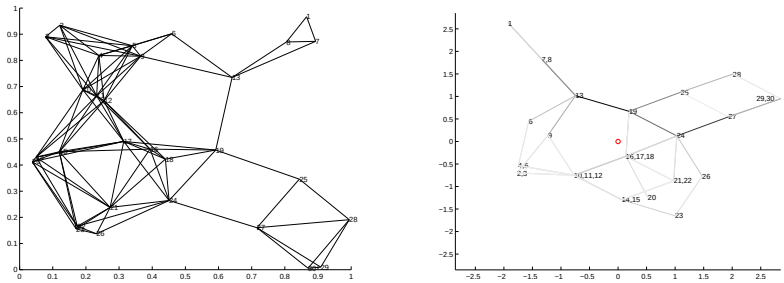
(joint work with Frank Göring, Markus Wappler)

Let $G = (N, E)$ be an undirected graph with node set $N = \{1, \ldots, n\}$ and edge set $E \subseteq \{ij : i, j \in N, i \neq j\}$. The *Laplace matrix* or *Laplacian* of the graph is the matrix $L = \mathrm{diag}(Ae) - A$, where A denotes the (possibly weighted) adjacency matrix, e the vector of all ones of appropriate dimension and $\mathrm{diag}(v)$ the diagonal matrix having v on its main diagonal. For symmetric matrices $H \in \mathbb{R}^{n\times n}$ we order the eigenvalues by $\lambda_1(H) \leq \lambda_2(H) \leq \cdots \leq \lambda_n(H)$. Because L is positive semidefinite and $Le = 0$, we have $\lambda_1(L) = 0$ with eigenvector e. Fiedler [1] showed that $\lambda_2(L)$ is tightly related to edge and vertex connectivity of the graph and called $\lambda_2(L)$ the *algebraic connectivity* of the graph. In particular, $\lambda_2(L)$ is positive if and only if G is connected. The problem we study here corresponds to the "absolute algebraic connectivity" of Fiedler [1] and reads: maximize λ_2 over all non negatively weighted adjacency matrices having total weight one. We formulate this problem as a semidefinite program, then take its dual and interpret

this, in the style of vector labellings via Gram representations, as the following embedding problem.

$$
(1) \qquad \begin{array}{ll} \max & \sum_{i\in N} \|v_i\|^2 \\ \text{s.t.} & (\sum_{i\in N} v_i)^2 = 0 \\ & \|v_i - v_j\|^2 \le 1 \qquad \text{for } ij \in E \\ & v_i \in \mathbb{R}^n \text{ for } i \in N. \end{array}
$$

Thus, the (dual) problem is equivalent to finding an embedding of the nodes of the graph in $n-$space so that their barycenter is at the origin (we will call this the *equilibrium constraint*), the distances of adjacent nodes are bounded by one, and the sum of their squared norms is maximized ([3] arrive at the same problem in the search of fastest mixing Markov chains and exhibit connections to a maximum variance unfolding problem). The figure below shows a random graph and an optimal embedding with optimal (primal) edge weights displayed in grey scale.



In 2-space the embedding corresponds physically to centripetal forces in equilibrium when rotating a net (connect the nodes by weightless chords of length one) around the common barycenter of the nodes. As the example suggests, the embedding is tightly linked to the separator structure of the graph. Our main results read (for proofs and further details see the full version [2])

Theorem 1 (Separator-Shadow)**.** *Let $v_i \in \mathbb{R}^n$ for $i \in N$ be an optimal solution of* (1) *for a connected graph $G = (N, E)$ and let $K_1 \dot\cup S \dot\cup K_2$ be a partition of N with no node in K_1 adjacent to a node in K_2. Then, for at least one $j \in \{1, 2\}$, for every $i \in K_j$ the straight line segment $[0, v_i]$ intersects the convex hull of the points in S.*

So if S is a separator and zero is not in the convex hull of its embedding, then all but one of the induced partitions are embedded in the shadow of this convex hull w.r.t. the origin. If zero is contained in its convex hull, the next result allows to find an optimal embedding whose dimension is either determined by a unique "heavy" part or bounded by the size of the separator plus one.

Theorem 2. *Let $v_i \in \mathbb{R}^n$ for $i \in N$ be an optimal solution of* (1) *for a connected graph $G = (N, E)$ and let $S \subset N$ with $0 \in \mathcal{S} = \operatorname{conv}\{v_s : s \in S\}$ be a separator in G inducing a partition $(S, K_1, \dots, K_m)$ of N so that no node in K_j is adjacent to a node in K_h for $j \ne h$, $j, h \in M = \{1, \dots, m\}$. Set $\mathcal{L} = \operatorname{span} \mathcal{S}$ and, for $j \in M$, $\beta_j = \sum_{i\in K_j} \|p_{\mathcal{L}^\perp}(v_i)\|$.*

(i) If $\beta_{\hat\jmath} > \sum_{j\in M\setminus\{\hat\jmath\}} \beta_j$ for one $\hat\jmath \in M$ then there exist $h \in \mathcal{L}^\perp$ and an optimal embedding $v'_i \in \mathbb{R}^n$ of (1) *with $v'_i = v_i$ for $i \in S$, $v'_i \in \mathcal{L} + \operatorname{span}\{h, v_i : i \in$*

$K_{\hat{\jmath}}\}$ *for* $i \in K_{\hat{\jmath}}$ *and* $v'_i \in \mathcal{L} + \{\beta \sum_{i \in K_{\hat{\jmath}}} v'_i : \beta \geq 0\}$ *for* $i \in \bigcup_{j \in M \setminus \{\hat{\jmath}\}} K_j$. *If, in addition, there exists* $\bar{b} \in \operatorname{span}\{v_i : i \in K_{\hat{\jmath}}\}$, $\|\bar{b}\| = 1$ *so that* $\langle \bar{b}, v_i \rangle \geq 0$ *for all* $i \in K_{\hat{\jmath}}$, *then such an embedding exists with* $h = 0$.

(ii) *If* $\beta_{\hat{\jmath}} \leq \sum_{j \in M \setminus \{\hat{\jmath}\}} \beta_j$ *for all* $\hat{\jmath} \in M$ *then there exist vectors* $d_1, d_2, d_3 \in \mathcal{L}^\perp$, $\|d_1\| = \|d_2\| = \|d_3\| = 1$ *with* $\dim \operatorname{span}\{d_1, d_2, d_3\} \leq 2$, $b_j \in \{d_1, d_2, d_3\}$, $j \in M$, *and an optimal embedding* $v'_i \in \mathbb{R}^n$, $i \in N$, *of* (1) *with* $v'_i = v_i$ *for* $i \in S$ *so that for each* $j \in M$ *we have* $v'_i \in \mathcal{L} + \{\beta b_j : \beta \geq 0\}$ *for all* $i \in K_j$. *One may assume* $b_j = d_1$ *for at most one* $j \in M$.

(iii) *If, in case (ii), the index* $\hat{\jmath} \in M$ *is the only* $j \in M$ *satisfying* $b_j = d_1$ *and at most* $|S| - 1$ *nodes of* S *are adjacent to nodes in* $K_{\hat{\jmath}}$, *then there is an optimal embedding of dimension at most* $|S|$.

This leads to a bound on the dimension of an optimal embedding of minimal dimension.

Theorem 3. *Let* $G = (N, E)$ *be a connected graph, then there exists an optimal embedding for* (1) *of dimension at most tree-width of* G *plus one.*

In fact, the proof identifies a certain "central" node in any tree decomposition and its size gives rise to the bound. Even though we exhibit examples where this bound is tight, the bound is often too pessimistic. In particular, we have not yet been able to construct a planar graph whose optimal embedding of minimal dimension exceeds dimension three. A slightly different formulation leads to a graph parameter that is closed under taking minors. The latter resembles other parameters in the style of the Colin de Verdière graph parameter (see the survey [4]); these connections need to be explored.

References

[1] M. Fiedler. Laplacian of graphs and algebraic connectivity. *Combinatorics and Graph Theory*, 25:57–70, 1989.

[2] F. Göring, C. Helmberg, and M. Wappler. Embedded in the shadow of the separator. Preprint 2005-12, Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany, Sept. 2005.

[3] J. Sun, S. Boyd, L. Xiao, and P. Diaconis. The fastest mixing markov process on a graph and a connection to a maximum variance unfolding problem. Technical report, Electrical Engineering Department, Stanford University, May 2004. Submitted to SIAM Review.

[4] H. van der Holst, L. Lovász, and A. Schrijver. The Colin de Verdière graph parameter. In L. et al., editor, *Graph theory and combinatorial biology*, volume 7 of *Bolyai Soc. Math. Stud.*, pages 29–85. János Bolyai Mathematical Society, Budapest, 1999.

An improved algorithm for computing Steiner minimal trees in $\Re^d$

Kurt M. Anstreicher
(joint work with Marcia Fampa)

The Euclidean Steiner tree problem (ESTP) in $\Re^d$ is defined as follows: Given a set of p points in $\Re^d$, find a tree of minimal Euclidean length that spans these points but that can also utilize additional points in the construction of the tree. We refer to the original points as *terminal nodes*, and any additional nodes in the spanning tree as *Steiner points*. The ESTP has been shown to be NP-Hard [3] and has received considerable attention in the literature. For a comprehensive survey see [5]. The solution of an ESTP is called a Steiner minimal tree (SMT). The *topology* of a Steiner tree corresponds to choosing the number of Steiner points and the edges between points, but not the geometric position of the Steiner points. A topology is called a *Steiner topology* if each Steiner point has degree equal to three and each terminal node has degree three or less. A Steiner topology with p terminal nodes is a *full Steiner topology* if there are $p-2$ Steiner points and each terminal node has degree equal to one. A Steiner tree which corresponds to some topology, but with certain edges shrunk to zero length, is said to be degenerate. Any SMT with a nonfull Steiner topology can also be associated with a full Steiner topology for which the tree is degenerate.

A number of papers have considered the exact solution of the ESTP in $\Re^2$. Melzak [6] was the first to present an algorithm to solve the problem, which was based on the enumeration of all Steiner topologies and on the determination of the length of the SMT corresponding to each topology. At the present time the best exact solution algorithm for the ESTP in the plane, the GeoSteiner algorithm [9], can handle typical problem instances with thousands of terminals. Methods specialized for $\Re^2$ cannot be applied to problems in higher dimensions, however, and very few papers have considered exact methods for $d \geq 3$. A polynomial time approximation scheme (PTAS) is known for the ESTP in $\Re^d$, see [2].

Gilbert and Pollak [4] proposed computing SMTs in $\Re^d$ by enumerating all Steiner topologies and computing the minimal length for the tree associated with each topology. Unfortunately the number of Steiner topologies having p terminal nodes grows extremely fast in p, so the enumeration of all topologies is only possible for very small values of p. Smith [7] proposes an implicit enumeration scheme for all full Steiner topologies on a given set of p terminals. The root node in the resulting Branch and Bound (B&B) tree corresponds to the unique full Steiner topology for three given terminals, and the nodes at depth k in the B&B tree enumerate all full Steiner topologies having $k+3$ terminal nodes. Branching is accomplished by adding a new terminal node and creating children whose topologies are obtained by "merging" a given edge in the tree with the new terminal node to create a new full Steiner topology. The merge operation is illustrated in Figure 1. In the figure a new terminal node c is merged with an edge l in a tree T having n edges to produce a new tree T^+ having $n+2$ edges and an additional Steiner node.

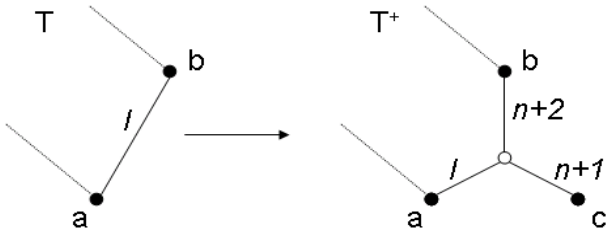


FIGURE 1. Generating tree T^+ from T

It is easily shown that the merging operation cannot *decrease* the minimal length of the tree, and therefore if the minimal length of a Steiner tree at some node in the B&B tree is *longer* than that of a known Steiner tree on all p terminals the given node may be "fathomed" and its descendents removed from further consideration. If a node cannot be fathomed, a new terminal is selected to be added to the given topology and one child node is created for each one of its edges using the merge operation. Computational results in [7] obtain SMTs for the vertices of regular d-polytopes with 16 or fewer vertices. Results for the simplex and octahedron in dimensions $3 \leq d \leq 9$ are sufficient to disprove a conjecture of Gilbert and Pollak on the "Steiner ratio" in these dimensions. (The Steiner ratio in $\Re^d$ is the minimal possible ratio between the length of a SMT for a given set of terminals in $\Re^d$ and the length of a minimal spanning tree on the same terminals which is not permitted to usa any Steiner points.) Additional computations [8] support a new conjecture on the Steiner ratio in $\Re^3$.

In this work we consider improvements to Smith's B&B algorithm for computing Steiner minimal trees in $\Re^d$. We use a conic formulation for the problem of computing the SMT for a given topology. This formulation permits use of high-performance software (MOSEK [1]) that obtains a rigorous lower bound on the minimal length of a SMT, as opposed to the putatively near-optimal values used by Smith. In addition we use a procedure based on fixing dual variables to estimate the effect of the merge operation that produces the children of a given node in the B&B tree. This allows us to eliminate some children without actually computing the SMTs associated with their topologies and also permits the implementation of a "strong branching" strategy. In the context of Smith's enumeration scheme the branching decision at a node in the tree corresponds to choosing the next terminal node to add. Using our dual-based estimates we choose the next terminal so as to minimize the number of children produced, and in the case of a tie to maximize the sum of the bounds for the child nodes.

In numerical experiments our improvements result in substantial computational gains compared to Smith's original algorithm. We have solved a variety of different problems using the original and improved algorithms. Both algorithms were coded in C and all runs were performed on a 1.8 GHz Pentium CPU running Linux. To investigate the effect of d we created problems with $p = 10$ terminals randomly

	Unfathomed nodes		CPU time (seconds)	
d	Smith	Smith+	Smith	Smith+
2	5,617.0	55.0	717.8	68.4
3	31,854.4	719.6	2,334.1	753.5
4	296,928.4	5,134.4	16,153.0	5,735.6
5	399,031.8	2,741.0	20,805.6	4,680.5

TABLE 1. Average performance on instances with 10 terminals

distributed in the unit hypercube in $\Re^d$, $d = 2, 3, 4, 5$. Average computational results on these problems are reported in Table 1. In the table we give the average number of unfathomed nodes in the B&B tree, and the average CPU time, for five instances with each d using both Smith's original algorithm and our improved version (Smith+). The results on these problems show that while problems with a fixed p are generally more difficult as d increases, our modifications of Smith's algorithm can substantially reduce the computational effort to solve problems in $\Re^d$ to optimality.

REFERENCES

[1] E.D. Andersen and K.D. Andersen, *The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm*, In *High Performance Optimization*, H. Frenk, C. Roos, T. Terlaky and S. Zhang, eds., Applied Optimization **33**, Kluwer Academic Publishers, 2000, 197-232.

[2] S. Arora, *Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems*, J. ACM **45** (1998), 753-782.

[3] M.R. Garey, R.L. Graham and D.S. Johnson, *The complexity of computing Steiner minimal trees*, SIAM Journal on Applied Mathematics **32** (1977), 835-859.

[4] E.N. Gilbert and H.O. Pollack, *Steiner minimal trees*, SIAM J. Applied Math. **16** (1968), 1-29.

[5] F.K. Hwang, D.S. Richards and W. Winter, *The Steiner tree problem*, Annals of Discrete Mathematics **53**, North-Holland, 1992.

[6] Z.A. Melzak, *On the problem of Steiner*, Canadian Mathematical Bulletin **4** (1961), 143-148.

[7] W.D. Smith, *How to find Steiner minimal trees in Euclidean d-space*, Algorithmica **7** (1992), 137-177.

[8] W.D. Smith and J.M. Smith, *On the Steiner ratio in 3-space*, Working paper, Department of IEOR, University of Massachusetts, 1994.

[9] D.M. Warme, P. Winter and M. Zachariasen, *Exact algorithms for plane Steiner tree problems: a computational study*, In *Advances in Steiner Trees*, D.Z. Du, J.M. Smith and J.H. Rubinstein, eds., Kluwer Academic Publishers, 2000, 81-116.

Semidefinite versus copositive relaxations for some combinatorial optimization problems

Franz Rendl
(joint work with I. Dukanovic, J. Povh)

1. Stable sets

Several quadratic problems in 0-1 variables lead in a natural way to relaxations involving semidefinite matrices. The stable set problem is one of the most prominent examples:

$$\alpha(G) := \max e^T x \text{ such that } x_i x_j = 0 \ [ij] \in E(G), \ x_i \in \{0,1\}.$$

Here $e^T = (1,\ldots,1)$, $J = ee^T$ and $\alpha(G)$ is the stability number of the graph G with nodes $V(G) = \{1,\ldots,n\}$ and edge set $E(G)$. If $x \in \{0,1\}^n$ we consider

$$X := \frac{1}{x^T x} x x^T. \tag{1}$$

Hence $x_{ij} = 0$ if $[ij] \in E(G)$ and $e^T X e = e^T x$. It is not hard to see that

$$\alpha(G) = \max\langle J, X\rangle \text{ such that } A_G(X) = 0, \ \mathrm{tr}(X) = 1, \ X \succeq 0, \ rank(X) = 1.$$

The vector $A_G(X)$ is defined as $(A_G(X))_{ij} = x_{ij} + x_{ji}$ for $[ij] \in E(G)$. Leaving out the rank constraint, we arrive at the semidefinite programming (SDP) relaxation introduced by Lovász [4].

$$\alpha(G) \le \theta(G) := \max\langle J, X\rangle \text{ such that } A_G(X) = 0, \ \mathrm{tr}(X) = 1, \ X \succeq 0.$$

Instead of requiring $X \succeq 0$, we could alternatively have asked that X lies in the cone $C^* := \{\sum y_i y_i^T : y_i \ge 0\}$ of completely positive matrices, as can be seen from (1). DeKlerk and Pasechnik [3] in fact show:

$$\alpha(G) = \max\langle J, X\rangle \text{ such that } A_G(X) = 0, \ \mathrm{tr}(X) = 1, \ X \in C^*.$$

Unfortunately, it is NP-hard to optimize over C^*. This suggests however to investigate tractable cones K which give better approximations to C^* than P, the cone of semidefinite matrices:

$$C^* \subseteq K \subseteq P.$$

Before addressing this issue, we consider some more cases similar to the stability number.

2. Quadratic assignment problem

We denote by Π the set of permutation matrices. The quadratic assignment problem (QAP) can be defined as follows:

$$z_{QAP} := \min \mathrm{tr} AXBX^T \text{ such that } X \in \Pi.$$

Using $x = vec(X)$ and $L = B \otimes A$, we can reformulate the problem as follows: If $X = (x_1,\ldots,x_n)$, we consider

$$Y = xx^T.$$

We partition the $n^2 \times n^2$ matrix Y into $n \times n$ blocks Y^{ij} by $Y^{ij} := x_i x_j^T$. If $X \in \Pi$ then $\sum_i Y^{ii} = \sum_i x_i x_i^T = I$ and $\mathrm{tr}(Y^{ij}) = x_i^T x_j = \delta_{ij}$. Finally, $\langle J, Y^{ij} \rangle = (e^T x_i)(e^T x_j) = 1$. Let us define

$$F := \{Y \in C^* : \sum_i Y^{ii} = I,\ \mathrm{tr}(Y^{ij}) = \delta_{ij},\ \langle J, Y^{ij} \rangle = 1\}.$$

Povh shows the following result.

Theorem 1. [6]

$$F = conv\{xx^T : x = vex(X),\ X \in \Pi\}$$

Hence the copositive problem $\min\{\mathrm{tr}LY :\ Y \in F\}$ also solves QAP.

3. Graph coloring

Using duality, we can rewrite $\theta(G)$ as

$$\theta(G) = \min t \text{ such that } tI + A_G^T(y) - J \succeq 0.$$

The adjoint operator A_G^T of A_G is given by $A_G^T(y) = \sum_{[ij] \in E} y_{ij} E_{ij}$, with $E_{ij} = e_i e_j^T + e_j e_i^T$. Here $I = (e_1, \ldots, e_n)$. Lovász [4] observed that $\theta(G)$ is also a lower bound on the chromatic number $\chi(\bar{G})$ of $\bar{G}$, the complement graph of G.

A t-coloring of $\bar{G}$ can be viewed as a partition of V into t stable sets S_i in $\bar{G}$. We denote the characteristic vectors of S_i by χ_i and call

$$M = \sum_{i=1}^{t} \chi_i \chi_i^T$$

a coloring matrix. Coloring matrices M of $\bar{G}$ can be characterized as follows.

$$M \text{ is t-coloring matrix of } \bar{G} \iff$$

$$m_{ij} \in \{0,1\},\ diag(M) = e,\ m_{ij} = 0\ [ij] \in \bar{E},\ \tau M - J \succeq 0 \iff \tau \geq t.$$

Lovász ignores $m_{ij} \in \{0,1\}$ and concludes

$$\chi(\bar{G}) \geq \min \tau \text{ such that } \tau I + A_G(y) - J \succeq 0 = \theta(G).$$

Since coloring matrices are also in C^*, we can tighten this relaxation by considering

$$\chi(\bar{G}) \geq t^* = \min \tau \text{ such that } \tau I + A_G^T(y) \in C^*,\ \tau I + A_G^T - J \succeq 0.$$

Dukanovic [1], see also [2], shows that t^* is in fact bounded from above by the fractional chromatic number.

Theorem 2. *$t^* \leq \chi_f(\bar{G})$ with equality holding if G is vertex transitive.*

4. Approximations of C^*

In the previous examples we saw that copositive relaxations sometimes are superior to SDP relaxations. To overcome the intractability of copositive programming, Parrilo [5] introduced a family of SDP approximations of C^* which comes arbitrarily close to C^*. The basic idea goes as follows. For a symmetric matrix M, let $p_M(x) := \sum_{ij} x_i^2 x_j^2 m_{ij}$ and $q_{M,r} := (\sum_k x_k^2)^r p_M(x)$. Then

$$M \in C^* \iff p_M(x) \geq 0 \ \forall x \iff q_{M,r}(x) \geq 0 \ \forall x.$$

A sufficient condition for $q_{M,r}(x) \geq 0$ to hold $\forall x$ is that q can be written as a sum of squares (SOS) of polynomials in x. Parrilo shows that $q_{M,0}$ is SOS $\iff$ $M \in P + N$, where N is the cone of elementwise nonnegative matrices. Parrilo also shows that $q_{M,1}$ is SOS if and only if there exist n symmetric matrices M^i such that

$$M - M^i \succeq 0,\ m^i_{ii} = 0 \forall i,\ m^j_{ii} + 2m^i_{ij} = 0 \ \forall i \neq j,\ m^i_{jk} + m^j_{ik} + m^k_{ij} \geq 0 \forall i < j < k.$$

This certificate is computationally rather expensive, as it involves n semidefiniteness constraints on matrices of order n and $O(n^3)$ inequality constraints. Dukanovic [1] suggests the following simplification in the case that M has the symmetry of a vertex transitive group.

Theorem 3. *Let H be a vertex transitive subgroup of the automorphism group of a given vertex transitive graph G. Let $\alpha_i \in H$ be such that $\alpha_i(i) = 1 \ \forall i$. Let M possess the symmetry of H. Then $q_{M,1}$ is SOS if and only if there exists $\bar{M}$ such that*

$$M \succeq \bar{M},\ \bar{m}_{11} = 0,\ \bar{m}_{ii} + 2m_{1,\alpha_i(1)} = 0 \ i > 1$$

$$\bar{m}_{ij} + \bar{m}_{\alpha_j(i),\alpha_j(1)} + \bar{m}_{\alpha_i(j),\alpha_i(1)} \geq 0 \ 1 < i < j.$$

Using this theorem, it is possible to optimize over the $r = 1$ approximation of C^* for rather large vertex transitive graphs. Computational results are given in [2].

References

[1] I. Dukanovic. *Semidefinite programming applied to graph coloring problem*. PhD thesis, University of Klagenfurt, Austria, 2006 (forthcoming).

[2] I. Dukanovic and F. Rendl. Copositive programming motivated bounds on the stability and the chromatic number. *Working paper, University of Klagenfurt* , 2005.

[3] E. de Klerk and D. V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, 12:875–892, 2002.

[4] L. Lovász. On the shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25:1–7, 1979.

[5] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

[6] J. Povh: A copositive and semidefinite programming approach to several combinatorial problems. Dissertation, University of Ljubljana, Faculty of mathematics and physics, Ljubljana, 2005 (forthcoming).

Reverse median problems on graphs

Johannes Hatzl
(joint work with Rainer E. Burkard, Elisabeth Gassner)

An instance of the reverse p-median problem is given by a graph $G = (V, E)$ with edge lengths $l_e \in \mathbb{R}_+$ for $e \in E$ and vertex weights $w_v \in \mathbb{R}_+$ for $v \in V$. Furthermore, a budget $B > 0$ and a set $Y = \{y_1, \ldots, y_p\} \subset V$ of prespecified vertices representing the locations of p facilities are known. The task is to use the budget in order to change the length of some edges such that the overall sum of the weighted distance of the vertices to the respective closest facility becomes as small as possible. Thereby, the distance $d^l(v_i, v_j)$ in G of vertex v_i to vertex v_j is the length of a shortest (v_i, v_j) path in G corresponding to the edge lengths l_e. In order to improve the given locations, we are allowed to reduce the edge lengths l_e. The cost for reducing the edge e by x_e units are given by some function $f_e(x_e) > 0$. Thus, the decision variables are x_e, describing the reduction of the edge lengths $\bar{l}_e = l_e - x_e$. Additionally, upper bounds u_e on the maximum allowable edge reduction are taken into account.
Using the notation introduced above, the problem can formally be stated as follows:

$$
\begin{aligned}
\min \quad & \sum_{v \in V} \left(w_v \min_{1 \le j \le p} d^{\bar{l}}(v, y_j) \right) & \\
\text{s.t.} \quad & \bar{l}_e = l_e - x_e & \forall e \in E \\
& \sum_{e \in E} f_e(x_e) \le B & \\
& 0 \le x_e \le u_e & \forall e \in E.
\end{aligned}
$$

It is known that the reverse median problem is strongly $\mathcal{NP}$-hard.

Theorem 1 ([2]). *The reverse 1-median problem on bipartite graphs is strongly $\mathcal{NP}$-hard even for the unit cost model, i.e., $f(x_e) = x_e$. Furthermore, there does not exist any polynomial time algorithm with constant approximation ratio (unless $\mathcal{P} = \mathcal{NP}$).*

In [1] a linear time algorithm for the reverse 1-median problem on a tree is proposed. Here, we restrict our analysis to the reverse 2-median problem on trees and the reverse 1-median problem on cacti with one cycle. It turns out that these problems can be solved in polynomial time. In Burkard, Gassner and Hatzl [3] it is shown that both problems under consideration can be transformed to an equivalent reverse 2-median problem on a path, which can be stated in the following form:
Let $P = (V, E)$ be a path, i.e., $|V| = |E| + 1 = n$ and $V = \{1, 2, \ldots, p-1, p, p+1, \ldots, n\}$ and $E = \{e_1, \ldots, e_{n-1}\}$ where $e_i = (i, i+1)$ for $i = 1, \ldots, n-1$. Furthermore, two prespecified vertices 1 and p are given. R2MP is then defined in

the following way:

$$\begin{aligned}
\min \quad & \sum_{j=1}^{p} w_j \min\{d^{\bar{l}}(v_j,1), d^{\bar{l}}(v_j,p)\} + \sum_{j=p+1}^{n} \left(w_j \sum_{i=p}^{j-1} \bar{l}_i \right) \\
\text{s.t.} \quad & \bar{l}_i = l_i - x_i && i = 1,\dots,n-1 \\
& 0 \le x_i \le u_i && i = 1,\dots,n-1 \\
& \sum_{i=1}^{n-1} x_i \le B.
\end{aligned}$$

For R2MP an $\mathcal{O}(n \log n)$ time algorithm is proposed in [3]. In this paper, we will only discuss some properties of an optimal solution and some ideas that will finally lead to this algorithm. For all the details we refer to [3].
An edge $e = (k, k+1)$ for $k = 1,\dots,p-1$ is called critical with respect to the modified edge lengths $\bar{l} = (\bar{l}_1,\dots,\bar{l}_{n-1})$ if

$$d^{\bar{l}}(k,1) \le d^{\bar{l}}(k,p)$$

and

$$d_i^{\bar{l}}(k+1,1) \ge d^{\bar{l}}(k+1,p).$$

This definition means that a critical edge can be deleted from the path P in order to get the shortest path tree. Let $x^* = (x_1^*,\dots,x_{n-1}^*)$ be an optimal solution and k a critical index with respect to $\bar{l} = (l_1 - x_1^*,\dots,l_{n-1} - x_{n-1}^*)$. Then x^* is an optimal solution of the following linear program $LP(k)$:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n-1} (l_i - x_i) W_i^k \\
\text{s.t.} \quad & 0 \le x_i \le u_i && i = 1,\dots,n-1 \\
& \sum_{i=1}^{n-1} x_i = B.
\end{aligned}$$

where

$$W_i^k = \begin{cases} \sum_{j=i+1}^{k} w_j & i = 1,\dots,k-1, \\ 0 & i = k, \\ \sum_{j=k+1}^{i} w_j & i = k+1,\dots,p-1, \\ \sum_{j=p+1}^{n} w_j & i = p,\dots,n-1. \end{cases}$$

Note that $LP(k)$ is a continuous knapsack problem and can therefore be solved in linear time. This observation immediately leads to an $\mathcal{O}(n^2)$ algorithm by solving $LP(k)$ for any $k \in \{1,\dots,p-1\}$. It should also be pointed out that the time complexity of $\mathcal{O}(n^2)$ can even be achieved for a more general model, namely using cost functions $f_i(x_i) = C_i x_i$ where $C_i \in \mathbb{R}_+$ for all $e_i \in E$.
However, using information from an optimal solution of $LP(k)$ for $LP(k+1)$ yields a faster algorithm for the unit cost model.

Using the notations

$$L^k := \{e_1, \ldots, e_{k-1}\};$$
$$R^k := \{e_k, \ldots, e_{p-1}\}.$$

it can be shown that if x_e^k and x_e^{k+1} are optimal solutions of $LP(k)$ and $LP(k+1)$, then

$$\sum_{e \in L^k} x_e^k \leq \sum_{e \in L^{k+1}} x_e^{k+1}$$

and

$$\sum_{e \in R^k} x_e^k \geq \sum_{e \in R^{k+1}} x_e^{k+1}.$$

Using this observation and the fact that

$$W_1^k > W_2^k > \ldots > W_{k-1}^k > W_k^k = 0 < W_{k+1}^k < \ldots < W_{p-1}^k$$

it can be concluded that if we have $x_e^k = u_e$ for some $e \in L^k$ then $x_e^{k+1} = u_e$ also holds if $e \in L^{k+1}$. Similarly, we have that $x_e^{k+1} = 0$ for some edge in R^{k+1} if $x_e^k = 0$. Using this information we do not need to solve $LP(k+1)$ from the scratch and an $\mathcal{O}(n \log n)$ time algorithm for R2MP can be achieved. For the reverse 1-median problem on a cycle an $\mathcal{O}(n)$ time algorithm is possible. Concluding we get

Theorem 2.

(1) The reverse 2-median problem on a tree can be solved in $\mathcal{O}(n \log n)$.
(2) The reverse 1-median problem on a cactus with one cycle can be solved in $\mathcal{O}(n \log n)$.
(3) The reverse 1-median problem on a cycle can be solved in $\mathcal{O}(n)$.

References

[1] O. Berman, D.I. Ingco, and A. Odoni, *Improving the location of minisum facilities through network modification*, Ann. Oper. Res. **40** (1992), 1–16.
[2] R.E. Burkard, E. Gassner, J. Hatzl, *A linear time algorithm for the reverse 1-median problem on a cycle*, to appear in Networks.
[3] R.E. Burkard, E. Gassner, J. Hatzl, *Reverse 2-median problems on trees*, submitted.

Increasing distances in graphs

Stefan Krause

We discuss a combinatorial problem related to Minimum Multicut; complexity of various special cases is determined and approximation results are presented.

Given an undirected graph $G = (V, E)$ with edge costs and pairs of vertices (s_i, t_i) for $i = 1, \ldots, k$ we want to find a minimum cost edge set S whose removal from G increases the distance of s_i and t_i, that is, the number of edges on a shortest s_i-t_i-paths by at least 1 for each $i = 1, \ldots, k$. In other words, the removal of S shall block all shortest s_i-t_i-paths for all $i = 1, \ldots, k$. This problem was first proposed by Bienstock and Diaz in [2] as Blocking Shortest Paths (BSP).

In [1] Baier discusses a more general problem where edges have arbitrary lengths and s_i-t_i-paths up to some length are to be blocked. However, we restrict ourselves to unit length instances and consider shortest paths only in order to achieve interesting and stronger results.

Instances of BSP with one pair (s, t) can be solved efficiently by constructing a special digraph $D = (V, A)$ and then determining a minimal s-t-cut in D. This result can easily be generalized to instances of BSP with pairs (s, t_i), $i = 1, \ldots, k$, that is, one vertex occurs in all pairs.

Solving BSP with two disjoint pairs is *NP*-complete, even under some assumptions about the underlying graph G, namely, if G is a grid graph with maximum degree $\Delta(G) = 3$ or if G is known to be planar and bipartite with unit costs.

If $\Delta(G) = 2$ then BSP can be solved in polynomial time even for non-unit costs and non-fixed k. BSP with unit costs and fixed k can be solved in polynomial time if G is outerplanar or if $\Delta(G)$ is bounded.

An open problem that comes to mind at once when characterizing polynomial-time solvable and *NP*-hard cases of BSP are the *trianlge instances* with three pairs (r, s), (r, t), and (s, t). The corresponding cut problem, Minimum Multiwaycut with three terminals is *NP*-hard in general graphs but polynomial-time solvable in planar graphs. However, for BSP the complexity is unknown.

A straightforward approach to approximate BSP is to separately solve the subproblems with one pair each. The union of these solutions is a factor k approximation, where k is the number of pairs.

To determine approximate solutions for BSP one can also use the well-known greedy algorithm for Set Cover. Applied to BSP this algorithm iteratively removes an edge e with maximum ratio between the number of considered shortest paths containing e and its edge cost. The approximation ratio can be shown to increase at most linear in the number of vertices and logarithmic in the number of pairs. For $k = 1$ pair an almost tight bound is known.

Another standard technique for approximation is LP based rounding. For this purpose one formulates BSP as a mixed integer program and solves its relaxation. Then all edges which are selected with a fractional value larger than some bound are removed. Unfortunately, there are instances for which all edge variables have the same fractional value ruling out good approximation results and iterative rounding. The best approximation guarantee one can give is the maximum length of the considered shortest paths.

An obvious question is whether these results hold for directed graphs as well. In fact, considering digraphs is a generalization and consequently *NP*-completeness of the problem is preserved. Additionally, triangle instances as mentioned before are *NP*-hard in the directed case. Other variations of BSP are the problems of increasing the distances by exactly 1 or of increasing them by at least $d \geq 2$. For the former it is even *NP*-hard to decide whether a feasible solution exists. The latter is *NP*-hard for non-fixed d (see [1]), or for $k \geq 2$.

Open problems. Besides some other special cases of BSP the P/NP-status of the following problems is open.

(1) Given an undirected graph with or without edge costs, find a minimum (cost) edge set whose deletion increases the distance of any two of three given vertices by at least 1.
(2) Given an undirected graph with or without edge costs, find a minimum (cost) edge set whose deletion increases the distance of two given vertices by at least some fixed $d \geq 2$.

References

[1] Georg Baier, *Flows with path restrictions*, Ph.D. thesis, Technische Universität Berlin, 2003.
[2] Daniel Bienstock and Nicole Diaz, *Blocking small cuts in a network, and related problems*, SIAM J. Comput. **22** (1993), no. 3, 482–499.

A new approach to alternating paths

Gyula Pap

Edmonds' matroid intersection algorithm, and Edmonds' non-bipartite matching algorithm are famous applications of so-called alternating paths. In these algorithms we maintain a primal solution, and in each step we try to find a better primal solution. For this purpose, we build up an augmentation structure which either helps to find the augmentation, or helps to construct a certificate of optimality. A great variety of generalizations has been investigated since then. Deep min-max characterizations were shown for a bunch of problems. Many of these characterizations are direct extensions, or analogues of those for non-bipartite matching, matroid intersection – let us mention path-matching, even factors, square-free simple 2-matchings in bipartite graphs, K_{tt}-free simple t-matchings in bipartite graphs, covering pairs of sets, packing fully node-disjoint $\mathcal{A}$-paths, matroid matching. To construct algorithms for these problems, a natural approach is to find the analogue of the augmentation structure – the auxiliary digraph in Edmonds' matroid intersection algorithm, or the alternating forest in Edmonds' matching algorithm. However, these structures are specific – it is not so easy to figure out what should be their analogue for other problems.

In this talk an interpretation of Edmonds' algorithms is proposed, which could be helpful in finding generalizations for other problems. In this interpretation we replace the augmentation structure by a so-called 3-Way Lemma. This Lemma either certifies optimality, or finds an augmentation, or finds a "nice configuration". We have a Reduction Lemma claiming that the reduction of this "nice configuration" is an equivalent reduction. An algorithm is put together from these two lemmas, we only use them as black boxes. These two lemmas admit direct, constructive proofs. The 3-Way Lemma for non-bipartite matching can in fact be proved using alternating forests; however, there is a direct proof, too.

Lemma 1 (3-Way Lemma for Matching)**.** *We are given a matching M in graph G. Then at least one of the following alternatives holds:*

a) *There is a set $Z \subseteq V$ of nodes such that $|M| = \frac{1}{2}(|V| + |Z| - is(G - Z))$, where $is(G')$ denotes the number of isolated nodes in G'.*
b) *There is a matching N with $|N| = |M| + 1$.*
c) *There is a matching N with $|N| = |M|$ s.t. there is an N-alternating odd cycle C in G incident with an N-exposed node.*

Edmonds' matching algorithm can be interpreted as follows. We maintain a pair G, M where G is a graph, and M is a matching in G. Apply the 3-Way Lemma for G, M! In case **a)** we conclude that M is a maximum matching, the set Z verifies optimality. In case **b)** we consider the new pair G, N, that is an augmentation. In case **c)** we consider the new pair $G/C, N/C$, that is a contraction. The algorithm re-applies the 3-Way Lemma for the new pair. After a sequence of contractions the algorithm is stuck with either an augmentation, or a verification of optimality. Both of these cases is easy to expand to the original graph, which is given by the following lemma:

Lemma 2 (Equivalent Reduction for matching)**.** *We are given a matching N in graph G, and an N-alternating odd cycle C incident with an N-exposed node. Then both of the following hold:*

(1) If N' is a matching in G/C with $|N'| > |N/C|$, then there is an expansion N'' in G with $|N''| > |N|$.
(2) If $|N/C| = \frac{1}{2}(|V/C| + |Z| - c(G/C - Z))$, then $|N| = \frac{1}{2}(|V| + |Z| - c(G - Z))$, where $c(G')$ denotes the number of odd components of G'.

The concept of this version of Edmonds' algorithm may be called "Relax – Reduce". Notice that the condition in **a)** is equivalent to χ_M being a maximum fractional matching – a relaxation of 0-1 matching. Thus the 3-Way Lemma may be thought of as checking, whether in the χ_M is a maximum fractional matching.

Let us note in the end that for some of the other problems mentioned above, no simple concept of alternating paths', or augmenting structure is known – this is the case for example for path-matching, K_{tt}-free simple t-matchings, and packing fully node-disjoint $\mathcal{A}$-paths. However, a 3-Way Lemma can be formulated, and proved directly, which provides a combinatorial algorithm in the end. To apply the "Relax – Reduce" principle for some combinatorial optimization problem, you have to figure out what should be the relaxation of your problem, and what sort of reduction you do with the "nice configuration". Other applications of "Relax – Reduce" are given for K_{tt}-free simple t-matchings in bipartite graphs, matroid intersection, and packing fully node-disjoint $\mathcal{A}$-paths (manuscripts of the author, to be submitted).

References

[1] G. Pap, *Alternating paths revisited I: even factors*, EGRES TR-2004-18, submitted.
[2] G. Pap, *Packings and 2-packings of node-disjoint A-paths*, manuscript.
[3] G. Pap, *Alternating paths revisited II: K_{tt}-free simple t-matchings in bipartite graphs*, manuscript.
[4] G. Pap, *Hypo-matchings in directed graphs*, to appear.

Sampling-based approximation algorithms for multi-stage stochastic optimization

Chaitanya Swamy
(joint work with David Shmoys)

Stochastic optimization problems provide a means to model uncertainty in the input data where the uncertainty is modeled by a probability distribution over the possible realizations of the data. We consider a broad class of these problems, called *multi-stage stochastic programming problems with recourse*, where the uncertainty evolves through a series of stages and one take decisions in each stage in response to the new information learned. Multi-stage stochastic programming has received a great deal of attention within the Operations Research community, both in terms of asymptotic convergence results, as well as computational work in various application domains. These problems are often computationally quite difficult, both from a practical perspective, as well as from the viewpoint of computational complexity theory; even very specialized (sub)problems are $\#P$-complete [1].

Our main result is to give the first fully polynomial randomized approximation scheme (FPRAS) for a broad class of multi-stage stochastic linear programming problems with any constant number of stages, without placing any restrictions on the underlying probability distribution or on the cost structure of the input. For a rich class of k-stage stochastic linear programs (LPs), where k is assumed to be constant and not part of the input, we show that, for any probability distribution over the inputs, for any $\varepsilon > 0$, we can compute, with high probability, a solution with expected cost at most $(1+\varepsilon)$ times the optimal expected cost, in time bounded by a polynomial in the input size, $\frac{1}{\varepsilon}$, and a parameter λ that is an upper bound on the ratio between the cost of the same action over successive stages. The algorithm accesses the input by means of a "black-box" (simulation) procedure that can generate, for any node in the scenario tree, a sample of the input according to the conditional distribution for this node. This is an extremely general model of the distribution, since it allows for all types of correlated effects within different parts of the input. We improve upon our earlier work [3], which handles the special case in which $k = 2$, in two ways: a) we are now able to handle *any fixed number of stages*, and b) whereas the earlier algorithm is based on the ellipsoid method, we can now show that a simple algorithm that is most commonly used in practice, the *sample average approximation* (SAA) method, also yields an approximation scheme. As a corollary of this FPRAS, we also obtain the first approximation algorithms for the analogous class of multi-stage stochastic integer programs, which includes the multi-stage versions of the set cover, vertex cover, multicut on trees, facility location, and multicommodity flow problems.

Although our results are much more general, to describe our results in a more meaningful way we shall focus on a canonical example of the class of problems, a 3-stage stochastic variant of the fractional set covering problem. We are given a family of sets over a ground set and a probability distribution over the subsets that specifies a target set of ground elements that must be covered. We can view the

three stages as specified by a scenario tree with 3 levels of nodes: the root, internal nodes, and leaves. The root corresponds to the initial state, each leaf is labeled with a target subset of elements that must be covered, and for each internal node there is a conditional distribution of the target sets at leaves within its subtree (where we condition on the fact that we have reached that node). One can buy (fractionally) sets at any node paying a cost that depends both on the set and the node at which it is bought. We want to be able to compute, given a node in the tree, the desired action, so as to minimize the expected total cost of fractionally covering the realized target set. This problem can be modeled as an exponentially large LP in which there is, for each set S and each node in the tree, a variable that indicates the fraction of S that is bought at that node. The constraints say that for each leaf, for each ground element e in its corresponding target set, the total fraction bought of sets S that contain e along this root-leaf path must be at least 1. If we view the probability of reaching a node as specified, it is straightforward to express the expected total cost as a linear function of these decision variables.

In the sample average approximation method, we merely sample scenarios a given (polynomial) number of times N, and by computing the frequencies of occurrence in these samples, we derive a new LP that is a polynomial-sized approximation to the original exponential-sized LP, and then solve this compact LP explicitly. We first argue that using (approximate) subgradients one can establish a notion of closeness between two functions so that if two functions are close under this notion, then minimizing one function is equivalent to approximately minimizing the other. Next, we show that the objective functions of the "true" problem, and the sample-average problem constructed with a polynomially bounded sample size, satisfy this "closeness-in-subgradients" property with high probability. This shows that minimizing the sample-average problem yields a near-optimal solution to the true problem, and proves the polynomial-time convergence of the SAA method. Our proof does not rely on anything specific to discrete probability distributions and thus extends to the case of continuous distributions.

For the class of 2-stage problems considered by Shmoys and Swamy [3], it is relatively easy to show this closeness property and thereby obtain polynomial sample bounds. For example, for the 2-stage set covering problem, one first reformulates the exponential size 2-stage LP as a compact convex program that has variables corresponding only to the decisions made at the root to (fractionally) buy sets. At any point, each component of the subgradient of this convex objective function can be estimated by sampling a leaf from the scenario tree and using the optimal dual solution for the LP that minimizes the cost to cover each element in this leaf's target set to the extent it is not already covered by the root variables. Since the variance in the subgradient components is polynomially bounded (shown in [3]), one can get a good estimate of the subgradient using polynomially many samples.

Compare now the 3-stage and 2-stage problems. The 3-stage problem can also be formulated as a compact convex program with variables corresponding only to the decisions made at the root. But in the 3-stage version, a *2-stage stochastic LP* plays the analogous role of the LP and we need to obtain a near-optimal dual

solution for this exponentially large mathematical program to show the closeness property. Moreover, one difficulty that is not encountered in the 2-stage case, is that now this *2-stage recourse LP is different in the sample average and the "true" problems*, since the conditional distribution of scenarios given a second-stage outcome is only *approximated* in the sample average problem. Thus to show the closeness property one has to argue that solving the dual of the sample average 2-stage recourse LP yields a near-optimal dual solution to the "true" 2-stage recourse LP. We introduce a novel *compact non-linear formulation of this dual*, for which we can prove such a statement for the duals, and thereby obtain the "closeness-in-subgradients" property for the 3-stage problem. In fact, this formulation yields a new means to provide lower bounds on 2-stage stochastic LPs, which might be of interest in its own right. The analogous idea can be applied inductively to obtain the FPRAS for any fixed number of stages. We believe that our proof is of independent interest and that our approach of using subgradients will find applications in proving convergence results in other stochastic models as well.

Due to its simplicity and its use in practice, the SAA method has been studied extensively in the stochastic programming literature. However, for multi-stage problems with arbitrary (correlated) distributions, to the best of our knowledge, there are no results known about the rate of convergence of the SAA solution to the true optimal solution (with high probability). In fact, we are not aware of any work (even outside of the sample average approach) that proves worst-case bounds on the sample size required for solving multi-stage stochastic linear programs with arbitrary distributions in the black-box model. Very recently, Shapiro [2] proved bounds on the sample size required in the SAA method for multi-stage problems when *the distributions in the different stages are independent*. Moreover, even for a fixed number of stages, these bounds are not polynomial in the input size or λ.

An immediate consequence of our FPRAS for multi-stage stochastic linear programs is that we obtain approximation algorithms for several natural multi-stage stochastic integer programming problems, by extending the rounding approach of [3]. In the black-box model without any cost restrictions, we obtain performance guarantees of $k \log n$ for k-stage set cover, $2k$ for k-stage vertex cover and k-stage multicut on trees, and a performance guarantee of $O(k)$ for the k-stage facility location problem. Finally, we obtain a FPRAS for the k-stage multicommodity flow problem as a direct consequence of our stochastic linear programming result.

References

[1] M. Dyer and L. Stougie. Computational complexity of stochastic programming problems. SPOR-Report 2003-20, Dept. of Mathematics and Computer Science, Eindhoven Technical University, Eindhoven, 2003.

[2] A. Shapiro. On complexity of multistage stochastic programs. Published electronically in *Optimization Online*, 2005. http://www.optimization-online.org/DB_FILE/2005/01/1041.pdf.

[3] D. B. Shmoys and C. Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. http://ist.caltech.edu/~cswamy/papers/stochopt.pdf. Preliminary version appeared as "Stochastic optimization is (almost) as easy as deterministic optimization" in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 228–237, 2004.

Using inverse optimization and flow techniques for group decision making, rank aggregation, clustering and data mining

Dorit Hochbaum

We introduce models for problems of group decision making, aggregate ranking and clustering techniques for data mining. The problems are modeled as graph problems. One of these problems we call the equal paths problem. This problem as well as all problems studied here have convex objective function representing penalties for deviating from specified a-priori comparison/ranking beliefs. These problems are shown to be solvable in polynomial time using network flow techniques such as parametric cut and fractional multicommodity linear programming.

One application of the aggregate ranking problem is to determine the ranking of sports teams based on the outcomes of games played. Current techniques are based on finding a maximum eigenvector. Our alternative model has a number of advantages including the ability to differentiate between games based on some measure of significance. Further, the problem is stated as a combinatorial graph problem. This problem is shown to be solved in polynomial time even with a convex objective function, using flow techniques.

We point out several problems relating to the robustness of a set of evaluations (game outcomes) with respect to generating a ranking. These problems appear to be new and are shown to be NP-hard.

Another area that addresses various forms of rankings has to do with data mining with applications to customer segmentation, patient diagnosis and assessment of bankruptcy risk. We demonstrate new models for these problems and how to solve them with flow techniques.

Problems of group decision making and of multi-criteria decision making are closely related to the problems above. We demonstrate that the dominant problems in these areas can also be modeled and solved with flow related techniques.

The link of all the models to inverse optimization will be described. Specifically it is shown that the inverse shortest paths problems with known paths is solved in polynomial time with flow techniques for any convex penalty function, and that the case with multiple sources and destination is solved in polynomial time using the dual of multicommodity flow problem.

Approximation algorithms for facility location

Jens Vygen
(joint work with Jens Maßberg)

We present new approximation algorithms for two facility location problems [9]. The first one is the Universal Facility Location Problem, where we improve the approximation ratio to 6.702. The second one is a problem with service capacities that occurs in VLSI design. Here we give, among other results, a fast 5-approximation algorithm and a polynomial-time 4.1-approximation algorithm.

1. Improved local search for universal facility location [8]

An instance of the Universal Facility Location Problem consists of:

- a finite set $\mathcal{D}$ of customers (or clients);
- a finite set $\mathcal{F}$ of potential facilities;
- a metric c on $V := \mathcal{D} \cup \mathcal{F}$, i.e. distances $c_{ij} \geq 0$ $(i, j \in V)$ with $c_{ij} = c_{ji}$ and $c_{ij} + c_{jk} \geq c_{ik}$ for all $i, j, k \in V$;
- a demand $d_j \geq 0$ for each $j \in \mathcal{D}$;
- and for each $i \in \mathcal{F}$ a cost function $f_i : \mathbb{R}_+ \to \mathbb{R}_+ \cup \{\infty\}$, which is left-continuous and non-decreasing.

We look for:

- a function $x : \mathcal{F} \times \mathcal{D} \to \mathbb{R}_+$ with $\sum_{i \in \mathcal{F}} x_{ij} = d_j$ for all $j \in \mathcal{D}$

such that $c(x) := c_F(x) + c_S(x)$ is minimum, where

$$c_F(x) := \sum_{i \in \mathcal{F}} f_i\left(\sum_{j \in \mathcal{D}} x_{ij} \right) \qquad \text{and} \qquad c_S(x) := \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{D}} c_{ij} x_{ij}.$$

$f_i(z)$ is interpreted to be the cost to install capacity z at facility i. These functions are given by an oracle that, for each $i \in \mathcal{F}$, $u, c \in \mathbb{R}_+$ and $t \in \mathbb{R}$, computes $f_i(u)$ and $\max\{\delta \in \mathbb{R} : u + \delta \geq 0,\ f_i(u + \delta) - f_i(u) + c|\delta| \leq t\}$. This oracle can be implemented trivially for all special cases of the Universal Facility Location Problem considered before (for example the Capacitated Facility Location Problem).

The first approximation algorithm for the Capacitated Facility Location Problem (with performance ratio 8.53) was due to Pál, Tardos and Wexler [7], extending an earlier result for uniform capacities by Korupolu, Plaxton and Rajamaran [4]. Mahdian and Pál [5] generalized this to the Universal Facility Location Problem and improved the approximation ratio to 7.88. Their main local search operation is called Pivot and moves demand from/to a certain facility to/from other facilities.

The approximation guarantee for the Capacitated Facility Location Problem was then improved to 5.83 by Zhang, Chen and Ye [10], using a more general operation which Garg, Khandekar and Pandit [2] call DoublePivot. However, it is not known whether DoublePivot can be implemented in polynomial time for the Universal Facility Location Problem. Zhang, Chen and Ye [10] remarked that they can improve the approximation ratio for the Universal Facility Location Problem to $7 + \varepsilon$ for any positive ε without DoublePivot. Very recently, Garg, Khandekar and Pandit [2] announced an approximation ratio of 5.83 for the Universal Facility Location Problem, but they withdrew this claim after seeing that they could not show how to implement DoublePivot in polynomial time [V. Pandit, personal communication 2005].

We introduce a new operation that allows us to improve the approximation ratio for the Universal Facility Location Problem to 6.702. This is the best approximation ratio known today. The operation is a generalization of the Pivot operation but can deal with general forests on the set of facilities rather

than stars. We use it for comets only, graphs that result from a star by appending one edge.

Note that in all above-mentioned results the demand of a customer can be split, i.e. served by more than one facility. Unless $P = NP$, no approximation algorithms exist for the variant where each customer must be served by a single facility, even for the CAPACITATED FACILITY LOCATION PROBLEM, as the *NP*-complete PARTITION problem polynomially transforms to the decision whether a solution of finite cost exists.

2. Network design and facility location with service capacities [6]

We present the first constant-factor approximation algorithms for the following problem: Given

- a metric space (V, c),
- a set $D \subseteq V$ of terminals/customers,
- demands $d : D \to \mathbb{R}_+$,
- a facility opening cost $f \in \mathbb{R}_+$, and
- a capacity $u \in \mathbb{R}_+$,

find

- a partition $D = D_1 \dot\cup \cdots \dot\cup D_k$ and
- Steiner trees T_i for D_i $(i = 1, \ldots, k)$

with

$$c(E(T_i)) + d(D_i) \leq u$$

for $i = 1, \ldots, k$, such that

$$\sum_{i=1}^{k} c(E(T_i)) + kf$$

is minimum.

This problem arises in VLSI design. It generalizes the bin-packing problem and the Steiner tree problem. In contrast to other network design and facility location problems, it has the additional feature of upper bounds on the service cost that each facility can handle.

Among other results, we obtain a 4.1-approximation in polynomial time, a 4.5-approximation in cubic time and a 5-approximation as fast as computing a minimum spanning tree on (D, c).

The algorithms start by constructing a tree connecting all terminals. In the first algorithm, this is a minimum spanning tree, in the second one it is an approximate Steiner tree (with respect to a slightly different metric), and in the third one it results from an approximate tour by deleting an edge. The second step deletes expensive edges. The third step splits connected components whose load exceeds the capacity u.

In the rectilinear plane $(V, c) = (\mathbb{R}^2, \ell_1)$, the most relevant case for VLSI design, we have a 4-approximation algorithm running in $O(n \log n)$ time, where $n = |D|$. Even in this case there is no $(2 - \varepsilon)$-approximation algorithm unless $P = NP$.

This algorithm is already used for two applications in VLSI design. One consists of constructing efficient repeater trees [1]. The other application, in which the problem arises naturally, and which was the motivation to study the problem, is clocktree design [3]. Here the customers are storage elements which have to be served by a periodic clock signal, generated (or distributed) by special objects, the facilities that are to be placed. Exchanging a previously used greedy heuristic by the new approximation algorithm led to 10% less power consumption on many complex chips.

References

[1] C. Bartoschek, S. Held, D. Rautenbach, J. Vygen, *Efficient generation of short and fast repeater tree topologies*, accepted for publication in the Proceedings of the International Symposium on Physical Design (2006).

[2] N. Garg, R. Khandekar, V. Pandit, *Improved approximation for universal facility location*, Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (2005), 959–960.

[3] S. Held, B. Korte, J. Maßberg, M. Ringe, J. Vygen, *Clock scheduling and clocktree construction for high performance ASICs*, Proceedings of the IEEE International Conference on Computer-Aided Design (2003), 232–239.

[4] M.R. Korupolu, C.G. Plaxton, R. Rajaraman, *Analysis of a local search heuristic for facility location problems*, Journal of Algorithms **37** (2000), 146–188.

[5] M. Mahdian, M. Pál, *Universal facility location*, in: Algorithms - ESA 2003, Proceedings of the 11th ESA Conference; LNCS 2832 G. di Battista, U. Zwick, eds.), Springer, Berlin 2003, pp. 409–421.

[6] J. Maßberg, J. Vygen, *Approximation algorithms for network design and facility location with service capacities*, in: Approximation, Randomization and Combinatorial Optimization; Proceedings of the 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2005); LNCS 3624 (C. Chekuri, K. Jansen, J.D.P. Rolim, L. Trevisan, eds). Springer, Berlin 2005, pp. 158–169.

[7] M. Pál, É. Tardos, T. Wexler, *Facility location with hard capacities*, Proceedings of the 42nd Annual IEEE Symposium on the Foundations of Computer Science (2001), 329–338.

[8] J. Vygen, *From stars to comets: improved local search for universal facility location.* Report No. 05947-OR, Research Institute for Discrete Mathematics, University of Bonn, 2005.

[9] J. Vygen, *Approximation algorithms for facility location problems (lecture notes)*, Report No. 05950-OR, Research Institute for Discrete Mathematics, University of Bonn, 2005.

[10] J. Zhang, B. Chen, Y. Ye, *Multi-exchange local search algorithm for the capacitated facility location problem*, in: Integer Programming and Combinatorial Optimization; Proceedings of the 10th International IPCO Conference; LNCS 3064 (G. Nemhauser, D. Bienstock, eds.), Springer, Berlin 2004, pp. 219–233.

Covering bi-set functions by digraphs

András Frank

In several applications in the area of connectivity augmentation, it proves useful to work with functions defined on pairs of sets rather than sets. Given a ground-set V, by a **bi-set** $X = (X_O, X_I)$ we mean a pair of subsets X_O, X_I of V for which $\emptyset \subseteq X_I \subseteq X_O \subseteq V$. X_O is the **outer** member of X while X_I is the **inner** member. Let $\mathcal{P}_2 = \mathcal{P}_2(V)$ denote the set of all bi-sets of V. The subset of $\mathcal{P}_2$ consisting of all bi-sets for which $X_I = X_O$ may be identified with the set of all subsets of V

and therefore all results concerning bi-sets may be specialized to those on sets. A function on $\mathcal{P}_2(V)$ will be called a bi-set function on V.

A directed edge $a = uv$ **enters** or **covers** a bi-set $X = (X_O, X_I)$ if a enters both X_O and X_I. a **leaves** X if it leaves both X_O and X_I. For a directed graph $D = (V, A)$, $\varrho(X) := \varrho_D(X) := \varrho_A(X)$ denotes the number of edges entering (covering) X while $\delta(X) := \delta_D(X) := \delta_A(X)$ denotes the number of edges leaving X. For a vector $z : A \to \mathbf{R}$, let $\varrho_z(X) := \sum[z(a) : a \in A, a \text{ covers } X]$. $\delta_z(X)$ is defined analogously. Let $D = (V, A^*)$ denote the the complete directed graph on node set V in which ordered pair of nodes $\{u, v\}$ defines an edge uv.

Let p be a bi-set function on V. A digraph $D = (V, A)$ **covers** p if $\varrho_D(X) \geq p(X)$ for every $X \in \mathcal{P}_2(V)$. A function $z : A \to \mathbf{R}$ covers p if $\varrho_z(X) \geq p(X)$ for every $X \in \mathcal{P}_2(V)$. Two bi-sets are **independent** (with respect to D) if no directed edge of D can cover both. For the complete digraph $D = (V, A^*)$, this is equivalent to requiring that their inner sets are disjoint or their outer sets are co-disjoint. A family of bi-sets is **independent** if their members are pairwise independent.

The intersection $\cap$ and the union $\cup$ of bi-sets is defined in a staightforward manner: for $X, Y \in \mathcal{P}_2$ let $X \cap Y := (X_O \cap Y_O, X_I \cap Y_I)$, $X \cup Y := (X_O \cup Y_O, X_I \cup Y_I)$. We write $X \subseteq Y$ if $X_O \subseteq Y_O, X_I \subseteq Y_I$. When $X \subseteq Y$ or $Y \subseteq X$, then we call X and Y **comparable**. Two bi-sets are **intersecting** if they are not comparable and $X_I \cap Y_I \neq \emptyset$. Two bi-sets are **crossing** if they are intersecting and the union of their outer members are not V.

A family of bi-sets is called **laminar** (**cross-free**) if it has no two intersecting (crossing) members. A family $\mathcal{F}$ of bi-sets is **intersecting** (**crossing**) if both the union and the intersection of any two intersecting (respectively, crossing) members of $\mathcal{F}$ belong to $\mathcal{F}$.

A nonnegative integer-valued set-function $p : \mathcal{P}_2 \to \mathbf{Z}_+$ is said to satisfy the **supermodular inequality** on $X, Y \in \mathcal{P}_2$ if

$$p(X) + p(Y) \leq p(X \cap Y) + p(X \cup Y). \tag{1}$$

If the reverse inequality holds, we speak of the **submodular** inequality. p is said to be **fully supermodular** or supermodular if it satisfies the supermodular inequality for every pair of bi-sets X, Y. If (1) holds for intersecting (crossing) pairs, we speak of **interecting** (**crossing**) **supermodular** functions. Analogous notions can be introduced for submodular functions.

Sometimes (1) is required for those intersecting (crossing) pairs for $p(X) > 0$ and $p(Y) > 0$. In this case p is called **positively intersecting** (resp., **positively crossing**) **supermodular**.

Theorem 1 (Frank and Jordán, 1995)**.** *Let $p : \mathcal{P}_2(V) \to \mathbf{Z}_+$ be an integer-valued positively crossing supermodular function on bi-sets and let $D = (V, A^*)$ be the complete digraph on V. Then*

(2)
$$\min\{x(A^*) : x : A^* \to \mathbf{Z}_+ \text{ covers } p\} = \max\{\textstyle\sum_{X \in \mathcal{I}} p(X) : \mathcal{I} \subseteq \mathcal{P}_2(V) \text{ cross-free}\}.$$

This result (with slightly different wording) was used to solve the directed node-conenctivity augmentation problem. It also implies the directed edge-connectivity

augmentation theorem from [4], an extension of Győri's theorem on interval [6], and a theorem on restricted matchings in bipartite graphs [3].

Theorem 2. *Let $p : \mathcal{P}_2(V) \to \mathbf{Z}_+$ be an integer-valued positively intersecting supermodular function on bi-sets. Let $D = (V, A)$ be a digraph so that $p(Z) > 0$ implies $\varrho(Z) > 0$ for every bi-set Z and let $g : A \to \mathbf{Z}_+$ be a function. Then the system*

$$\{\varrho_x(Z) \geq p(Z) : \text{ for every } Z \in \mathcal{P}_2(V), 0 \leq x \leq g\} \tag{3}$$

is totally dual integral.

This is an extension of a result of [2] concerning set-functions. Its proof relies on the standard uncrossing technique. It can be used to describe the convex hull of the rooted k-node-connected subgraphs of a digraph.

Theorem 3. *For $i = 1, 2$ let $p_i : \mathcal{P}_2(V) \to \mathbf{Z}_+$ be an integer-valued fully supermodular function on bi-sets. Then*

$$\min\{x(A^*) : x : A^* \to \mathbf{Z}_+ \text{ covers } p_1, p_2\} =$$
$$= \max\{p_1(X_1) + p_2(X_2) : X_1, X_2 \in \mathcal{P}_2 \text{ independent}\}.$$

This theorem may be considered as an extension of Edmonds' (poly)matroid intersection theorem [1]. It is also used to derive a min-max result for the minimum number of new edges whose addition to an initial digraph results in a digraph in which there are k_i internally node-disjoint paths from s_i to t_i $(i = 1, 2)$. In a third application, we determine the minimum number of new edges whose addition to an initial bipartite graph results in an elementary bipartite graph.

References

[1] J. Edmonds, Submodular Functions, Matroids, and Certain Polyhedra, in: *Combinatorial Structures and their Applications* (R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds.), Gordon and Breach, New York, (1970) 69–87.

[2] A. Frank, *Kernel systems of directed graphs*, Acta Scientiarum Mathematicarum (Szeged) 41, 1-2 (1979) 63–76.

[3] A Frank, *Restricted t-matchings in bipartite graphs,* in: Submodularity, (guest editor S. Fujishige) Discrete Applied Mathematics, Vol. 131, Issue 2. (September 2003). 337–346.

[4] A. Frank, Augmenting graphs to meet edge-connectivity requirements, SIAM J. on Discrete Mathematics, (1992 February), Vol.5, No 1., 22–53.

[5] A. Frank and T. Jordán, Minimal edge-coverings of pairs of sets, J. Combinatorial Theory, Ser. B, Vol. 65, No. 1 (1995, September) 73–110.

[6] E. Győri, A minimax theorem on intervals, J. Combinatorial Theory, Ser. B, 37 (1984) 1–9.

Computing earliest arrival flows with multiple sources

Martin Skutella
(joint work with Nadine Baumann)

Earliest arrival flows are motivated by applications related to evacuation. In typical evacuation situations, the most important task is to get people out of an endangered building or area as fast as possible. Since it is usually not known how long a building can withstand a fire before it collapses or how long a dam can resist a flood before it breaks, it is advisable to organize an evacuation such that as much as possible is saved no matter when the inferno will actually happen. In the more abstract setting of network flows over time, the latter requirement is captured by so-called earliest arrival flows. Before we discuss this in more detail, we first give a short and descriptive introduction into flows over time.

Flows over time. We consider a network $N = (V, A)$ with capacities $u_e \geq 0$ and transit times $\tau_e \geq 0$ on the arcs $e \in A$. The capacity of an arc bounds the flow rate (i.e., flow per time) at which flow can enter the arc. The transit time of an arc specifies the amount of time it takes for flow to travel from the tail to the head of the arc. Moreover, there is a set of source nodes $S^+ \subseteq V$ and a set of sink nodes $S^- \subseteq V \setminus S^+$. Each source $s \in S^+$ has a supply $v(s) > 0$ and each sink $t \in S^-$ a demand $-v(t) > 0$ such that $\sum_{w \in S^+ \cup S^-} v(w) = 0$. A *flow over time* specifies for each arc e and each point in time the flow rate at which flow enters the arc (and leaves the arc again τ_e time units later). Flow conservation constraints require that at every point in time and for every intermediate node $w \in V \setminus (S^+ \cup S^-)$ the flow entering and leaving node w must cancel out each other.

Flows over time have been introduced by Ford and Fulkerson [6]. Given a network with a single source node s, a single sink node t, and a time horizon $\theta \geq 0$, they consider the problem of sending as much flow as possible from s to t within time θ. It turns out that a maximal s-t-flow over time can be determined by a static min-cost flow computation where transit times of arcs are interpreted as cost coefficients.

Ford and Fulkerson [6] also introduce the concept of *time-expanded networks* that consist of one copy of the node set of the given network for each time unit (we call such a copy a *time layer*). For each arc e of the original network with transit time τ_e the time-expanded network contains copies connecting any two time layers at distance τ_e. On the positive side, most flow over time problems can be solved by static flow computations in time-expanded networks. On the negative side, time-expanded networks are huge in theory and in practice. In particular, the size of a time expanded network is linear in the given time horizon θ and therefore exponential (but still pseudopolynomial) in the input size.

Hoppe and Tardos [11] consider the *quickest transshipment problem* which is defined as follows. Given a network with several source and sink nodes with given supplies and demands, find a flow over time with minimal time horizon θ that

satisfies all supplies and demands. Hoppe and Tardos give a strongly polynomial algorithm for this problem which, however, relies on submodular function minimization and is highly nontrivial.

Earliest arrival flows. Shortly after Ford and Fulkerson introduced flows over time, the more elaborate *s-t-earliest arrival flow problem* was studied by Gale [7]. Here the goal is to find a single s-t-flow over time that simultaneously maximizes the amount of flow reaching the sink t up to any time $\theta \geq 0$. A flow over time fulfilling this requirement is said to have the *earliest arrival property* and is called *earliest arrival flow*. Gale [7] showed that s-t-earliest arrival flows always exist. Minieka [14] and Wilkinson [17] both gave pseudopolynomial-time algorithms for computing earliest arrival flows based on the Successive Shortest Path Algorithm. Hoppe and Tardos [10] present a fully polynomial time approximation scheme for the earliest arrival flow problem that is based on a clever scaling trick.

In a network with several sources and sinks with given supplies and demands, flows over time having the earliest arrival property do not necessarily exist [3]. We give a simple counterexample with one source and two sinks. For the case of several sources with given supplies and a single sink, however, earliest arrival flows do always exist [15]. This follows, for example, from the existence of lexicographically maximal flows in time-expanded networks; see, e.g., [14]. We refer to this problem as the *earliest arrival transshipment problem*. Hajek and Ogier [8] give the first polynomial time algorithm for the earliest arrival transshipment problem with zero transit times. Fleischer [3] gives an algorithm with improved running time. Fleischer and Skutella [5] use condensed time-expanded networks to approximate the earliest arrival transshipment problem for the case of arbitrary transit times. They give an FPTAS that approximates the time delay as follows: For every time $\theta \geq 0$ the amount of flow that should have reached the sink in an earliest arrival transshipment by time θ, reaches the sink at latest at time $(1+\varepsilon)\theta$. Tjandra [16] shows how to compute earliest arrival transshipments in networks with time dependent supplies and capacities in time polynomial in the time horizon and the total supply at sources. The resulting running time is thus only pseudopolynomial in the input size.

Earliest arrival flows are motivated by applications related to evacuation. In the context of emergency evacuation from buildings, Berlin [1] and Chalmet et al. [2] study the quickest transshipment problem in networks with multiple sources and a single sink. Jarvis and Ratliff [12] show that three different objectives of this optimization problem can be achieved simultaneously: (1) Minimizing the total time needed to send the supplies of all sources to the sink, (2) fulfilling the earliest arrival property, and (3) minimizing the average time for all flow needed to reach the sink. Hamacher and Tufecki [9] study an evacuation problem and propose solutions which further prevents unnecessary movement within a building.

Our contribution. While it has previously been observed that earliest arrival transshipments exist in the general multiple-source single-sink setting, the problem of computing one efficiently has been open. All previous algorithms rely on time expansion of the network into exponentially many time layers. We solve this open

problem and present an efficient algorithm which, in particular, does not rely on time expansion.

Using a necessary and sufficient criterion for the feasibility of transshipment over time problems given by Klinz [13], we first recursively construct the earliest arrival pattern, that is, the piece-wise linear function that describes the time-dependent maximum flow value. As a by-product, we present a new proof for the existence of earliest arrival flows that does not rely on time expansion. We finally show how to turn the earliest arrival pattern into an earliest arrival flow by slightly extending the network and applying the quickest transshipment algorithm of Hoppe and Tardos [11].

The running time of our algorithm is polynomial in the input size plus the number of breakpoints of the earliest arrival pattern. Since the earliest arrival pattern is more or less explicitly part of the output of the earliest arrival transshipment problem, we can say that the running time of our algorithm is polynomially bounded in the input plus output size.

References

[1] G. N. Berlin. *The use of directed routes for assessing escape potential*. National Fire Protection Association, Boston, MA, 1979.

[2] L. G. Chalmet, R. L. Francis, and P. B. Saunders. Network models for building evacuation. *Management Science*, 28:86–105, 1982.

[3] L. K. Fleischer. Faster algorithms for the quickest transshipment problem. *SIAM Journal on Optimization*, 12:18–35, 2001.

[4] L. K. Fleischer and M. Skutella. Quickest flows over time. Submitted. Revised 2005.

[5] L. K. Fleischer and M. Skutella. The quickest multicommodity flow problem. In W. J. Cook and A. S. Schulz, editors, *Integer Programming and Combinatorial Optimization*, volume 2337 of *Lecture Notes in Computer Science*, pages 36–53. Springer, Berlin, 2002.

[6] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[7] D. Gale. Transient flows in networks. *Michigan Mathematical Journal*, 6:59–63, 1959.

[8] B. Hajek and R. G. Ogier. Optimal dynamic routing in communication networks with continuous traffic. *Networks*, 14:457–487, 1984.

[9] H. W. Hamacher and S. Tifecki. On the use of lexicographic min cost flows in evacuation modeling. *Naval Research Logistics*, 34:487–503, 1987.

[10] B. Hoppe and É. Tardos. Polynomial time algorithms for some evacuation problems. In *Proceedings of the 5th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 433–441, Arlington, VA, 1994.

[11] B. Hoppe and É. Tardos. The quickest transshipment problem. *Mathematics of Operations Research*, 25:36–62, 2000.

[12] J.J. Jarvis and H.D. Ratliff. Some equivalent objectives for dynamic network flow problems. *Management Science*, 28:106–108, 1982.

[13] B. Klinz. Cited as personal communication (1994) in [11].

[14] E. Minieka. Maximal, lexicographic, and dynamic network flows. *Operations Research*, 21:517–527, 1973.

[15] D. Richardson and É. Tardos. Cited as personal communication (2002) in [4].

[16] S. Tjandra. *Dynamic Network Optimization with Application to the Evacuation Problem*. PhD thesis, Univeristät Kaiserslautern, Shaker Verlag, Aachen, 2003.

[17] W. L. Wilkinson. An algorithm for universal maximal dynamic flows in a network. *Operations Research*, 19:1602–1612, 1971.

Provably near-optimal dual-balancing policies for stochastic inventory control models

Retsef Levi
(joint work with Robin Roundy, Martin Pál, David Shmoys, Van Anh Truong)

In this talk we address the long-standing problem of finding computationally efficient and provably good inventory control policies in supply chains with correlated, non-stationary (time-dependent) and evolving stochastic demands. This problem arises in many domains and has many practical applications such as dynamic forecast updates. We consider a broad class of fundamental inventory models and provide what we believe to be the first computationally efficient policies with constant worst-case performance guarantees; that is, there exists a constant C such that, for any given joint distribution of the demands, the expected cost of the policy is guaranteed to be within C times the expected cost of an optimal policy. More specifically, we provide a worst-case performance guarantee of 2 for these large class of models.

The models. In this talk we focus on the *periodic-review stochastic inventory control problem*. A sequence of random demands for a single commodity at a single location occurs over a finite planning horizon of T discrete periods. The random demands over the T periods can be non-stationary, correlated and evolve over time. The goal is to coordinate a sequence of orders over the planning horizon aiming to satisfy these demands with minimum expected cost. In each period s, we can order a number of units up to a given capacity u_s. These units are assumed to arrive only after a lead time of L periods. We consider a traditional cost structure with per unit ordering, holding and backlogging penalty costs that are incurred at the end of each period. The cost parameters are time-dependent and the only assumption is that we do not have a speculative motivation to hold inventory or have shortages. The goal is to find a policy of orders with minimum expected overall discounted cost over the given planning horizon.

The assumptions that we make on the demand distributions are very mild and generalize all of the currently known approaches in the literature to model correlation and non-stationarity of demands over time (for details about the different approaches we refer the reader to Iida and Zipkin 2001 and Lingxiu and Lee 2003 [3, 1]). As part of the model, we will assume that at the beginning of each period, we are given what we call an *information set* that contains all of the information that is available at the beginning of the time period (e.g., the realized demands so far and external information that becomes available). In addition, we assume that in each period there is a known conditional joint distribution of the future demands that is a function of the observed information set at the beginning of the period (but is independent of the specific inventory policy). The only assumption on the demands is that in each period and for each observed information set, all the future demands are well-defined and have finite mean. We consider only policies that are *non-anticipatory*, i.e., in each period they can use only the current information set.

Related Literature. These models have attracted the attention of many researchers over the years and there exists a huge body of related literature. The dominant paradigm in almost all of the existing literature has been to formulate these models using a dynamic programming framework. This framework has turned out to be very effective in characterizing the optimal policy of the overall system. Surprisingly, the optimal policies for these rather complex models follow simple forms, known as *state-dependent base-stock policies* (see [3, 1, 14] for details). In each period, there exists an optimal target base-stock level that is determined only by the given (observed) information set. The optimal policy aims to keep the inventory level at each period as close as possible to the target base-stock level. That is, it orders up to the target level (or up to capacity) whenever the inventory level at the beginning of the period is below that level, and orders nothing otherwise.

Unfortunately, these rather simple forms of policies do not always lead to efficient algorithms for computing the optimal policies. This is especially true in the presence of correlated and non-stationary demands, which cause the state space of the relevant dynamic programs to grow exponentially and explode very fast. This phenomena is known as *the curse of dimensionality*. Moreover, because of this phenomenon, it seems unlikely that there exists an efficient algorithm to solve these huge dynamic programs.

Muharremoglu and Tsitsiklis (see [9]) have proposed an alternative approach to the dynamic programming framework. They have observed that this problem can be decoupled into a series of *unit supply-demand subproblems*, where each subproblem corresponds to a single unit of supply and a single unit of demand that are matched together. This novel approach enabled them to substantially simplify some of the dynamic programming based proofs on the structure of optimal policies, as well as to prove several important new structural results. However, their computational methods are essentially dynamic programming approach applied to the unit subproblems, and hence they suffer from similar problems in the presence of correlated and non-stationary demand.

As a result of this apparent computational intractability, many researchers have attempted to construct computationally efficient (but suboptimal) heuristics for these problems. However, we are aware of no computationally efficient policies for which there exist constant performance guarantees. For details on some of the proposed heuristics and a discussion of others see [3, 1]. one specific class of suboptimal policies that has attracted a lot of attention is the class of *myopic policies*. In a myopic policy, in each period we attempt to minimize the expected cost for that period, ignoring the impact on the cost in future periods. The myopic policy is attractive since it yields a base-stock policy that is easy to compute online, that is, it does not require information on the control policy in future periods. In many cases, the myopic policy seems to perform well (see for example [13, 2, 3]). However, in many other cases, especially when the demand can drop significantly from period to period, the myopic policy performs poorly and even arbitrarily bad (see [6]).

Our work. is distinct from the existing literature in several significant ways, and is based on three novel ideas:

Marginal cost accounting scheme. We introduce a novel approach for cost accounting in stochastic inventory control problems. The standard dynamic programming approach directly assigns to the decision of how many units to order in each period only the expected holding and backlogging costs incurred in that period although this decision might effect the costs in future periods. Instead, our new cost accounting scheme assigns to the decision in each period *all* the expected costs that, once this decision is made, become independent of any decision made in future periods, and are dependent only on the future demands. The marginal holding cost accounting approach is based on the key observation that once we place an order for a certain number of units in some period, then the expected ordering and holding cost that these units are going to incur over the rest of the planning horizon is a function only of the realized demands over the rest of the horizon, not of future orders. Hence, with each period, we can associate the overall expected ordering and holding cost that is incurred by the units ordered in this period, over the entire horizon. We note that similar ideas of holding cost accounting were previously used in the context of models with continuous time, infinite horizon and stationary (Poisson distributed) demand (see, for example, the work of Axsäter and Lundell [11] and Axsäter [10]). In an uncapacitated model the decision of how many units to order in each period effect the expected backlogging cost in only a single future period, namely, a lead time ahead. However, this is not necessarily true in a capacitated model, where this decision might effect the expected backlogging cost in several periods into the future. Thus, for capacitated models we introduce a marginal backlogging cost accounting approach. Suppose that in the current period the order placed was not up to capacity, we wish to account for the potential backlogging cost in future periods incurred directly by the decision not to use the full available capacity. Assume temporarily that we order up to capacity in each one of the periods. Suppose now that in the current period we do not order up to capacity. Then expected marginal backlogging cost associated with the current period is the overall increase in the expected backlogging cost over the entire horizon resulting from this decision. The marginal backlogging cost accounting scheme for the capacitated model is in fact a generalization of the traditional period backlogging cost accounting scheme. As we will show it turns out that both the expected marginal holding and backlogging costs are straightforward to compute in most common scenarios. We believe that this new approach will have more applications in the future in analyzing stochastic inventory control problems.

Cost balancing. The idea of cost balancing was used in the past to construct heuristics with constant performance guarantees for deterministic inventory problems. The most well-known examples are the Silver-Meal Part-Period balancing heuristic for the lot-sizing problem (see [12]) and the Cost-Covering heuristic of

Joneja for the joint-replenishment problem [4]. We are not aware of any application of these ideas to stochastic inventory control problems. The key observation is that any policy in any period incurs potential expected costs due to overordering (namely, expected holding costs) and underordering (namely, expected backlogging costs). For the periodic-review stochastic inventory control problem (both uncapacitated and capacitated variants), we use the marginal cost accounting approach to construct policies that, in each period, balance the expected (marginal) ordering and holding cost against the expected (marginal) backlogging cost.

Using these ideas we provide what is called a 2-approximation algorithm for the uncapacitated and capacitated variants of the periodic-review stochastic inventory control problem; that is, the expected cost of our policies is no more than twice the expected cost of an optimal policy [6, 8]. Moreover, these ideas have been extended in subsequent work and have been used to provide 2-approximation for multi-echelon inventory models [7] and for models with lost-sales [5].

References

[1] L. Dong and H. L. Lee. Optimal policies and approximations for a serial multiechelon inventory system with time-correlated demand. *Operations Research*, 51, 2003.

[2] E. Ignall and A. F. Venott. Optimality of myopic inventory policies for several substitute products. *Management Science*, 15:284–304, 1969.

[3] T. Iida and P. Zipkin. Approximate solutions of a dynamic forecast-inventory model. Working paper, 2001.

[4] D. Joneja. The joint replenishment problem: new heuristics and worst case performance bounds. *Operations Research*, 38:723–771, 1990.

[5] R. Levi, G. Janakiraman, and M. Nagarajan. Provably near-optimal balancing policies for stochastic inventory control models with lost-sales. 2005. Submitted.

[6] R. Levi, M. Pal, R. O. Roundy, and D. B. Shmoys. Approximation algorithms for stochastic inventory control models. Technical Report TR1412, ORIE Department, Cornell University, 2004. Submitted.

[7] R. Levi, R. O. Roundy, , and V. A. Truong. Provably near-optimal balancing policies for multi-echelon stochastic inventory control models. Working paper, 2005.

[8] R. Levi, R. O. Roundy, D. B. Shmoys, and V. A. Truong. Approximation algorithms for capacitated stochastic inventory control models. submitted, 2004.

[9] A. Muharremoglu and J. N. Tsitsiklis. A single-unit decomposition approach to multi-echelon inventory systems. Working paper, 2001.

[10] S. Axsäter. Simple solution procedures for a class of two-echelon inventory problems. *Operations Research*, 38:64–69, 1990.

[11] S. Axsäter and P. Lundell. In process safety stock. In *Proceedings of the 23rd IEEE Conference on Decision and Control*, pages 839–842, 1984.

[12] E. A. Silver and H. C. Meal. A heuristic selecting lot-size requirements for the case of a deterministic time varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, 14:64–74, 1973.

[13] A. F. Veinott. Optimal policy for a multi-product, dynamic, non-stationary inventory problem. *Management Science*, 12:206–222, 1965.

[14] P. H. Zipkin. *Foundation of inventory management*. The McGraw-Hill Companies, Inc, 2000.

On the structure of Lehman matrices

Bertrand Guenin
(joint work with Gérard Cornuéjols, Levent Tuncel)

1. Introduction

One of the best studied [3] integer programming model is the *set covering problem*. Given a $0,1$ matrix M and a non-negative cost vector c, we wish to solve,

$$\min\{c^T x : Mx \geq \mathbf{1}, x \geq \mathbf{0},\ x \text{ integer}\}. \tag{IP}$$

Unfortunately, the set covering problem is NP-Complete. There are however, many instances where the natural linear programming relaxation always has an integer solution among the optimal solutions. In theses cases linear programming can be used to solve the set covering problem. The integer program (IP) has this property exactly when the polyhedron $Q := \{Mx \geq \mathbf{1}, x \geq \mathbf{0}\}$ is integral. The matrix M is said to be *Ideal* in this case. Hence, a fundamental question is whether it is possible to characterize Ideal matrices.

We say that a $0,1$ matrix M is *Minimally Non Ideal* (MNI) if $Q = \{Mx \geq \mathbf{1}, x \geq \mathbf{0}\}$ has a fractional extreme point but all polyhedra obtained by setting any of the variables x_j to either 0 or 1 are integral. The problem of finding an explicit characterization of all MNI matrices has a counterpart for the set packing problem, namely the Strong Perfect Graph Theorem [7]. Cornuéjols and Novick [4] conjectured that a similar result exists in this case as well.

A first step towards a characterization of MNI matrices is given by a beautiful theorem of Lehman [5]. We state this result next. Given a MNI matrix M, the polyhedron $Q = \{Mx \geq \mathbf{1}, x \geq \mathbf{0}\}$ has a unique fractional point $\bar{x}$ and that point is non-degenerate. It follows that we can define two square, $0,1$ matrices A and B as follows: the rows of A are the rows of M corresponding to constraints satisfied at equality for $\bar{x}$ and the rows of B correspond to each of the extreme points of Q which are adjacent to $\bar{x}$. Then (except for one exception) after permutation of the rows of A the following linear equation holds:

$$AB^T = E + kI, \tag{1}$$

where E is the matrix of all ones, and k some positive integer. We call matrices which satisfy the previous relation *Lehman matrices*. Luetolf and Margot [6] enumerated, all (up to isomorphism) Lehman matrices of small sizes. At first sight these matrices seem to be of a bewildering complexity. However, these matrices are endowed with very strong structural properties.

Lehman matrices A, B appear to fall into two families depending on whether $k = 1$ or $k > 1$ in equation (1). We call matrices in these families respectively *thin* and *thick*. When $A = B$ then A is the point-line incidence matrix of a projective plane. The only known examples of thick matrices where $A \neq B$ are when one of A, B corresponds to the incidence matrix of a $(3,5)$-cage (the Petersen graph) or of a $(3,7)$-cage [1]. This leads to the following natural question: except for a

number of sporadic examples, do all thick Lehman matrices arise from projective planes ? We are interested here, however, in finding a structural characterization of thin Lehman matrices. We present a first step in that direction.

2. The result

A square $0,1$ matrix is *circulant* if each row is shifted (cyclically) one element to the right relative to the preceding row. The matrix C_r^n is the $n \times n$ circulant matrix where the non-zero entries of the first row are the first r entries. For integers r, s such that $rs = n+1$, the matrix D_s^n is the $0,1$ matrix such that $C_r^n (D_s^n)^T = E + I$. Thus C_r^n, D_s^n form a pair of thin Lehman matrices.

Two matrices X, Y are *isomorphic* if Y can be obtained from X by permuting the columns and rows of X. A square $0,1$ matrix is r-regular if $Xe = X^T e = re$ for some integer r where e denotes the vector of all ones. It is known that Lehman matrices are regular [2]. Hence, for a Lehman matrix A which is r-regular, $A = C_r^n + \Sigma_A$ where Σ_A is a $0, \pm 1$ matrix which is 0-regular. We define the *C-rank* of a thin r-regular Lehman matrix A to be the minimum, over all matrices A' isomorphic to A, of $rank(\Sigma_{A'})$ where $C_r^n + \Sigma_{A'} = A'$. Thus the C-rank is a measure of the complexity of thin Lehman matrices. Matrices with zero C-rank are isomorphic to a circulant matrix C_r^n. A *basic* Lehman matrix is a thin Lehman matrix A for which C-crank$(A) = 1$. Our main result is a complete characterization of basic Lehman matrices.

Consider a positive integer n and integers $i \in [n]$ and j such that $-n \leq j \leq n$. We define,

$$i \oplus j = \begin{cases} i + j - n & \text{if } i + j > n \\ i + j + n & \text{if } i + j < 1 \\ i + j & \text{otherwise.} \end{cases}$$

Consider a positive integer n and $i_0, i_1, i_2 \in [n]$. If there exists $t, t' \in [n-1]$ such that $i_0 \oplus t = i_1$ and $i_1 \oplus t' = i_2$ then we write $i_0 \leq_{[i_1]} i_2$. It means that following the cyclical ordering and starting from i_0 we visit i_1 before i_2. Consider a positive integer n and $i_0, j_0 \in [n]$, $n_R, n_C \in [n-1]$. A *block* $\mathcal{B}(i_0, j_0, n_R, n_C)$ is a $0,1$ matrix such that its entry (i, j) is 1 if and only if $i_0 \leq_{[i]} i_0 \oplus n_R - 1$ and $j_0 \leq_{[j]} j_0 \oplus n_C - 1$.

A *configuration* $\mathcal{C}(i, j, n_R, n_C, \rho, \sigma)$ consists of four blocks B_1, B_2, B_3, B_4 where B_1 is a block $\mathcal{B}(i, j, n_R, n_C)$, B_2 is a block $\mathcal{B}(i \oplus \rho, j, n_R, n_C)$, B_3 is a block $\mathcal{B}(i, j \oplus \sigma, n_R, n_C)$, and B_4 is a block $\mathcal{B}(i \oplus \rho, j \oplus \sigma, n_R, n_C)$. We define, $\Sigma(\mathcal{C}) := -B_1 + B_2 + B_3 - B_4$. A *rectangular configuration* is the configuration $\mathcal{C}(a, a \oplus n_R, n_R, r - n_R, tr, tr-1)$ where $a \in [n]$, $n_R \in [r-1]$, $t \in [s-1]$, and $C_r^n + \Sigma(\mathcal{C})$ is a *rectangular perturbation*. Our main result is as follows,

Theorem 1. *Let A, B be a a pair of basic Lehman matrices. Then up to isomorphism, both A and B are rectangular perturbations.*

As an example consider the following matrices,

$$A = \begin{pmatrix}
1&1&&&&1&1&1&&&&&&\\
&1&&&1&1&1&1&&&&&&\\
&&1&1&1&1&1&&&&&&&\\
&&&1&1&1&1&1&&&&&&\\
&&&&1&1&1&1&1&&&&&\\
&&1&1&1&1&&&1&&&&&\\
&&1&1&1&&&&1&1&&&&\\
&&&&&&&1&1&1&1&1&&\\
&&&&&&&&1&1&1&1&1&\\
&&&&&&&&&1&1&1&1&1\\
1&&&&&&&&&1&1&1&1&\\
1&1&&&&&&&&&1&1&1&\\
1&1&1&&&&&&&&&1&1&\\
1&1&1&1&&&&&&&&&&1
\end{pmatrix}
\quad
B' = \begin{pmatrix}
&1&1&&&&1&&&&&&&\\
&1&1&1&&&&&&&&&&\\
&&1&1&1&&&&&&&&&\\
&&&1&1&1&&&&&&&&\\
1&&&&1&1&&&&&&&&\\
1&&&&&1&&1&&&&&&\\
&&&&&&1&1&1&&&&&\\
&&&&&&&1&1&1&&&&\\
&&&&&&&&1&1&1&&&\\
&&&&&&&&&1&1&1&&\\
&&&&&&&&&&1&1&1&\\
&&&&&&&&&&&1&1&\\
&1&&&&1&&&&&&&&1
\end{pmatrix}$$

Matrix B' is isomorphic to a matrix B such that $AB^T = E + I$, i.e. A, B are a pair of thin Lehman matrices. The configuration corresponding to A is $\mathcal{C}(1, 3, 2, 3, 5, 4)$ (a rectangular configuration for $a = 1$, $n_R = 2$, and $t = 1$). The configuration corresponding to B' is $\mathcal{C}(5, 7, 2, 1, 9, 8)$ (a rectangular configuration for $a = 5$, $n_R = 2$, and $t = 3$).

We also showed that we can construct matrices with arbitrary high C-rank by combining matrices with lower C-rank and that under certain conditions, we can decompose matrices with high C-rank into matrices with lower C-rank.

References

[1] R. R. Singleton A. J. Hoffman. On moore graphs with diameters 2, 3. *IBM Journal Res. Dev.*, 4:497–504, 1960.

[2] W. G. Bridges and H. J. Ryser. Combinatorial designs and related systems. *J. Algebra*, 13:432–446, 1969.

[3] G. Cornuéjols. *Packing and Covering*. Society for Industrial and Applied Mathematics, 2001. ISBN 0-89871-481-8.

[4] G. Cornuéjols and B. Novick. Ideal 0, 1 matrices. *J. Comb. Theory Ser. B*, 60:145–157, 1994.

[5] A. Lehman. On the width-length inequality and degenerate projective planes. In W. Cook and P.D. Seymour, editors, *Polyhedral Combinatorics*, volume 1 of *DIMACS Series in Discrete Math. and Theoretical Computer Science*, pages 101–105, 1990.

[6] C. Luetolf and F. Margot. A catalog of minimally nonideal matrices. *Math. Methods of Oper. Res.*, 47(2):221–241, 1998.

[7] P. D. Seymour R. Thomas M. Chudnovsky, N. Robertson. The strong perfect graph theorem. 2005.

Compact formulations of some mixed integer programs

Michele Conforti
(joint work with Laurence Wolsey)

Given a set of the form $P = \{(x, y) : \ Ax + By \leq d\}$, we address the question of finding a formulation for the mixed-integer set $Z^{MIP} = \{(x, y) \in P, \ y \text{ integer}\}$ associated with P.

For us, a *formulation* in the (x, y)-space is a polyhedral description of $\text{conv}(Z^{MIP})$ in the original space. It consists of a finite set of inequalities such that $\text{conv}(Z^{MIP}) = \{(x, y) : \ A'x + B'y \leq d'\}$. A formulation of Z^{MIP} is *extended* whenever it gives a polyhedral description of $\text{conv}(Z^{MIP})$ in a space that uses variables (x, y, w) and includes the original space, so that $\text{conv}(Z^{MIP})$ is the projection of this polyhedral description onto $w = 0$. An extended formulation is *compact* if the size of its polyhedral description is polynomial in the size of the description of P.

Finding an extended formulation for a mixed-integer set Z^{MIP} which is compact is important. For instance, tight formulations for relaxations allow us to strengthen the linear programming representations of hard MIPs, and theoretically a proof that a problem has a compact extended formulation implies that one can optimize a linear objective over Z^{MIP} using linear programming, and thus demonstrates that this problem is in $\mathcal{P}$.

We find extended formulations that are compact for generalizations of certain mixed-integer sets that arise as relaxations of lot-sizing and network design problems and have been studied in the last decade. For multi-item production planning problems in which these sets typically arise as single-item relaxations, these compact formulations provide an a priori strengthening of the original representation. For the mixed-integer sets that we study, the formulations in their original space are known to have exponential size, and they have only been partially characterized so far. Furthermore the convex hulls of these mixed-integer sets have an exponential number of vertices.

Given a mixed-integer set Z^{MIP}, the approach that we use here to compute an extended formulation for $\text{conv}(Z^{MIP})$ is as follows. We study the sets $V_{Z^{MIP}}$, $R_{Z^{MIP}}$ of vertices and extreme rays of $\text{conv}(Z^{MIP})$. We then find a small number of subsets $V^i \subseteq \text{conv}(Z^{MIP})$ and $R^i \subseteq R_{Z^{MIP}}$ whose union contains $V_{Z^{MIP}}$ and $R_{Z^{MIP}}$ respectively. For each of the pairs (V^i, R^i) we compute a compact formulation for $\text{conv}(V^i) + \text{cone}(R^i)$. This compact formulation will typically be an extended formulation. The last step is to derive a compact formulation which is extended for the convex hull of the union of these polyhedra. For this we use a classical result of Balas [2].

The idea of breaking the set $V_{Z^{MIP}} \cup R_{Z^{MIP}}$ into a small number of subsets has been used before: one approach found in Pochet and Wolsey [5] and developed systematically in the thesis of Van Vyve [6] is to develop an extended MIP representation for such problems explicitly including all the extreme points, and then

tighten with valid inequalities until a extended formulation is obtained; another is to generate an extended formulation based on an explicit or implicit representation of all the extreme points and rays as in Miller and Wolsey [4]. A simple example of the approach studied here has appeared very recently in Atamturk [1].

Extended formulations and the union of polyhedra

A *polyhedron* P is the intersection of a finite number of half-spaces. Equivalently, $P = \{x \in \mathbb{R}^n : Ax \geq b\}$. We use Minkowski-Weyl's theorem which asserts that a pointed polyhedron P has a finite set of vertices V_P and a finite set of extreme rays R_P and $P = \text{conv}(V_P) + \text{cone}(R_P)$. Conversely, for every pair of finite families V and R, there is a matrix $[A|b]$ such that $\{x \in \mathbb{R}^n : Ax \geq b\} = \text{conv}(V) + \text{cone}(R)$. We also use the fact that $\text{cone}(R) = \{x \in \mathbb{R}^n : Ax \geq 0\}$.

Lemma 1. *(Balas [2]) Assume $P^i = \{x \in \mathbb{R}^n : \ Q^i x \geq q^i\}$ are m polyhedra. For $1 \leq i \leq m$, let V^i, R^i be the sets of vertices and extreme rays of P^i, so $P^i = \text{conv}(V^i) + cone(R^i)$. Let*

$$P = \text{conv}(\cup_{i=1}^m V^i) + cone(\cup_{i=1}^m R^i).$$

Then the following set of inequalities provides an extended description of P:

$$\begin{gathered} x = \textstyle\sum_{i=1}^m x^i \\ Q^i x^i \geq q^i \delta^i, \ 1 \leq i \leq m \\ \textstyle\sum_{i=1}^m \delta^i = 1 \\ \delta^i \geq 0, \ 1 \leq i \leq m. \end{gathered}$$

Lemma 2. *The polyhedron P defined in Lemma 1 is the closure of the set $\text{conv}(\cup_{i=1}^m P^i)$. If all the polyhedra P^i have the same recession cone, i.e. $R^i = R^j$ for $i, j \in \{1, \ldots, m\}$, then $\text{conv}(\cup_{i=1}^m P^i)$ is a closed set and $P = \text{conv}(\cup_{i=1}^m P^i)$.*

We will need the following straightforward "extended" version of Lemma 1:

Remark 1. *If an extended formulation $\{(x, w) \in \mathbb{R}^{n+p} : \ A^i x + B^i w \geq d^i\}$ is given for each of the polyhedra $P^i = \text{conv}(V^i) + cone(R^i)$, then the following set of inequalities provides an extended formulation of the polyhedron $P = \text{conv}(\cup_{i=1}^m V^i) + cone(\cup_{i=1}^m R^i)$:*

$$\begin{gathered} x = \textstyle\sum_{i=1}^m x^i \\ A^i x + B^i w \geq d^i \delta^i, \ 1 \leq i \leq m \\ \textstyle\sum_{i=1}^m \delta^i = 1 \\ \delta^i \geq 0, \ 1 \leq i \leq m. \end{gathered}$$

Given a set $P = \{(x, y) : \ Ax + By \leq d\}$, we use the above results to obtain an (extended) formulation for the polyhedron $P_I = \text{conv}((x, y) \in P, \ y \text{ integer})$. Specifically we study the sets V_{P_I}, R_{P_I} of vertices and extreme rays of P_I. We then find subsets $V^1, \ldots, V^m$ of P_I such that $V_{P_I} \subseteq \cup_{i=1}^m V^i$. We also find sets

$R^1, \ldots, R^m$ such that $R_{P_I} = \cup_{i=1}^m R^i$ and we compute a formulation for each of the m polyhedra $P^i = \text{conv}(V^i) + \text{cone}(R^i)$.

The formulations of the polyhedra P^i in their natural space typically involve an exponential number of inequalities: For each of the polyhedra P^i, we increase the dimension of the space by adding extra variables and find an extended formulation, which is compact. We finally use Remark 1 to obtain an extended formulation for the polyhedron P_I. The number m will be small, and thus the extended formulation will be compact.

Three examples

We apply the approach to derive compact extended formulations for three sets. The first set studied is the continuous mixing set with upper bound:

$$X^{CMIXUB} = \{(s, r, y) \in \mathbb{R}^1_+ \times \mathbb{R}^m_+ \times \mathbb{Z}^m_+ : s \leq u\,; s + r_t + y_t \geq b_t \text{ for } t = 1, \ldots, m\}.$$

This set provides a relaxation motivated by the problem of treating upper bounds on stocks in lot-sizing models.

The second set we consider is the mixing set with two divisible capacities:

$$X^{2DIV} = \{(s, y, z) \in \mathbb{R}^1_+ \times \mathbb{Z}^m_+ \times \mathbb{Z}^m_+ : s + y_t + Cz_t \geq b_t,\ t = 1, \ldots, m\}$$

where $C \in \mathbb{Z}^1$ with $C \geq 2$. This set is a relaxation for lot-sizing problems in which two machines with different capacities can produce the same item, and is close to a model treating lower bounds on production studied recently by Constantino, Miller and Van Vyve [3] and Van Vyve [6].

The last set we consider is the divisible capacity single node flow model:

$$X^{FDIV} = \{(s, x, y) \in \mathbb{R}^1_+ \times \mathbb{R}^n_+ \times \mathbb{Z}^n_+ : s + \sum_{j=1}^n x_j \geq b, x_j \leq C_j y_j \text{ for } j = 1, \ldots, n\}.$$

References

[1] A. Atamtürk. Strong formulations of robust mixed 0-1 programming. Technical report, Dept. of IE and OR, Univ. of California, Berkeley, 2003, revised 2005.

[2] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. invited paper with foreword by G. Cornuéjols and W.R. Pulleyblank. *Discrete Applied Mathematics*, 89:1–44, 1998.

[3] M. Constantino, A. Miller, and M. Van Vyve. Lot-sizing with lower bounds. Technical report, University of Lisbon, 2004.

[4] A. Miller and L.A. Wolsey. Tight formulations for some simple MIPs and convex objective IPs. *Mathematical Programming B*, 98:73–88, 2003.

[5] Y. Pochet and L.A. Wolsey. Polyhedra for lot-sizing with Wagner-Whitin costs. *Mathematical Programming*, 67:297–324, 1994.

[6] M. Van Vyve. *A solution approach of production planning problems based on compact formulations for single-item lot-sizing models.* PhD thesis, Université Catholique de Louvain, 2003.

A simple GAP-canceling algorithm for generalized maximum flow

David P. Williamson
(joint work with Mateo Restrepo)

We give a simple primal algorithm for the generalized maximum flow problem that repeatedly finds and cancels generalized augmenting paths (GAPs). We use ideas of Wallacher [8] to find GAPs that have a good trade-off between the gain of the GAP and the residual capacity of its arcs; our algorithm may be viewed as a special case of Wayne's algorithm for the generalized minimum-cost circulation problem [9].

Most previous algorithms for the generalized maximum flow problem are dual-based; previous primal algorithms of Cohen and Megiddo [3] and Wayne [9] require subroutines to test the feasibility of linear programs with two variables per inequality (TVPIs). Drawing on work of Aspvall [1], we give an $O(mn)$ time and $O(n)$ space algorithm for finding negative-cost GAPs which can be used in place of the TVPI tester. This algorithm is analogous to the negative-cost cycle detection algorithm used for many minimum-cost flow routines with some interesting and important differences.

Let m denote the number of arcs in the graph, and let n the number of nodes. Let the capacity of the arcs be integers bounded by B, and let the gains be ratios of integers, where the integers are also bounded by B. Our approach yields an overall algorithm for the generalized maximum flow problem with $O(m \log(mB/\varepsilon))$ iterations of $O(mn)$ time to compute an flow with value at least $(1-\varepsilon)$ times the optimal, or $O(m^2 \log(mB))$ iterations to compute an optimal flow, for an overall running time of $O(m^3 n \log(mB))$.

The first polynomial-time combinatorial algorithms for the problem are due to Goldberg, Plotkin, and Tardos [4]. The fastest known running time for this problem is $\tilde{O}(m^2 n \log B)$, and is due to Radzik [7], building on earlier work of Goldfarb, Jin, and Orlin [5].

We have also performed a preliminary implementation of this algorithm, using some ideas of Cherkassky and Goldberg [2] for negative-cost cycle detection. However, our tests show that it is not currently competitive with good linear program solvers such as CPLEX [6].

References

[1] B. I. Aspvall. *Efficient Algorithms for Certain Satisfiability and Linear Programming Problems*. PhD thesis, Department of Computer Science, Stanford University, Aug. 1980. Also appears as Technical Report STAN-CS-80-822.

[2] B. V. Cherkassky and A. V. Goldberg. Negative-cycle detection algorithms. *Mathematical Programming*, 85:277–311, 1999.

[3] E. Cohen and N. Megiddo. New algorithms for generalized network flows. *Mathematical Programming*, 64:325–336, 1994.

[4] A. V. Goldberg, S. A. Plotkin, and É. Tardos. Combinatorial algorithms for the generalized circulation problem. *Mathematics of Operations Research*, 16:351–381, 1991.

[5] D. Goldfarb, Z. Jin, and J. B. Orlin. Polynomial-time highest-gain augmenting path algorithms for the generalized circulation problem. *Mathematics of Operations Research*, 22:793–802, 1997.
[6] ILOG. *CPLEX 9.0*, 2003.
[7] T. Radzik. Improving time bounds on maximum generalised flow computations by contracting the network. *Theoretical Computer Science*, 312:75–97, 2004.
[8] C. Wallacher. A generalization of the minimum-mean cycle selection rule in cycle canceling algorithms. Technical report, Institut für Angewandte Mathematik, Technische Universität Carolo-Wilhelmina, Nov. 1991.
[9] K. D. Wayne. A polynomial combinatorial algorithm for generalized minimum cost flow. *Mathematics of Operations Research*, 27:445–459, 2002.

Finding *all* optimal solutions for submodular function minimization

S. Thomas McCormick
(joint work with Satoru Fujishige and Maurice Queyranne)

Given a submodular function $f : 2^E \to \mathbb{R}$ on a finite ground set E, we consider the problem of finding all optimal solutions to the *Submodular Function Minimization (SFM)* problem:

$$\min_{S \subseteq E} f(S).$$

Solving SFM was an important open problem in combinatorial optimization until Grötschel, Lovász, and Schrijver [8, 9] solved it in 1981 using the Ellipsoid Algorithm. Because Ellipsoid is impractical, it remained an open problem to find a *combinatorial* (non-Ellipsoid) SFM algorithm until two nearly simultaneous papers in 1999 by Schrijver [15] (formally published in 2000) and Iwata, Fleischer, and Fujishige (IFF) [11] (formally published in 2001).

Both of these results were based on a framework set up by Cunningham [4], which was in turn based on an LP duality for SFM by Edmonds [5]. A useful way to view Cunningham's framework is that it makes SFM look like a generalization of Min Cut (which is a special case of SFM) with a corresponding primal problem that looks like Max Flow.

For Min Cut, Picard and Queyranne [14] showed that the submodularity of the s–t cut function implies that the set of all min cuts is closed under union and intersection, and hence forms a lattice, or *ring family*. They further showed how to use any primal optimal solution (max flow) to compute a directed graph on the nodes such that a node subset S is a min cut if and only if it corresponds to an ideal of the partial order formed by the strongly connected components of the directed graph. This is quite nice as this partial order can be computed in only $O(m)$ time from a max flow, and it represents the family of all min cuts (which can be exponentially large) in $O(m)$ space.

There are many practical applications where having access to all min cuts is helpful; Picard and Queyranne survey many of these. There are similarly many applications of SFM where having access to all optimal solutions would be helpful.

For example, Martin Cooper [3] considers the problem from an Artificial Intelligence perspective (and [12, Theorem 3.5] shows that $O(n^2)$ SFM calls suffice to compute all SFM solutions); Shen, Coullard and Daskin [17] consider a facility location problem where knowing all SFM solutions would help in satisfying real-world constraints; Huh and Roundy [10] use parametric SFM to determine an optimal replacement sequence for semiconductor fabrication tools, and again having all SFM solutions would help in satisfying real-world constraints; and Baumann and Skutella [1] also use parametric SFM to compute flows over time. See Fleischer [6], Fujishige [7, Chapter VI], McCormick [13], or Schrijver [16, Chapter 45] for surveys of SFM containing other applications.

It is well-known that the property that the set of optimal solutions forming a ring family extends from Min Cut to SFM. We extend Picard and Queyranne's result from Min Cut to SFM using three tools: (1) Edmonds' duality result for SFM, and its associated complementary slackness; (2) the fact that the combinatorial SFM algorithms represent their primal solution as a convex combination of greedy vertices; and (3) an algorithm of Bixby, Cunningham, and Topkis (BCT) [2] for finding the tight sets for a greedy vertex.

Define the *base polyhedron* of f as $B(f) = \{y \in \mathbb{R}^E \mid y(S) \leq f(S)\ \forall S \subset E \text{ and } y(E) = f(E)\}$, and $y_e^- = \min(y_e, 0)$. Then the gist of Edmonds' duality result is the min-max relation that $\min_{S \subseteq E} f(S) = \max_{y \in B(f)} y^-(E)$. The combinatorial SFM algorithms maintain a primal $y \in B(f)$, represented as $y = \sum_i \lambda_i v^i$ for $\sum_i \lambda_i = 1$ and $\lambda_i \geq 0$, where each v^i is a vertex generated by the Greedy Algorithm. By Carathéodory's Theorem the SFM algorithms are able to update this representation such that it has only $O(n)$ terms.

Weak duality for Edmonds' LP says that for any primal $y \in B(f)$ and any dual $S \subseteq E$, $y^-(E) \leq y^-(S) \leq y(S) \leq f(S)$. Complementary slackness says that y and S are jointly optimal if and only if (1) $y_e < 0 \implies e \in S$ ($\iff y^-(E) = y^-(S)$); (2) $e \in S \implies y_e \leq 0$ ($\iff y^-(S) = y(S)$); and (3) S is *tight* ($\iff y(S) = f(S)$). Given a primal optimal y, define index sets $M = \{e \in E \mid y_e < 0\}$ and $Z = \{e \in E \mid y_e = 0\}$. Then to find all optimal SFM solutions, all we need to do is to find all tight sets for y, and then all tight sets T such that $M \subseteq T \subseteq M \cup Z$ are precisely all optimal SFM solutions.

Given the representation $y = \sum_i \lambda_i v^i$, note that T is tight for $y \iff T$ is tight for each v^i. BCT [2] give an algorithm for finding the lattice of tight sets for a vertex v^i that requires $O(n^2)$ function evaluations, and which produces a directed graph D^i such that T is tight if and only if it corresponds to an ideal of the partial order induced by the strongly connected components of D^i.

Therefore, if we take the graph D whose arc set is the union of the $O(n)$ arc sets of the D^i, then the tight sets for y are precisely the ideals of the partial order induced by the strongly connected components of D. Computing D takes only $O(n^3)$ extra function evaluations. The material up to this point is already contained in the Remark at the end of [7, Section 14.2].

Schrijver's Algorithm produces an exact optimal primal point y, but the various IFF Algorithms do not. It is possible to use $O(n)$ calls to an IFF algorithm to

find an exact primal optimal point, though it is not clear how to also produce its required representation as a convex combination of vertices.

However, we note that the strongly polynomial IFF algorithms do carry enough information to get all optimal solutions. As they proceed, they use some variation of a "proximity lemma" to develop sets IN and OUT such that $e \in \text{IN} \implies e$ belongs to *every* SFM optimal solution, and $e \in \text{OUT} \implies e$ belongs to *no* SFM optimal solution, and they work on the reduced ground set $R \equiv E - \text{IN} - \text{OUT}$. They use a scale factor δ and recognize optimality when $\delta \leq 0$, and which point they know that their current primal $y \in \mathbb{R}^R$ satisfies $y \leq 0$. This implies that $y^-(R) = y(R) = \tilde{f}(R)$, where $\tilde{f}$ is the appropriately modified f on R (note that this also implies that R is the unique largest SFM solution). They also have a convex combination representation of y. Therefore we can apply the above algorithm to find all tight sets w.r.t. y, which then gives us all SFM solutions to the original problem.

This leaves an open problem. It is easy to see that in fact there always exists an *integral* primal optimal solution y, but apparently none of the existing SFM algorithms can directly find one. It is possible to use $O(n)$ calls to SFM to find an integral optimal primal point, but it is not clear how to keep the required convex combination representation of it. Thus it would be interesting to find an SFM algorithm that can find an integral primal optimal point together with a representation of it as a convex combination of vertices.

References

[1] N. Baumann and M. Skutella (2005). Computing Earliest Arrival Flows with Multiple Sources. Talk at Oberwolfach Workshop on Combinatorial Optimization, 8 November 2005.

[2] R. E. Bixby, W. H. Cunningham, and D. M. Topkis (1985). The Partial Order of a Polymatroid Extreme Point. *Math. of OR*, **10**, 367–378.

[3] M. C. Cooper (2005). Email message to Maurice Queyranne 22 June

[4] W. H. Cunningham (1985). On Submodular Function Minimization. *Combinatorica*, **3**, 185–192.

[5] J. Edmonds (1970). Submodular Functions, Matroids, and Certain Polyhedra. In *Combinatorial Structures and their Applications*, R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds., Gordon and Breach, 69–87.

[6] L. K. Fleischer (2000). Recent Progress in Submodular Function Minimization. *Optima*, September 2000, 1–11.

[7] S. Fujishige (2005). *Submodular Functions and Optimization: Second Edition*. Elsevier.

[8] M. Grötschel, L. Lovász, and A. Schrijver (1981). The Ellipsoid Algorithm and its Consequences in Combinatorial Optimization. *Combinatorica*, **1**, 499–513.

[9] M. Grötschel, L. Lovász, and A. Schrijver (1988). *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag.

[10] W. T. Huh and R. O. Roundy (2002). A Continuous-Time Strategic Capacity Planning Model. Working paper, SORIE, Cornell University, submitted to *Operations Research*.

[11] S. Iwata, L. Fleischer, and S. Fujishige (2001). A Combinatorial, Strongly Polynomial-Time Algorithm for Minimizing Submodular Functions. *J. ACM*, **48**, 761–777.

[12] P. Jeavons, D. Cohen, and M. C. Cooper (1998). Constraints, Consistency and Closure. *Artificial Intelligence*, **101** 251–265.

[13] S. T. McCormick (2005). Submodular Function Minimization. A chapter in the forthcoming *Handbook on Discrete Optimization*, Elsevier, K. Aardal, G. Nemhauser, and R. Weismantel, eds.
[14] J-C. Picard and M. N. Queyranne (1980). On the Structure of All Minimum Cuts in a Network and Applications. *Math. Prog. Study*, **13**, 8–16.
[15] A. Schrijver (2000). A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time. *J. Combin. Theory Ser. B* **80**, 346–355.
[16] A. Schrijver (2003). *Combinatorial Optimization: Polyhedra and Efficiency.* Springer, Berlin.
[17] Z-J. M. Shen, C. Coullard, and M. S. Daskin (2003). A Joint Location-Inventory Model. *Transportation Science*, **37**, 40–55.

Valid inequalities for mixed integer linear programs

Gérard Cornuéjols

Consider first a mixed 0,1 linear program in the form

$$\begin{array}{lrll} \min & cx & & \\ & Ax & \geq & b \\ & x_j & = & 0,1 \quad \text{for } j = 1,\dots,n \\ & x_j & \geq & 0 \quad \text{for } j = n+1,\dots,n+p, \end{array}$$

where the matrix $A \in \mathbb{Q}^{m\times(n+p)}$, the row vector $c \in \mathbb{Q}^{n+p}$ and the column vector $b \in \mathbb{Q}^m$ are data, and $x \in \mathbb{Q}^{n+p}$ is a column vector of variables. We assume that the constraints $Ax \geq b$ include $x_j \geq 0$ for $j = 1,\dots,n+p$, and $x_j \leq 1$ for $j = 1,\dots,n$.

Consider the polyhedron $P := \{x \in \mathbb{R}^{n+p}_+ : \ Ax \geq b\}$ and the mixed 0,1 linear set $S := \{x \in \{0,1\}^n \times \mathbb{R}^p_+ : \ Ax \geq b\}$. The set $\text{conv}(S)$ is a polyhedron and, ideally, we would like to have its linear description in the form $\text{conv}(S) = Dx \geq d$. Constructing the description $Dx \geq d$ would reduce the solution of a mixed 0,1 linear program to that of a linear program. But this goal is too ambitious in general as the number of inequalities needed in the description $Dx \geq d$ is typically enormous. This is not surprising considering that mixed 0,1 linear programming is NP-hard! A more reasonable goal is to obtain an intermediate set between P and $\text{conv}(S)$ over which one can optimize a linear function in polynomial time, and then to use recursion to get tighter approximations of $\text{conv}(S)$.

Sherali and Adams [8], Lovász and Schrijver [7] and Balas, Ceria and Cornuéjols [2] propose an approach for doing this which generates intermediate sets Q between P and $\text{conv}(S)$ as projections of higher dimensional sets that have a polynomial description. The polyhedron $P \subseteq \mathbb{R}^{n+p}$ is first lifted into a higher dimensional space $\mathbb{R}^{n+p+q}$ where the formulation is strengthened. This strengthened formulation is then projected back onto the original space $\mathbb{R}^{n+p}$, thus defining Q. In this process the constraints of the higher dimensional formulation are defined explicitly whereas those of Q are only known implicitly through projection, thus allowing Q to have a nonpolynomial number of constraints. This approach is known under the name of *lift-and-project*. The version of the lift-and-project procedure proposed by Balas, Ceria and Cornuéjols [2] is as follows:

Step 0: Select $j \in \{1, \ldots, n\}$.
Step 1: Generate the nonlinear system $x_j(Ax - b) \geq 0$ and $(1 - x_j)(Ax - b) \geq 0$.
Step 2: Linearize the system by substituting y_i for $x_i x_j$, $i \neq j$, and x_j for x_j^2. Call this polyhedron M_j.
Step 3: Project M_j onto the x-space. Let P_j be the resulting polyhedron.

The set $\cap_{j=1}^n P_j$ is called the *lift-and-project closure* of P. The lift-and-project closure is a better approximation of $\text{conv}(S)$ than P:

$$\text{conv}(S) \subseteq \cap_{j=1}^n P_j \subseteq P.$$

How much better is it in practice? Bonami and Minoux [5] performed such an experiment. On instances from the MIPLIB library [4], they found that the lift-and-project closure reduces the integrality gap by 37% on average (the *integrality gap* is the difference between the objective value optimized over $\text{conv}(S)$ and the relaxation P).

One can obtain a stronger relaxation (Sherali-Adams [8]) by skipping Step 0 and considering the nonlinear constraints $x_j(Ax - b) \geq 0$ and $(1 - x_j)(Ax - b) \geq 0$ for all $j = 1, \ldots, n$ in Step 1. Then, in Step 2, variables y_{ij} are introduced for all $i = 1, \ldots, n+p$ and $j = 1, \ldots, n$ with $i > j$. Note that the size of the linear system generated in Step 2 is much larger than in the previous procedure ($\frac{n(n-1)}{2} + np$ new variables and $2nm$ constraints, instead of just $n + p - 1$ new variables and $2m$ constraints before). The Sherali-Adams relaxation is at least as strong as the lift-and-project closure defined above, and it can be strictly stronger since the Sherali-Adams procedure takes advantage of the fact that $y_{ij} = y_{ji}$ whereas this is not the case for the lift-and-project closure $\cap_{j=1}^n P_j$. How much better is the Sherali-Adams relaxation in practice? Surprisingly, on the MIPLIB instances, Bonami and Minoux [5] found that it only provides a marginal improvement over the lift-and-project bound (38.5 % versus 37 %). Lovász and Schrijver [7] proposed an even stronger version of the lift-and-project procedure. It would be interesting to investigate its strength on the MIPLIB instances.

Next we consider mixed integer linear sets where the variables x_j can take general nonnegative integer values, instead of just 0 or 1. Let

$$S = \{x \in \mathbb{Z}_+^n, y \in \mathbb{R}_+^p : \sum_{j=1}^n a_j x_j + \sum_{j=1}^p g_j y_j = b\}$$

Let $b = \lfloor b \rfloor + f_0$ where $0 < f_0 < 1$.
Let $a_j = \lfloor a_j \rfloor + f_j$ where $0 \leq f_j < 1$.
Gomory showed that the following inequality (called *Gomory mixed integer cut* or *GMI* cut) is valid for S.

$$\sum_{f_j \leq f_0} \frac{f_j}{f_0} x_j + \sum_{f_j > f_0} \frac{1 - f_j}{1 - f_0} x_j + \sum_{g_j > 0} \frac{g_j}{f_0} y_j - \sum_{g_j < 0} \frac{g_j}{1 - f_0} y_j \geq 1. \quad (1)$$

Gomory suggest to generate GMI cuts (1) from equations arising in the optimal tableau of the linear programming relaxation. Observe that, if we multiply $\sum_{j=1}^{n} a_j x_j + \sum_{j=1}^{p} g_j y_j = b$ by an integer k, the coefficients of the x_j variables in (1) remain between 0 and 1, but the coefficients of the y_j variables are multiplied by the factor k, which makes the cut weaker. This suggests the following idea (Andersen, Cornuéjols and Li [1]): Take linear combinations of the constraints in order to produce equations $\sum_{j=1}^{n} a_j x_j + \sum_{j=1}^{p} g_j y_j = b$ where the norm of the vector g is small. And generate (1) from these "reduced" equations. These cuts are called *reduce-and-split* cuts. Computational experiments show that the reduce-and-split cuts are usually very different from the GMI cuts generated from the rows of the optimal tableau and that, in some cases, they can be significantly stronger. To illustrate this, we present a few instances of the MIPLIB where the improvement was particularly striking. The gap closed is reported after 20 rounds of cuts. The last two columns give the number of nodes in a branch-and-bound algorithm using these cuts in the formulation. In other instances, the reduce-and-split cuts did not improve on the GMI cuts from the optimal tableau. Therefore it seems that a hybrid approach that uses both types of cuts is a reasonable strategy.

Name	GMI gap	R&S gap	GMI nodes	R&S nodes
flugpl	14%	100%	184	0
gesa2	46%	97%	743	116
gesa2o	92%	98%	9145	75
mod008	47%	88%	1409	82
pp08a	83%	92%	7467	745
rgn	15%	100%	874	0
vpm1	44%	98%	7132	1
vpm2	41%	61%	38946	4254

References

[1] K. Andersen, G. Cornuéjols and Y. Li, Reduce-and-split cuts: Improving the performance of mixed integer Gomory cuts, *Management Science 51* (2005) 1720-1732.

[2] E. Balas, S. Ceria and G. Cornuéjols, A lift-and-project cutting plane algorithm for mixed 0-1 programs, *Mathematical Programming 58* (1993) 295–324.

[3] E. Balas and R. Jeroslow, Strengthening cuts for mixed integer programs, *European Journal of Operations Research 4* (1980) 224-234.

[4] R.E. Bixby, S. Ceria, C.M. McZeal and M.W.P. Savelsbergh, An updated mixed integer programming library: MIPLIB 3.0, *Optima 58* (1998) 12-15.

[5] P. Bonami and M. Minoux, Using rank-1 lift-and-project closures to generate cuts for 0-1 MIPs, a computational investigation, to appear in *Discrete Optimization*.

[6] R.E. Gomory, An algorithm for integer solutions to linear programs, *Recent Advances in Mathematical Programming*, R.L. Graves and P. Wolfe eds., McGraw-Hill, New York (1963) 269–302.

[7] L. Lovász and A. Schrijver, Cones of matrices and set-functions and 0-1 optimization, *SIAM Journal of Optimization 1* (1991) 166–190.

[8] H. Sherali and W. Adams, A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems, *SIAM Journal on Discrete Mathematics 3* (1990) 311–430.

Geometry and expansion: A survey of recent results

Sanjeev Arora

Graph expansion occurs as a unifying concept across several areas of theoretical computer science, including theory of communication networks, theory of error-correcting codes, theory of approximation algorithms, and theory of computational pseudo-randomness. This brief survey concerns new, geometric ways of looking at expansion that have engendered new breakthroughs in approximation algorithms, geometric embeddings of metric spaces, and probabilistically checkable proofs.

In approximation algorithms the breakthrough is new $O(\sqrt{\log n})$-approximation algorithms for a host of NP-hard optimization problems, starting with the discovery of such an algorithm for SPARSEST CUT in [ARV]. These new algorithms rely on a new analysis of a family of semidefinite programs.

In geometric embeddings new results include an almost-tight embedding of ℓ_1-spaces into ℓ_2 with distortion $O(\sqrt{\log n} \log\log n)$. There have also been a spate of results ruling out certain types of embeddings, most notably a paper of Khot and Vishnoi which rules out $O(1)$-distortion embedding of ℓ_2^2 into ℓ_1.

Constructions of PCPs in recent years have relied upon theorems in Fourier Analysis which are also geometric in nature, and this has also become clearer thanks to the results on embeddings.

Yet another connection between geometry and expansion is that the above results rely upon a geometric analog of the study of expansion, namely, *isoperimetric problems*. The simplest is the classical result that every closed set in $\Re^2$ whose area is A has perimeter at least $2\sqrt{\pi A}$, the perimeter of the circle of area A. One can in fact prove the stronger statement that if this set has perimeter "close to" $2\sqrt{\pi A}$, then it "looks like" a circle of area A. The latter type of theorems we be referred to as *Strong Isoperimetric Theorems*. Isoperimetric theorems about the n-dimensional sphere and the boolean hypercube play an important role in the above results.

References

[AHK05] S. Arora, E. Hazan, and S. Kale. $O(\sqrt{\log n})$ approximation to Sparsest Cut in $\tilde{O}(n^2)$ time. *IEEE Foundations of Computer Science, 2004.*

[ALN] Sanjeev Arora, James Lee, and Assaf Naor. Euclidean distortion and the sparsest cut. *ACM STOC 2005.*

[ARV] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings, and graph partitioning. In *ACM STOC 2004*, pages 222–231.

Rearrangeably nonblocking multicast switching networks

Alexander Martin
(joint work with Peter Lietz)

Switching networks are directed acyclic graphs with specified disjoint sets of input and output nodes. A connection request is a partial function from the set of output nodes to the set of input nodes specifying which input node needs to be routed to which output nodes. A switching network is called rearrangeably nonblocking with respect to *multicast* traffic if all connection requests are routable, that is, if for each request there exists a set of mutually vertex-disjoint directed trees connecting each input node to its designated output nodes. A switching network is called rearrangeably nonblocking with respect to *unicast* traffic if all *one-one* connection requests are routable, that is, if for each such request there exists a set of mutually vertex-disjoint directed paths connecting each input node to its designated output node. In the following, we will use the shorthand terms multicast-rearrangeable and unicast-rearrangeable.

Clos networks have been introduced by Charles Clos in [1] in order to reduce network size and cost. A Clos network is composed from switches, which are arranged in three stages: the input stage, the center stage and the output stage. The switches within each stage have the same dimension. Every pair of switches in consecutive stages is connected via exactly one link. See [3] for an exhaustive treatment of Clos networks.

A Clos network is completely described by the following set of parameters. We denote by n_1 the number of inputs of each input stage switch and by r_1 the number of input stage switches. By m we denote the number of center stage switches. Lastly, by n_2 we denote the number of outputs of each output stage switch and by r_2 the number of output stage switches. A Clos network is called symmetric if $n_1 = n_2$ and $r_1 = r_2$. In this case we denote these parameters by n and r. Finally, the number of inputs is denoted by $N_1 = n_1 \cdot r_1$ and the number of outputs is denoted by $N_2 = n_2 \cdot r_2$. Figure 1 shows a symmetric Clos network with $n = 4$, $r = 4$ and $m = 6$.

The Slepian-Duguid Theorem states that a symmetric Clos network is unicast-rearrangeable if and only if $m \geq n$. While the Slepian-Duguid theorem is a straightforward application of König's edge coloring theorem, matters become much more involved in the multicast case. The problem of characterizing multicast-rearrangeable Clos networks has been partially addressed in [2, 6], however, as stated in [4], no necessary and sufficient conditions on n, r and m are known such that the associated Clos network is multicast-rearrangeable. As for fixed n and r, the number of crosspoints of a Clos network grows linearly with m, and as applications of multicast connection networks abound, we consider it crucial to work towards filling this gap.

For a fixed Clos network, we represent each multicast connection request as a binary $r_2 \times r_1 \cdot n_1$ matrix such that the sum of each row is at most n_2. The matrix entry (i, j) is set to one if and only if input j is requested by output switch i.

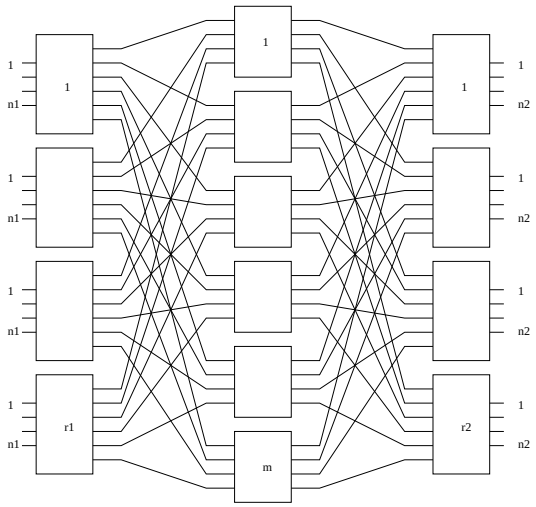


FIGURE 1. Clos network

The routability of a connection request translates into the following property of the associated matrix. A request is routable if and only if one can assign to each nonzero in the associated matrix a number out of $\{0, \ldots, m-1\}$ such that

(1) each number is used at most once in each row,
(2) within each block, each number is used in at most one column,

where by the k-th block we mean the submatrix consisting of the columns $k \cdot n_1$ up to $(k+1) \cdot n_1 - 1$, for $k = 0, \ldots, r_1 - 1$.

The above described property can be naturally presented as a vertex-coloring or higher dimensional matching problem. We will henceforth identify a multicast request with its associated matrix. In order to verify that some given Clos network is multicast-rearrangeable, we have to verify that each request has the above property. As the number of possible requests is prohibitively large, we have to single out a smaller class of requests such that the routability of this class of requests guarantees the routability of all possible requests. We will do so using essentially three ideas.

(1) We do not consider requests whose routability can be established by some known theorem.
(2) We do not consider requests whose routability can be reduced to the routability of some request which is strictly harder to satisfy.
(3) Out of each symmetry class of requests, we only consider one representative.

As an example for (1), if each block and each row contains at most m nonzeros, then the routability follows from König's edge coloring theorem. As for (2), if the columns of two different blocks can be merged into one block in such a way that the nonzeros do not collide, we can discard the verification of the request, as the transformed request is provably harder to satisfy. Finally, if two requests are equal up to permutation of rows, permutation of columns within a block and permutation

of entire blocks, then the routability of one request entails the routability of the other, hence we will only consider the lexicographically minimal representative of each symmetry class.

By efficiently enumerating and solving a moderately sized class of requests, we were able to establish the fact that the symmetric Clos network with $n = 8$ and $r = 4$ is multicast-rearrangeable if and only if $m \geq 11$, see [5]. The resulting Clos network has 880 crosspoints. In comparison, the trivial 32×32 switching network has 1024 crosspoints. We are optimistic that our method is applicable to networks with considerably larger dimensions as well, in which case we would expect proportionally even larger savings than in the 32×32 case.

References

[1] C. Clos, *A study of nonblocking switching networks*, Bell Syst. Tech. J., **32** (1953), 406–424.

[2] F. K. Hwang, *Rearrangeability of multiconnection three-stage networks*, Networks **2** (1972), 301–306.

[3] F. K. Hwang, *The mathematical theory of nonblocking switching networks*, World Scientific, Singapore, (1998).

[4] F. K. Hwang, *A survey of nonblocking multicast three-stage clos networks*, IEEE Commun. Mag., Oct. (2003), 34–37.

[5] P. Lietz, A. Martin *Rearrangeably nonblocking multicast Clos networks*, Preprint, Department of Mathematics, Darmstadt University of Technology, (2006), in preperation.

[6] G. M. Masson, B. W. Jordan, *Generalized multi-stage connection networks*, Networks **2** (1972), 191–209.

An exact approach to crossing minimization

Petra Mutzel

(joint work with Christoph Buchheim, Markus Chimani, Dietmar Ebner, Carsten Gutwenger, Michael Jünger, Gunnar Klau, René Weiskircher)

Crossing minimization is among the oldest and most fundamental problems in the areas of automatic graph drawing and *VLSI*design. At the same time, it is very easy to formulate: "*Given a graph $G = (V, E)$, draw it in the plane with a minimum number of edge crossings*". A drawing of G is a mapping of each vertex $v \in V$ to a distinct point and each edge $e = (v, w) \in E$ to a curve connecting the incident vertices v and w without passing through any other vertex. Common points of two edges that are not incident vertices are called *crossings*. The minimum number of crossings among all drawings of G is denoted by $\mathrm{cr}(G)$.

It is well known that the general crossing minimization problem is *NP*-hard [5]. The problem has been studied extensively in the literature from a theoretic point of view and many bounds exist for a variety of graph classes (see, e.g., [7, 10, 1, 9, 2, 6]. However, so far no practical algorithm which is able to solve even small instances to provable optimality has been known. Recently, we have presented the first algorithm that is able to compute the crossing number of general sparse graphs of moderate size [3]. The approach is based on integer linear programming techniques.

The natural ILP-approach would be to introduce a 0/1-variable for each edge pair $e, f \in E$ which is one if the two edges cross in a crossing minimal drawing and zero otherwise. We could obtain inequalities arising from Kuratowski's theorem that force the crossing of at least two edges in every Kuratowski subdivision, i.e., a subdivision of K_5 or $K_{3,3}$. Unfortunately, even deciding whether there is a drawing for a given set of edge crossings is *NP*-complete [8]. This problem is known as the *realizability problem* and can be stated as follows: *"Given a set of edge pairs D, does there exist a drawing of G such that two edges $e, f \in E$ cross each other if and only if $\{e, f\} \in D$?*

One way to work around the realizability problem is the reduction to *simple drawings*. A drawing is called simple if each edge crosses at most one other edge. Not surprisingly, there are graphs that do not admit any simple drawing, e.g., graphs with more than $4|V| - 6$ edges. Furthermore, Bodlaender and Grigoriev proved that it is *NP*-complete to determine whether there is a simple drawing for a given graph G [2]. If there is such a drawing, we denote the minimum number of crossings among all simple drawings of G by $\mathrm{crs}(G)$.

Even if there is a simple drawing for G, its crossing number $\mathrm{crs}(G)$ does not necessarily coincide with $\mathrm{cr}(G)$. However, given a graph $G = (V, E)$ we can create a graph $G^* = (V^*, E^*)$ by replacing every edge $e \in E$ with a path of length $|E|$. It is easy to show that for any non-negative number K the graph G can be drawn with K crossings if and only if there is a *simple drawing* of G^* with K crossings.

Therefore, it is "sufficient" to solve the crossing minimization problem restricted to simple drawings in order to solve the general crossing minimization problem. Since the transformation can obviously be done in polynomial time, the *NP*-completeness of the corresponding decision problem for simple drawings follows immediately from the *NP*-completeness for the general crossing number problem [5].

We present an integer linear programming formulation for the crossing minimization problem restricted to simple drawings. Let $G = (V, E)$ be a graph and let D be a set of unordered pairs of edges of G. We call D *simple* if for every $e \in E$ there is at most one $f \in E$ such that $(e, f) \in D$. Furthermore, D is called *realizable* if there is a drawing of G such that there is a crossing between edges e and f if and only if $(e, f) \in D$.

For every graph G and every simple D, we denote by G_D the graph that is obtained by introducing a dummy node $d_{e,f}$ for each pair of edges $(e, f) \in D$. More precisely, we introduce dummy nodes on both e and f and identify them. Notice that G_D is only well-defined if D is simple, as otherwise it would not be clear where to place the dummy nodes. For both edges e_1 and e_2 resulting from splitting e, we set $\hat{e}_1 = \hat{e}_2 = e$, analogously for f. It is not hard to see that if D is simple, then D is realizable if and only if G_D is planar. Using a linear time planarity testing and embedding algorithm, we can thus test in time $O(|V| + |D|)$ whether D is realizable, and compute a realizing drawing in the affirmative case.

For a set of pairs of edges $D \subseteq E^2$ we define

$$x_{e,f}^D = \begin{cases} 1 & \text{if } (e,f) \in D \\ 0 & \text{otherwise .} \end{cases}$$

Next, for every subgraph $H = (V', E')$ of G_D, let $\hat{H} = \{\hat{e} \mid e \in E'\} \subseteq E$. Less formally, $\hat{H}$ contains all edges of G involved in the subgraph H of G_D.

We can show that the following ILP-formulation is correct. We use $x(F)$ as an abbreviation for the term $\sum_{(e,f)\in F} x_{e,f}$:

$$\begin{array}{rll} \min & x(E^2) & \\ \text{s.t. } \sum_{f\in E} x_{e,f} & \leq 1 & \forall e \in E \\ x(\hat{H}^2 \setminus D) - x(\hat{H}^2 \cap D) & \geq 1 - |\hat{H}^2 \cap D| & \text{for every simple } D \\ & & \text{and every Kuratowski} \\ & & \text{subgraph } H \text{ in } G_D \\ x_{e,f} & \in \{0,1\} & \forall\ e, f \in E \end{array}$$

It is clearly impractical to generate all "Kuratowski" constraints in advance and solve the ILP in a single step. Instead, we embed the given formulation into a branch-and-cut framework, separating violated inequalities dynamically during runtime.

A crucial factor in this approach is the *separation problem*: "Given a class of valid inequalities and a vector $y \in \mathbb{R}^n$, either prove that y satisfies all inequalities in the class, or find an inequality which is violated by y.". Although we can easily separate violated inequalities for integral solution vectors (based on planarity testing), the problem is more complex within the branch-and-cut framework since we have to deal with fractional values. So far, no polynomial time algorithm for the separation problem is known.

We have implemented a branch-and-cut algorithm based on our new ILP-formulation using CPLEX combined with column generation, new preprocessing techniques, and heuristics for the separation problem and upper bounds. We used a common benchmark set in graph drawing consisting of 11,389 graphs with 10 to 100 vertices and 9 to 158 edges (collected by the University of Rome III, see [4].) We have been able to compute the exact crossing number for about 90 percent of all graphs on up to 50 vertices in the benchmark set within a time limit of 30 minutes per graph. We have also been able to solve the crossing number for the complete graph on 8 vertices K_8 to provable optimality.

This is ongoing work. Our future plans are to study the polyhedron defined by the set of realizable crossing vectors. We expect to be able to solve larger instances by tightening the ILP-formulation using additional constraints. One possibility to get such constraints could be to study k-crossing critical graphs.

References

[1] S. N. Bhatt and F. T. Leighton. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences*, 28:300–343, 1984.

[2] H. Bodlaender and A. Grigoriev. Algorithms for graphs embeddable with few crossings per edge. Research Memoranda 036, Maastricht: METEOR, Maastricht Research School of Economics of Technology and Organization, 2004.
[3] C. Buchheim, M. Chimani, D. Ebner, C. Gutwenger, M. Jünger, G. Klau, P. Mutzel and R. Weiskircher. Exact Crossing Minimization. In P. Eades and P. Healy, editors, *GD 2005: Proceedings of the 13th International Symposium on Graph Drawing*, *LNCS*. Springer-Verlag, to appear 2005.
[4] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of four graph drawing algorithms. *Computational Geometry: Theory and Applications*, 7(5-6):303–325, 1997.
[5] M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 4:312–316, 1983.
[6] M. Grohe. Computing crossing numbers in quadratic time. In *STOC 2001: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, 2001.
[7] R. K. Guy. Crossing numbers of graphs. In *Graph Theory and Applications (Proceedings)*, Lecture Notes in Mathematics, pages 111–124. Springer-Verlag, 1972.
[8] J. Kratochvíl. String graphs. II.: Recognizing string graphs is NP-hard. *Journal of Combinatorial Theory, Series B*, 52(1):67–78, 1991.
[9] J. Pach and G. Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997.
[10] P. Turán. A note of welcome. *Journal of Graph Theory*, 1:7–9, 1977.

Exact solution of combinatorial optimization problems without known decent ILP formulations

Alberto Caprara

We illustrate our experience on some NP-hard combinatorial optimization problems that share the following characteristics: (1) they arise in real-world applications, (2) they are very simple to state, (3) they do not seem to admit any decent ILP formulation, i.e., all the formulations that were tried do not allow the solution of toy instances by modern, general-purpose ILP solvers. The list of problems includes Sorting by Reversals [1], Reversal Median [2], Bandwidth [3], Linear Arrangement [5], Bidimensional Bin Packing [4], Protein Folding in the H-P Model [6].

Focusing attention on the optimal solution of (real-world) medium-size instances, we briefly illustrate what could be achieved for each of these problems, ranging from basically nothing to effective solution by strong ILP formulations of suitable relaxations, which are defined after a careful analysis of the combinatorial structure of the problem.

Specifically, for Sorting by Reversals the most effective approach is based on the definition of a non-trivial combinatorial relaxation, for which a natural ILP formulation can be derived, and the solution of this ILP. For Reversal Median, an analogous combinatorial relaxation and associated ILP formulation lead to an LP relaxation that is too cumbersome to be solved in practice, and the best thing is to resort to a combinatorial branch-and-bound algorithm based on the combinatorial relaxation. For Bandwidth, the best approach computes bounds by solving suitable (easy) ILP relaxations, keeping the integrality constraints (whose removal would

lead to extremely poor bounds). For Linear Arrangement, most of the work is in progress, but an approach based on an LP relaxation in which variables are associated with edge lengths, not necessarily associated with an ILP formulation of the problem, appears to be promising. For Bidimensional Bin Packing, the best way to derive bounds appears to be the solution of a suitable Bilinear Program, which is not directly connected to any ILP formulation of the problem. Finally, for Protein Folding in the H-P Model, there seems to be no reasonable known approach at all.

References

[1] A. Caprara, G. Lancia, S.K. Ng, *Sorting Permutations by Reversals through Branch-and-Price*, INFORMS Journal on Computing **13** (2001) 224–244.

[2] A. Caprara, *The Reversal Median Problem*, NFORMS Journal on Computing **15** (2003), 93–113.

[3] A. Caprara, J.J. Salazar, *Laying Out Sparse Graphs with Provably Minimum Bandwidth*, INFORMS Journal on Computing **17** (2005), 356–373.

[4] A. Caprara, M. Locatelli, M. Monaci, *Bidimensional Packing by Bilinear Programming*, in M. Jünger, V. Kaibel (eds.) *Proceedings of the Eleventh Conference on Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science **3509**, Springer-Verlag (2005), 377–391.

[5] J. Diaz, J. Petit, M. Serna, *A Survey on Graph Layout Problems*, ACM Computing Surveys **34** (2002), 313–356.

[6] W.H. Hart, S.C. Istrail, *Protein Folding in the Hydrophobic-Hidrophilic Model within 3/8 of Optimal*, Proceedings of the 27th ACM Symposium on Theory of Computing (1995), 157–168.

Towards industrial railroad optimization: Switching, routing, and scheduling

Marco Lübbecke

(joint work with Alberto Ceselli and Ines Spenke)

We discuss most recent and preliminary work on a large-scale combinatorial optimization problem arising in industrial practice.

Industrial switching involves moving materials on rail cars within or between industrial complexes and connecting with other rail carriers. Planning tasks include the making up of trains with a minimum shunting effort, the feasible and timely routing through an in-plant rail network on short paths, and assigning and scheduling of locomotives under safety and network capacity aspects. A human planner must resort to routine and simple heuristics, not least for the reason of unavailability of computer aided suggestions.

For the *routing/scheduling problem*, our set covering formulation with packing constraints

$$\begin{aligned}
\text{minimize} \quad & \sum_{k \in K} \sum_{p \in P_k} c_k^p \lambda_k^p & \\
\text{subject to} \quad & \sum_{k \in K} \sum_{p \in P_k} d_i^p \lambda_k^p \geq 1 & \forall \text{ requests } i \\
& \sum_{k \in K} \sum_{p \in P_k} a_{e,t}^p \lambda_k^p \leq 1 & \forall \text{ tracks } e, \quad \text{at time } t \\
& \sum_{p \in P_k} \lambda_k^p \leq 1 & \forall \text{ locomotives } k \\
& \lambda_k^p \in \{0, 1\} &
\end{aligned}$$

c_k^p	cost of path/schedule p, including setup cost for locomotive k
$d_i^p = 1$	iff request i is served on path p
$a_{e,t}^p = 1$	iff path p visits track b at time t
$\lambda_k^p = 1$	iff locomotive k executes path/schedule p

is a natural modeling choice. A feasible *schedule* (route and time) for locomotive $k \in K$ is encoded by a (particular) path $p \in P_k$ in our network. The packing constraints inhibit two different locomotives to visit the same track at the same time. These constraints are dynamically generated as well as the (exponentially many) variables of this model; that is, we propose a branch-and-cut-and-price (BCP) algorithm to solve this problem. The so-called pricing subproblem is a tailored resource constrained shortest path problem. We discuss some generic issues of th increasingly popular BCP technique for solving large-scale integer programs and particularities of our approach like special branching rules.

The *switching/shunting problem* can be described as follows. Given a set of stacks of a given maximum depth ("tracks"); an ordered list of items ("cars"), one for each stack; a set of sets of items, each set of items is identified by its color ("requests"), some items may/will remain uncolored ("not requested"); and one dedicated stack, called the train. *Allowed moves* are to move (groups of) items from the top of a stack to the top of the train, or from the top of the train to the top of a stack (preserving orderings). No stack may contain more items than allowed by the maximum depth. The problem is to finally have some stack which contains all colored items, consecutively, separated by color, and on the top of that stack (that is, no uncolored items "block" the access to the colored items). The objective is to minimize the number of moves. We think that this is an interesting combinatorial optimization problem, and may spawn a number of variants. We report on preliminary experience with a mixed integer programming formulation of a simplified version of this problem.

Our conclusion is that mixed integer programming is not only a universal modeling tool but became a more and more effective solution framework for all sorts

of practical (and of course, also mathematical) problem settings. We support this conclusion by the fact that within a rather short time period we were able to set up a rather involved implementation of our proposed algorithms. This brings us in a position to have a proof-of-concept after only a few months, even if our models are able to solve only small instances at the moment.

Our work is based on recent practical data from a German in-plant railroad.

On a problem by Fourier

Gerhard J. Woeginger

John Herivel relates the following story in his biography of the mathematical physicist Joseph Fourier (1768–1830). In 1788, Fourier corresponded with his friend and teacher C.L. Bonard, a professor of mathematics at Auxerre. In one of his letters Fourier posed the following problem: *"Here is a little problem of rather singular nature. It occurred to me in connection with certain propositions in Euclid we discussed on several occasions. Arrange 17 lines in the same plane so that they give 101 points of intersection. It is to be assumed that the lines extend to infinity, and that no point of intersection belongs to more than two lines."* One solution to Fourier's problem is to use four families of parallel lines with 2, 3, 4, and 8 lines, respectively.

The more general problem of deciding whether for given numbers n and m there exist n lines in the Euclidean plane with exactly m points of intersection is equivalent to the following question: Given two positive integers S and Q, decide whether there exist positive integers $x_1, \ldots, x_k$ with $\sum_{i=1}^{k} x_i = S$ and $\sum_{i=1}^{k} x_i^2 = Q$. The talk shows that this problem can be solved in time polynomially bounded in the logarithms of S and Q.

References

[1] J. Herivel, *Joseph Fourier – The Man and the Physicist*, Clarendon Press, Oxford (1975).

Secure network coding via filtered secret sharing

Cliff Stein

(joint work with Jon Feldman, Tal Malkin, Rocco Servedio)

Networks that carry information are now ubiquitous, and so the problem of using them efficiently is critical. One of the most exciting new ideas of the last few years in the theoretical study of information networks is *network coding*. This line of research (e.g., [14, 8, 16, 17, 6], see also [15]), introduced by Ahlswede et. al. [1], differs from traditional work on routing in networks in the following way. A packet sent through a network consists of routing information and data. Traditionally, routers manipulate the routing information, and just pass along the data. In network coding, we allow the routers to manipulate the data, i.e. we allow

the network to do computation on the data. It has been shown [1, 17, 8] that by doing so, we can increase the effective capacity of the network. Network coding has been suggested as a practical tool for use in content distribution networks over the Internet [7, 18], as well as for wireless networks [3, 19].

In a traditional multicast situation with a single source and multiple destinations, the amount of information that can be transmitted from the source s to a particular destination t_i is equal to the minimum cut κ_i between the source and destination. If we allow coding at the routers, we obtain the surprising result that we can *simultaneously* transmit $n = \min_i \kappa_i$ symbols of information to *every* destination [17]. Furthermore, given a network, we can construct such a network code in polynomial time [12]. In contrast, there are simple examples of networks in which this is not possible with traditional routing [9].

As any user of the Internet is painfully aware, it is imperative to consider security issues in any network scenario. To that end, several researchers have considered security issues in network coding. The problem of making a linear network code secure was first studied by Cai & Yeung [2], who considered a "wire-tap" adversary that can look at a bounded number of network edges. Jain [13] also considers this model, and gives more precise security conditions in certain cases. Ho et. al.[10] consider the related problem of network coding in the presence of a *Byzantine attacker* who can modify data sent from a node in the network.

In this paper we study secure multicast network coding against a wire-tap adversary where perfect (information-theoretic) security is required. We abstract away the network topology and reduce the problem of information-theoretically secure linear network coding to a new variant of secret sharing, which we call *filtered secret sharing* and believe to be of independent interest. Informally, while in classical threshold secret sharing security is maintained against an adversary who receives at most k of the n shares, in filtered secret sharing the adversary receives at most k among a set of $N \geq n$ *fixed linear combinations of all* n *shares*. In other words, the shares of the secret are passed through some fixed n-by-N linear filter, and then k out of N of these combinations are given to the adversary. This filtered secret sharing problem is investigated using techniques from secret sharing and from classical coding theory.

1. Motivation For Our Work

Making a system secure always comes at a cost. For example, if one uses cryptography, one pays a cost in computation time. In network coding, the cost is that less information can be transmitted in each time step. More precisely we will study trade-offs between *security, bandwidth and capacity* in linear multicast network coding schemes. We will later define each of these terms more precisely, but give an informal definition here. *Security* is characterized by how many edges an adversary can observe without obtaining any information about the message in the network. Information is transmitted as elements of a finite field $\mathbb{F}_q$. The logarithm of the field size is the edge *bandwidth*, or how much information (in bits) needs to travel through an edge in one step. In many applications, an edge will

have a physical upper limit on bandwidth; this will force us to make the bandwidth of our code small. For security, random symbols will be transmitted along with the information symbols; we measure the *capacity* of the network code as the number of information symbols transmitted in each step. The overall goal is to operate at a capacity close to the minimum cut value $n = \min_i \kappa_i$ and be secure against an adversary who can view many edges, under possibly limited edge bandwidth.

Cai & Yeung [2] considered one particular setting of security, bandwidth and capacity. Specifically, if n is the minimum cut value in the underlying network of N edges, and $k < n$ is the bound on the number of edges available to the adversary, they demonstrate the existence of a scheme with capacity $n-k$ as long as the edge bandwidth is greater than $\log \binom{N}{k}$. This result has two main drawbacks: (i) the construction of the scheme takes $\binom{N}{k}$ steps, and (ii) the bandwidth requirement is prohibitive for large k. Note that in the absence of security considerations the bandwidth requirement is at most the logarithm of the number of terminals in the network, and hence is at most $\log N$ [17, 12].

2. Our Results

We exhibit new trade-offs between security, bandwidth and capacity of secure linear network coding schemes. We give positive results on achievable parameters that are more powerful than those previously known. We also give new negative results showing that filtered secret sharing is unsolvable in certain cases.

We first show that by giving up a little bit of capacity (namely, sending $n-\Theta(k)$ symbols instead of $n-k$), we can efficiently construct a scheme that is secure with high probability, where the required bandwidth is only $\Theta(\log N)$, independent of k. This bound is superior to the bound of $\log \binom{N}{k}$ in most cases, and allows a trade-off between capacity and field size. For very large $k = \Theta(N)$, our bandwidth requirement becomes $\Theta(1)$, independent of both N and k.

Our negative result gives further support to our approach of giving up capacity in order to achieve security with a small bandwidth. We show that if one insists upon sending $n-k$ message symbols, then there are cases where the bandwidth must be almost as large as $\Theta(\sqrt{k}\log N)$. (We give more precise statements of both our positive and negative results later in the paper.)

3. Techniques

As mentioned above, we reduce the secure network coding problem to a variant of secret sharing, which we call filtered secret sharing. We then show that filtered secret sharing is actually equivalent to a certain generalized (classical) code construction problem. More precisely, we study the problem of designing a code that has large distance from a given code. Within this framework, we derive positive results using methods similar to those used in a proof of the Gilbert-Varshamov bound (see [11]), and negative results using a bound [5] on the covering radius [4] that linear codes can achieve.

Our method for constructing a secure network code has a nice feature that makes it more useful when the network code is fixed (in hardware, say). If we

are given a network and a network code, we can make this code secure *without changing the network code*, but only by applying a linear transformation to the input. Our ability to do this follows from a linear algebraic approach to network coding which actually abstracts away the network topology, along the lines of [14].

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. on Information Theory*, 46:1204–1216, 2000.

[2] N. Cai and R. W. Yeung. Secure network coding. In *International Symposium on Information Theory (ISIT '02)*, June 2002.

[3] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. In *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003.

[4] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes*. Elsevier, Amsterdam, 1997.

[5] G. D. Cohen and P. Frankl. Good coverings of hamming spaces with spheres. *Discrete Mathematics*, 56:125–131, 1985.

[6] C. Fragouli and E. Soljanin. Information flow decomposition for network coding. Submitted to IEEE Transactions on Information Theory, June 2004.

[7] C. Gkantsidis and P. R. Rodriguez. Network coding for large scale content distribution. Technical Report MSR-TR-2004-80, Microsoft Research, 2004.

[8] T. Ho, D. Karger, M. Medard, and R. Koetter. Network coding from a network flow perspective. In *International Symposium on Information Theory (ISIT '03)*, June 2003.

[9] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *IEEE International Symposium on Information Theory (ISIT)*, June 2003.

[10] T. Ho, B. Leong, R. Koetter, M. MÃ©dard, M. Effros, and D. R. Karger. Byzantine modification detection in multicast networks using randomized network coding. In *IEEE International Symposium on Information Theory (ISIT 2004)*, June 2004.

[11] W. C. Huffman and V. Pless. *Fundamentals of Error Correcting Codes*. Cambridge University Press, 2003.

[12] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. Submitted to IEEE Transactions on Information Theory, July 2003.

[13] K. Jain. Security based on network topology against the wiretapping attack. *IEEE Wireless Communications*, pages 68–71, February 2004.

[14] R. Koetter and M. Medard. An algebraic approach to network coding. *Transactions on Networking*, October 2003.

[15] Ralf Koetter. Network coding home page, `http://tesla.csl.uiuc.edu/~koetter/nwc`, 2004.

[16] A. R. Lehman and E. Lehman. Complexity classification of network information flow problemss. In *Symposium on Discrete Algorithms (SODA '04)*, January 2004.

[17] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49:371–381, Februray 2003.

[18] D. S. Lun, M. MÃ©dard, and M. Effros. On coding for reliable communication over packet networks. In *Proceedings 42nd Annual Allerton Conference on Communication, Control, and Computing*, October 2004.

[19] Y. Wu, P. A. Chou, and S.-Y. Kung. Information exchange in wireless networks with network coding and physical-layer broadcast. Technical Report MSR-TR-2004-78, Microsoft Research, 2004.

Collective decisions for tours

Ulrich Pferschy
(joint work with Christian Klamler)

Recently, there emerged a tendency of combining concepts from Social Choice Theory, Computer Science and Mathematical Optimization leading to a series of workshops devoted to the interaction of these traditionally separate fields, e.g. at Oberwolfach in March 2004. Within this trend we investigate the application of social choice concepts, such as preference relations and voting rules, to classical problems of Combinatorial Optimization. This opens up an interesting new way to look at well-studied problems and should lead to fruitful cooperations between two separate scientific communities joined by their common usage of mathematical language.

Following earlier related works on graph problems [3], [6], and scheduling problems [4], we concentrate our investigations on the most classical Combinatorial Optimization problem, the Traveling Salesman/person Problem (TSP) and continue with the Minimum Spanning Tree Problem (MST).

1. General Framework

The general idea of our approach is the following: Instead of assigning cost coefficients to edges and performing arithmetic operations on them to find a unique optimal solution value, we consider the assessment of every edge by a group of n individuals. Each of them expresses his/her preferences as a ranking, i.e. a total ordering of all alternatives (of all edges, in our case). The collection of these n rankings is called a *preference profile*. It is the central task of Social Choice Theory to determine a social preference by aggregating individual preferences (see e.g. Sen [7]). The resulting area of research is strongly axiomatic and focuses on characterizations and impossibility results instead of algorithmic concepts (see e.g. Nurmi [5]). We distinguish between a *social welfare function*, which aggregates the n orderings into a single total order, and a *social choice function*, which is applied on a subset of alternatives and selects a subset of alternatives taking into account the n orderings.

Among the most widely studied aggregation rules are the *Borda Count*, where the ranking by every individual is represented by a numerical score and the aggregation follows from summing up the scores, and the *Approval Vote*, where every individual gives a consistent bipartitioning of every subset of alternatives into approved and non-approved options. In the latter case aggregation is achieved by counting for every alternative the number of individuals by which it is approved.

Other mechanisms of major interest are the *Plurality Rule*, where alternatives are ranked according to the number of top positions they achieve in the individual rankings, and the *Simple Majority Rule*, where an alternative is socially preferred to another alternative if a majority of individuals values the former above the latter. Note that the latter rule does not necessarily induce a total ordering but might well lead to cycles.

The aim of our research is to establish and analyze algorithms which determine solutions that are in some way good or acceptable for a large number of individuals in the above social choice context.

2. Constructing Tours

We will consider the problem of finding circular tours in a complete graph $G = (V, E)$ visiting every vertex exactly once (see e.g. [1]). A natural first step is the analysis and comparison of a global and a local decision approach. The *global tour construction* procedure applies a social welfare function and adds edges to the tour according to the resulting ordering as long as they do not violate the tour condition. This can be seen as an analogon to the multi-fragment heuristic known from classical combinatorial optimization. The *local tour construction* procedure starts at an arbitrary vertex of the graph and submits all its neighbours to a decision by a social choice function producing a single winner. This edge is included in the tour and the procedure is iterated at the new endpoint of this edge with all its still unvisited neighbours. Obviously, this approach corresponds to the classical nearest neighbour heuristic.

To compare tours with each other, we use the following *pairwise dominance* criterion:

> Tour T *pairwise dominates* tour T' if and only if for all individuals $i \in \{1, \ldots, n\}$, there exists a bijection $g_i : T \to T'$ such that for all $a \in T$, a is ranked higher than $g_i(a)$ by individual i.

3. Results

The flavour of our results is negative in the following sense: Both construction rules can be pairwise dominated by each other. For the case of $|V| = 5$, i.e. $|E| = 10$, this means that the edge set computed by one of the two construction rules can be pairwise dominated by its complement, which is computed by the other rule. Hence, all n individuals would have preferred the use of a different tour construction.

More formally, we have the following statement:

Theorem 1. *Using the Borda Rule, for any complete graph with $|V| \geq 5$, there exist preference profiles such that:*
(i) the result of the local tour construction is pairwise dominated by the global tour construction.
(ii) the result of the global tour construction is pairwise dominated by the local tour construction.

Constructions with these properties can also be given for the Approval Vote. Note that for Approval Vote the dominance of the global tour can only be attained by a suitable utilization of tie-breaking, which consists of approving all but the lowest ranked option.

For the Plurality Rule it is easy to see that the globally constructed tour can never be pairwise dominated by any other tour. However, the above property holds for the local tour construction. The Simple Majority Rule does not necessarily produce a total ordering and thus requires some mechanism to deal with cycles such as assigning equal rank to the transitive closure of the elements in such a cycle. It can be shown that for the Simple Majority Rule the tours derived by local and global tour constructions can never be disjoint and hence can not pairwise dominate each other.

While these results are in line with the intuitive expectation from a classical optimization point of view, a study of the MST shows a more significant deviation of the social choice approach. Both the local construction (equivalent to Prim's algorithm) and the global construction (known as Kruskal's algorithm) are exact algorithms for the numerical problem. However, for the Borda Rule, the local construction can be pairwise dominated by the global construction which remains undominated.

For more details we point to our recent paper [2]. Extensions of these results to other classical combinatorial optimization problems and an integration into the properties and issues studied in social choice theory are under way. In particular, the comparison of different social choice/welfare functions and the development of algorithms that permit also positive results are points of major interest.

References

[1] Gutin, G. and A.P. Punnen (eds.), *The Traveling Salesman Problem and Its Variations*, Kluwer Academic Publishers, 2002.

[2] C. Klamler, U. Pferschy, "The traveling group problem", Department of Economics, University of Graz, submitted, (2005).

[3] J. Monnot, O. Spanjaard, "Bottleneck shortest paths on a partially ordered scale", *4OR* **1** (2003), 225–241.

[4] H. Moulin, R. Stong, "Fair queuing and other probabilistic allocation methods ", *Mathematics of Operations Research* **27** (2002), 1–20.

[5] Nurmi, H., *Comparing Voting Systems*, D. Reidel, 1987.

[6] P. Perny, O. Spanjaard, "A preference-based approach to spanning trees and shortest paths problems", *European Journal of Operational Research* **162** (2005), 584–601.

[7] Sen, A.K., "Social choice theory", in: Arrow, K.J. and M. Intriligator (eds.), *Handbook of Mathematical Economics vol. III*, Amsterdam, North Holland, 1986.

Selfish routing and the price of anarchy: A look forward, and a look back

Andreas S. Schulz

Nash equilibria and related game-theoretic concepts have lately received increasing attention in the theoretical computer science and combinatorial optimization communities. The "price of anarchy," put forward by Koutsoupias and Papadimitriou ([14], see also [17]) has emerged as the perhaps most prominent notion. It measures the worst-case efficiency loss of a system that is left to independent, selfishly acting

agents if compared to coordination by some central authority. While this view is now used in a variety of settings including location analysis, pricing, supply chain coordination, and system design, it was originally developed in the context of network routing. In this talk, we present a short geometric proof (taken from [9]) for several price-of-anarchy results that were recently established in a series of papers on selfish routing in multicommodity flow networks (including [4, 8, 18, 19, 20]). This novel proof also facilitates two new types of results (described in [9]). On the one hand, it leads to pseudo-approximation results that depend on the class of allowable cost functions. On the other hand, one can derive improved bounds on the inefficiency of Nash equilibria for situations in which there are either no fixed costs, or when fixed costs represent a nonnegligible fraction of the total cost. These tighter bounds help to explain empirical observations, e.g., in vehicular traffic networks [13]. Our analysis holds in the more general framework of (nonatomic) congestion games, and we also discuss the differences to atomic congestion games, which only have finitely many players. In particular, we address the existence and the complexity of computing pure Nash equilibria (as detailed in [10, 11, 15], among others), as well as the price of anarchy in the case of unsplittable flows [3, 5], splittable flows [7], and k-splittable flows [15]. We conclude by mentioning some other interesting directions in this context, including the price of anarchy for mixed Nash equilibria [6], the price of stability [1, 6, 21], the value of mediation [2, 6], and the attempt to obtain bounds on the value of states to which selfish play could lead [12, 16].

References

[1] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden, *The price of stability for network design with fair cost allocation*, Proceedings of the 45th Symposium on Foundations of Computer Science (2004), 295–304.

[2] I. Ashlahgi, D. Monderer, and M. Tennenholtz, *On the value of correlation*, Working paper, Technion, Haifa, Israel, 2005.

[3] B. Awerbuch, Y. Azar, and A. Epstein, *The price of routing unsplittable flow*, Proceedings of the 37th Annual ACM Symposium on Theory of Computing (2005), 57–66.

[4] C.K. Chau and K.M. Sim, *The price of anarchy for non-atomic congestion games with symmetric cost maps and elastic demands*, Operations Research Letters **31** (2003), 327–334.

[5] G. Christodoulou and E. Koutsoupias, *The price of anarchy in finite congestion games*, Proceedings of the 37th Annual ACM Symposium on Theory of Computing (2005), 67–73.

[6] G. Christodoulou and E. Koutsoupias, *On the price of anarchy and stability of correlated equilibria of linear congestion games*, in G.S. Brodal and S. Leonardi (eds.), Algorithms – ESA 2005, Lectures Notes in Computer Science **3669** (2005), 59–70.

[7] R. Cominetti, J.R. Correa, and N.E. Stier Moses, *Network games with atomic players*, Working paper, October 2005.

[8] J.R. Correa, A.S. Schulz, and N.E. Stier Moses, *Selfish routing in capacitated networks*, Mathematics of Operations Research **29** (2004), 961–976.

[9] J.R. Correa, A.S. Schulz, and N.E. Stier Moses, *On the inefficiency of equilibria in congestion games*, in M. Jünger and V. Kaibel (eds.), Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science **3509** (2005), 167–181.

[10] J. Dunkel, *The complexity of pure-strategy Nash equilibria in non-cooperative games*, Diploma thesis, Institute of Mathematics, Technische Universität Berlin, Berlin, Germany, July 2005.
[11] A. Fabrikant, C. Papadimitriou, and K. Talwar, *The complexity of pure Nash equilibria*, Proceedings of the 36th Annual ACM Symposium on Theory of Computing (2004), 604–612.
[12] M.X. Goemans, V.S. Mirrokni, and A. Vetta, *Sink equilibria and convergence*, Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (2005), 142–154.
[13] O. Jahn, R.H. Möhring, A.S. Schulz, and N.E. Stier Moses, *System-optimal routing of traffic flows with user constraints in networks with congestion*, Operations Research **53** (2005), 600–616.
[14] E. Koutsoupias and C. Papadimitriou, *Worst-case equilibria*, in C. Meinel and S. Tison (eds.), STACS 99, Lecture Notes in Computer Science **1563** (1999), 404–413.
[15] C. Meyers and A.S. Schulz, *Complexity and price of anarchy in congestion games*, Working paper, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, December 2005.
[16] V.S. Mirrokni and A. Vetta, *Convergence issues in competitive games*, in K. Jansen et al. (eds.), Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques, Lecture Notes in Computer Science **3122** (2004), 183–194.
[17] C. Papadimitirou, *Algorithms, games, and the Internet*, Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (2001), 749–753.
[18] T. Roughgarden, *The price of anarchy is independent of the network topology*, Journal of Computer and System Sciences **67** (2003), 341–364.
[19] T. Roughgarden and É. Tardos, *How bad is selfish routing?*, Journal of the ACM **49** (2002), 236–259.
[20] T. Roughgarden and É. Tardos, *Bounding the inefficiency of equilibria in nonatomic congestion games*, Games and Economic Behavior **47** (2004), 389–403.
[21] A.S. Schulz and N.E. Stier Moses, *On the performance of user equilibria in traffic networks*, Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (2003), 86–87.

Intermediate integer programming representations using value disjunctions

Robert Weismantel
(joint work with Matthias Köppe, Quentin Louveaux)

Extreme representations of the feasible points of a mixed-integer linear optimization problem are either given by means of the facet defining inequalities in the original space or by a set of feasible mixed integer points whose convex hull contains the feasible region. It is well known that in principle one such extreme representation can be transformed into the other extreme representation. However from an algorithmic point of view both extreme representations are very hard to achieve. This suggests to search for other, "intermediate" representations that are algorithmically more tractable, in the sense that they

- require less variables than the extreme representation by the vertices,
- require less constraints compared to the total number of facets of the convex hull,

- have a simpler combinatorial constraint structure than the facets of the convex hull in the original space and hence, the separation problem in the extended space is easier to solve.

In the literature, there are a couple of interesting examples of this type. Chopra and Rao [5, 6] introduced a directed formulation for the Steiner tree problem and showed that exponentially many inequalities in the undirected formulation are projections of a small number of directed inequalities. R. K. Martin [9] reports on the minimum spanning tree problem, which has as an inequality formulation of size $O(2^n)$. It can, however, alternatively be described as the projection of an extended formulation which requires $O(n^3)$ variables and $O(n^2)$ constraints. Moreover, there are many further compact extended formulations for specific combinatorial optimization problems, in particular for a variety of fixed-charge network problems.

For general binary mixed integer models extended formulations can be obtained from the so-called Lift-and-Project approach. This approach has its roots in the work of Egon Balas on disjunctive optimization [1, 2]. It was further refined in [10, 8, 3, 4] by introducing hierarchies of extended formulations whose variables represent more general subsets of original variables. The disadvantage of this approach is that the number of variables grows exponentially with the size of the subsets for which we introduce new variables. The tool that we propose in this paper to generate an extended formulation is the *value-disjunction procedure*. Resorting to this tool one can describe the convex hull of the given mixed-integer set as the intersection of several simpler polyhedra using the variables of an extended space. This is the *structure theorem* for the value disjunction procedure that we explain more formally below.

Let us partition the set $N = \{1, \ldots, n\}$ into subsets $N_1, \ldots, N_K$. For each of the sets N_i, we determine all the possible vectors ("values") generated by the columns A_j belonging to the variables indexed by N_i:

$$\mathcal{A}_i = \left\{ \sum_{j \in N_i} A_j x_j : x_j \in \{0, \ldots, u_j\} \text{ for } j \in N_i \right\}.$$

Since the integer variables are assumed to be bounded, the set $\mathcal{A}_i$ is finite; its cardinality $n_i = |\mathcal{A}_i|$ is at most $\prod_{j \in N_i}(1+u_j)$. Let the elements of $\mathcal{A}_i$ be numbered, $\mathcal{A}_i = \{\mathbf{f}_1^{N_i}, \ldots, \mathbf{f}_{n_i}^{N_i}\}$. We shall associate with $\mathbf{f}_k^{N_i}$ a new binary variable $y_k^{N_i}$. In order to simplify the subsequent expositions, we shall also use the abbreviating notations $A(\mathbf{x}^{N_i}) = \sum_{j \in N_i} A_j$, and moreover $A(\mathbf{y}^{N_i}) = \sum_{k=1}^{n_i} y_k^{N_i} \mathbf{f}_k^{N_i}$ and $A(\mathbf{y}) = \sum_{i=1}^{K} A(\mathbf{y}^{N_i})$.

We come to two major definitions that we make use of in this paper.

Definition 1. *For a given subset N_i, we define the* linking polyhedron *as*

$$V_i = \operatorname{conv}\Bigg\{ (\mathbf{x}^{N_i}, \mathbf{y}^{N_i}) \in \mathbb{Z}_+^{|N_i|} \times \{0,1\}^{n_i} : \sum_{j \in N_i} A_j x_j = \sum_{k=1}^{n_i} \mathbf{f}_k^{N_i} y_k^{N_i}$$

$$\sum_{k=1}^{n_i} y_k^{N_i} = 1$$

$$0 \le x_i \le u_i,\ i = 1, \ldots, n \qquad \Bigg\}. \tag{1}$$

Furthermore we define the aggregated polyhedron *as*

$$Q = \operatorname{conv}\Bigg\{ (\mathbf{y}, \mathbf{w}) \in \{0,1\}^{n_1 + \cdots + n_K} \times \mathbb{R}_+^d :$$

$$\sum_{i=1}^{K} \sum_{k=1}^{n_i} \mathbf{f}_k^{N_i} y_k^{N_i} + \sum_{j=1}^{d} G_j w_j \le \mathbf{b}$$

$$\sum_{k=1}^{n_i} y_k^{N_i} \le 1 \text{ for all } i = 1, \ldots, K \qquad \Bigg\}. \tag{2}$$

Theorem 1 (Structure Theorem for Value Disjunction)**.**

$$P = \Big\{ (\mathbf{x}, \mathbf{w}) \in \mathbb{R}^n \times \mathbb{R}^d : \text{there exists } \mathbf{y} \in [0,1]^{n_1 + \cdots + n_K} \text{ with}$$

$$(\mathbf{y}, \mathbf{w}) \in Q \text{ and } (\mathbf{x}^{N_i}, \mathbf{y}^{N_i}) \in V_i \text{ for all } i \Big\}. \tag{3}$$

Next we introduce the family of linking polyhedra. In the special but important case that such a linking polyhedron comes from the unweighted sum of a set of variables, we completely describe the polyhedron by means of linear inequalities and equations.

We show that the value disjunction procedure is a tool to compute complete descriptions in an extended space. As an example we consider the 0/1 knapsack problem with three distinct coefficients:

$$\sum_{j \in N_1} \mu x_j + \sum_{j \in N_2} \lambda x_j + \sum_{j \in N_3} \sigma x_j \le \beta, \tag{4}$$

where N_1, N_2, N_3 are pairwise disjoint index sets.

Finally, we investigate one way of making computational use of value disjunctions: By branching also on the new binary variables of the extended formulation instead of only on the original variables, it is possible to take more flexible branching decisions. In fact, we propose such a branching scheme for situations where none of the usual LP-based variable selection criteria provides a solid basis for taking a branching decision. Such situations frequently occur in very hard integer programs like the market-split instances [7]. We investigate the effect of branching simplifying the facet description: A branching decision is considered good if the facet descriptions of the generated subproblems are significantly simpler than the

original facet description. Using experiments with randomly generated problem instances, we show that it is possible to make a branching decision based on the structure of the problem which is better than branching on the original variables. Finally we report on simple computational experiments with a few hard integer programs, where we branch explicitly on the new binary variables and then solve the subproblems with the branch-and-cut system CPLEX. We obtain a significant reduction in both the number of nodes and the computation time.

References

[1] Egon Balas, *Intersection cuts – a new type of cutting planes for integer programming*, Operations Research **19** (1971), 19–39.

[2] ———, *Disjunctive programming: Cutting planes from logical conditions*, Nonlinear Programming 2 (O. L. Mangasarian et al., ed.), Academic Press, London, 1975, pp. 279–312.

[3] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols, *A lift-and-project cutting plane algorithm for mixed* $0-1$ *programs*, Mathematical Programming **58** (1993), 295–324.

[4] Daniel Bienstock and Mark Zuckerberg, *Subset algebra lift operators for 0-1 programming*, Tech. Report 1, CORC, 2002.

[5] S. Chopra and M. R. Rao, *The Steiner tree problem I: Formulations, compositions and extension of facets*, Mathematical Programming **64** (1994), no. 2, 209–229.

[6] ———, *The Steiner tree problem II: Properties and classes of facets*, Mathematical Programming **64** (1994), no. 2, 231–246.

[7] Gérard Cornuéjols and Milind Dawande, *A class of hard small* 0-1 *programs*, Integer programming and combinatorial optimization (Houston, TX, 1998), Lecture Notes in Computer Science, vol. 1412, Springer, Berlin, 1998, pp. 284–293. MR 2000h:90042

[8] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* $0-1$ *optimization*, SIAM Journal on Optimization **1** (1991), 166–190.

[9] R. Kipp Martin, *Using separation algorithms to generate mixed integer model reformulations*, Operations Research Letters **10** (1991), 119–128.

[10] H. Sherali and W. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM Journal of Discrete Mathematics **3** (1990), 411–430.

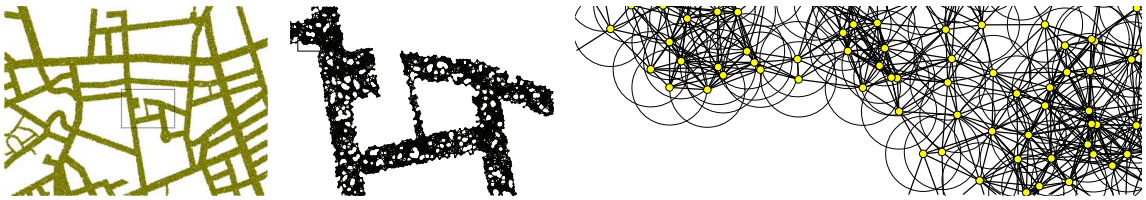
Algorithmic aspects of large sensor networks

Sándor P. Fekete

(joint work with Stefan Fischer, Alexander Kröller, Dennis Pfisterer)

The study of wireless sensor networks (WSN) has become a rapidly developing research area. Typical scenarios involve a large swarm of small and inexpensive processor nodes, each with limited computing and communication resources, that are distributed in some geometric region; communication is performed by wireless radio with limited range. Upon start-up, the swarm forms a decentralized and self-organizing network that surveys the region.

From an algorithmic point of view, these characteristics imply absence of a central control unit, limited capabilities of nodes, and limited communication between nodes. This requires developing new algorithmic ideas that combine methods of distributed computing and network protocols with traditional centralized network



(a) Sensor nodes in a polygonal region. (b) Zoom into reveals communication graph. (a)(c) Further zoom shows communication ranges.

FIGURE 1. Scenario of a sensor network in the streets around Braunschweig University of Technology.

algorithms. In other words: How can we use a limited amount of strictly local information in order to achieve distributed knowledge of global network properties?

We propose two procedures for dealing with this challenge. One identifies the boundaries of the network, the other constructs a clustering that describes the network topology. Most details are omitted in this extended abstract, see our upcoming SODA paper [2].

1. Model

A *Sensor Network* $G = (V, E)$ is a graph $G = (V, E)$, with an edge between any two nodes that can communicate with each other. The network is embedded in the plane by the mapping $p : V \to \mathbb{R}^2$, but positions are not available to the nodes. We assume that (G, p) is a *d-Quasi Unit Disk Graph* for parameter $d \leq 1$, i.e., neighbors in the graph are at most 1 apart and non-neighbors at least d. This model for radio networks is used often because it is both realistic and theoretically tractable [1]. Notice that every graph is a 0-QUDG. Here, we assume $d \geq \frac{1}{2}\sqrt{2}$.

We denote by $N_1(U)$ the set of nodes that are neighbor of any $u \in U$, and define $N_k(U) = N_1(N_{k-1}(U))$ for $k > 1$. $\Delta_k = \max_{v \in V} |N_k(\{v\})|$ is the size of a largest k-neighborhood in G. Notice that for practical geometric networks, $\Delta_k = \mathcal{O}(k^2\Delta_1)$ is a realistic assumption. For our analysis of algorithms, we use the well-established synchronized $\mathcal{CONGEST}$ model [3] with simultaneous wakeup.

2. Boundary Recognition

Our procedure to detect boundaries is based on structures where a certain set of nodes is provably on the inside of a cycle in the network. This is based on the fact that in a $\frac{1}{2}\sqrt{2}$-QUDG, a path crossing a cycle is always witnessed by some cycle node. Because non-neighboring nodes have minimum distance d, an independent node set requires a certain amount of space in the embedding. On the other hand, the space that can be surrounded by a cycle is limited due to the upper bound on the edge lengths. Now a packing argument serves as proof that a connected set of nodes containing some independent nodes is on the outside of a cycle. Based on this, we identify a structure for which we can even prove *insideness*: An m-flower is an induced subgraph whose node set consists of a seed $f_0 \in V$, independent nodes $f_{1,1}, \ldots, f_{1,m} \in V$, bridges $f_{2,1}, \ldots, f_{2,m} \in V$, hooks $f_{3,1}, \ldots, f_{3,m} \in V$,

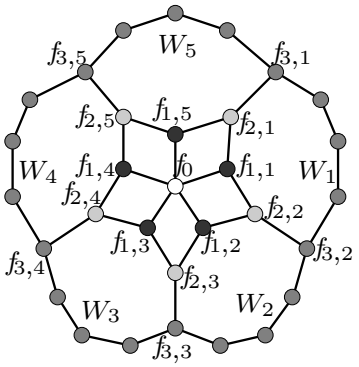


FIGURE 2. A 5-flower.

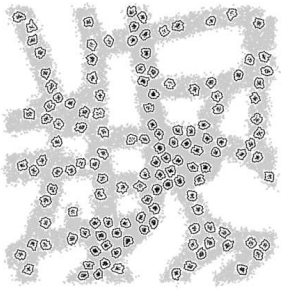


FIGURE 3. Result of the flower procedure.

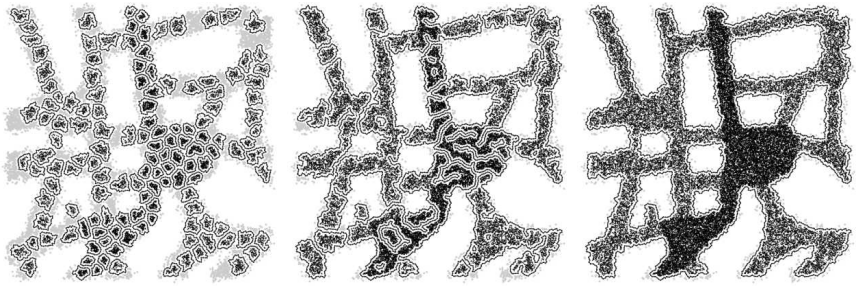


FIGURE 4. Snapshots and final state of the Augmenting Cycle algorithm.

and chordless paths $W_1, \ldots, W_m$. The edges are shown in Figure 2. A flower is *feasible*, if m and $|W_1|, \ldots, |W_m|$ fulfill some constraints not shown here due to space limitations. Flowers are useful due to the following fact:

Theorem 1. *In every d-QUDG embedding of a feasible m-flower, the independent nodes are placed on the inside of the cycle formed by* $C := \{f_{3,1}, \ldots, f_{3,m}\} \cup \bigcup_{j=1}^{m} W_j$. *Moreover, every network with a* $(\frac{3}{2} - \sqrt{2})$*-dense populated disk of radius 3 contains a feasible 4-flower in that disk.*

We say G is *ε-dense* in a region A, if every ε-ball in A contains at least one node. Because flowers are strictly local structures, they can be easily identified by local algorithms in time $\mathcal{O}(\Delta_1)$ and message complexity $\mathcal{O}(\Delta_1 \Delta_8)$. In an example network of 60,000 sparsely connected nodes, our procedure identified 138 disjoint flowers (Figure 3); a single suffices for the second stage of our algorithm.

In a second stage, we augment the flowers by adding extensions to their outer cycles, such that insideness can still be proven for all contained nodes. By repeating a local search procedure, the flowers grow to enclose more and more nodes, begin to merge together, leading to a single structure that contains most of the network. Our extension algorithm can be implemented locally, leading to runtime $\mathcal{O}(\Delta_K^K \Delta_1 + |V|)$ and message complexity $\mathcal{O}(\Delta_K^K |V|)$ per extension step, where K is a parameter between 10 and 20. See Figure 4 for a visualization.

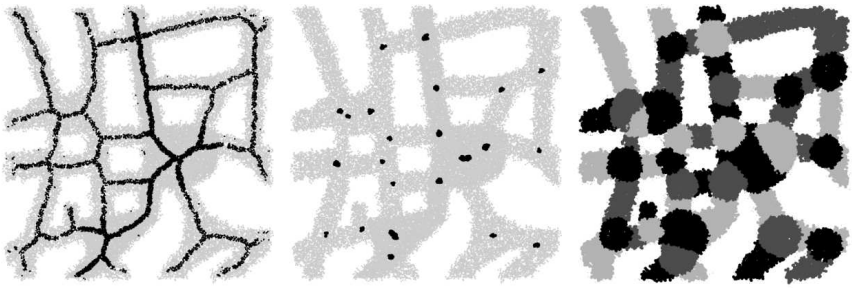


FIGURE 5. 2- and 3-Voronoi nodes; Final clustering.

We use the identified boundaries to construct our topological clustering. Nodes that have the same shortest path distance to k different boundaries are called k-Voronoi nodes, where distant parts of a single boundary are also considered as different. The 3-Voronois are small groups in the center of intersections, and the 2-Voronois form strips between them (Figure 5). Hence, we build intersection clusters around the 3-Voronois and street clusters along the 2-Voronois, leading to a clustering that reflects the network topology closely. We developed a local algorithm to perform the clustering. It has an initial synchronization step with runtime and message complexity $\mathcal{O}(|V|^3)$ that can run parallel to the boundary detection, and a clustering step with complexities $\mathcal{O}(|V| \log |V|)$.

REFERENCES

[1] Lali Barrière, Pierre Fraigniaud, and Lata Narayanan. Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *DIALM '01*, pages 19–27, New York, NY, USA, 2001. ACM Press.

[2] A. Kröller, S.P. Fekete, D. Pfisterer, and S. Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. To appear in Proc. 17th ACM-SIAM Symposium on Discrete Algorithms (SODA 2006).

[3] David Peleg. *Distributed computing: a locality-sensitive approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

Mixed integer cuts from two constraints and two integer variables

LAURENCE WOLSEY

This is joint work with Kent Andersen. We consider the generation of mixed integer cuts from two rows of an optimal LP tableau in which the two basic variables are required to be integer. More specifically we consider the Gomory corner polyhedron arising by dropping the nonnegativity constraints on these two integer variables.

Now distinguishing two of the continuous non-basic variables s_1, s_2, the set of interest can be represented in two different ways:

$$X_q^p = \{(x, s, y, z) \in \mathbb{Z}^2 \times \mathbb{R}_+^2 \times \mathbb{R}_+^p \times \mathbb{Z}_+^q : Ix + As + By + Cz = f\}$$

where I is a 2×2 identity matrix, A is a 2×2 nonsingular matrix, B, C are $2 \times p$ and $2 \times q$ matrices respectively and $f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$ with $0 < f_i < 1$ for $i = 1, 2$, or alternatively as

$$Z_q^p = \{(x, y, z) \in \mathbb{Z}^2 \times \mathbb{R}_+^p \times \mathbb{Z}_+^q : Rx + RBy + RCz \geq Rf\}.$$

where $R = -det(A)A^{-1}$.

Our main result is to derive strong valid inequalities for this set when A has a certain structure. Specifically we require that A is of the form $\begin{pmatrix} -a_1 & \omega \\ 1 & -a_2 \end{pmatrix}$ with $a_i > 0$ for $i = 1, 2$ and $a_1 a_2 > \omega$.

First we consider what happens when $p = q = 0$. Z_0^0 is the set

$$a_2 x_1 + \omega x_2 \geq \delta_2$$
$$x_1 + a_1 x_2 \geq \delta_1$$
$$x \in \mathbb{Z}^2,$$

where $\begin{pmatrix} \delta_2 \\ \delta_1 \end{pmatrix} = Rf = \begin{pmatrix} a_2 f_1 + \omega f_2 \\ f_1 + a_1 f_2 \end{pmatrix}$.

First we derive necessary and sufficient conditions for the inequality

$$x_1 + (\lceil \delta_1 \rceil - 1) x_2 \geq \lceil \delta_1 \rceil$$

to be valid for Z_0^0. Then we consider how to lift in the continuous variables so as to obtain a tight valid inequality for Z_0^p. We take both an intersection cut viewpoint based on a lattice-free triangle in the (x_1, x_2) space, and a disjunctive viewpoint in which we split the (x_1, x_2) space into two, derive a mixed integer rounding inequality for each part and then combine them to obtain a valid inequality for the union. Both approaches lead to the same inequality which, when the integer variables are included can be written as

$$(1) \qquad x_1 + (\lceil \delta_1 \rceil - 1) x_2 + \sum_{j=1}^{p} \bar{H} \begin{pmatrix} \beta_1^j \\ \beta_2^j \end{pmatrix} y_j + \sum_{j=1}^{q} H \begin{pmatrix} \gamma_1^j \\ \gamma_2^j \end{pmatrix} z_j \geq \lceil \delta_1 \rceil$$

where $H \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \max[G_{\frac{\delta_2 - \omega}{a_2}}(\frac{d_1}{a_2}), G_{f(\delta_1)}(d_2)]$ and $\bar{H} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \max[\frac{d_1^+}{\delta_2 - \omega}, \frac{d_2^+}{\bar{f}(\delta_1)}]$, where $G_\alpha(d) = \lfloor d \rfloor + \frac{\min[f(d), \alpha]}{\alpha}$ is the subadditive function associated to the mixed integer rounding inequality, and $\bar{f}(\delta) = \delta - \lceil \delta \rceil + 1$. Precise conditions are given for when this inequality is facet-defining.

It is also shown, using an example of Schrijver, that the inequality produced is one that cannot always be obtained by finitely generated split or mixed integer rounding inequalities.

Finally various other inequalities are derived in a similar way.

Open Questions

Is it true that in general the inequality (1) cannot be finitely generated using MIGs

or split cuts?
Given that the split cut is an intersection cut consisting of a cylindrical 1-simplex, and the inequality (1) is an intersection cut based on a cylindrical 2-simplex, is there some valid inequality for $k > 2$ rows that is easy to calculate based on a k-simplex?
Johnson [1] studies the minimal sets of points that are necessary to give a description of the polar of sets such as X_q^p using extrapolation of subadditive functions on $[0,1]^2$. Is it feasible to calculate $\mathrm{conv}(X_q^p)$ for instances with small values of p and q.
Finally there are the obvious computational questions of whether the inequalities proposed here are effective either in place of, or in combination with Gomory mixed integer cuts?

References

[1] E.L. Johnson. On the group problem for mixed integer programming. *Mathematical Programming Study*, 2:137–179, 1974.

On a convex optimization model for the consolidation of farmland

Peter Gritzmann
(joint work with Andreas Brieden, Christoph Metzger)

In many regions farmers cultivate a number of small lots that are distributed over a wider area. This leads to high overhead costs and economically prohibits use of high tech machinery hence results in a non-favorable cost-structure of production. The classical form of land consolidation is typically too expensive and too rigid, whence consolidation based on lend-lease agreements has been suggested. Of course, the underlying mathematical clustering problem is NP-hard even in the most simple cases.

We give and analyze a new approximate 0-1-convex optimization algorithm, where in effect the centers of gravity of the clusters are pushed apart. The core of the method is based on the use of Minkowski spaces whose unit balls stems from the dual of a cartesian products of permutahedra. It is shown how these unit balls themselves can be tightly approximated by Hardamard matrix based polytopes with only linearly many facets. The facet normals are then used as objective function vectors for a polynomial-time linear programming approximation algorithm. We derive a worst case bound for the approximation error but also report on the practical performance of this method for land consolidation in Northern Bavaria, Germany.

Max-algebra and combinatorial optimisation: Connections and open problems

Peter Butkovic

Let $a \oplus b = \max(a,b)$ and $a \otimes b = a + b$ for $a, b \in \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$. By *max-algebra* we understand the analogue of linear algebra developed for the pair of operations $(\oplus, \otimes)$, extended to matrices and vectors in the same way as in linear algebra, that is if $A = (a_{ij})$, $B = (b_{ij})$ and $C = (c_{ij})$ are matrices with entries from $\overline{\mathbb{R}}$ of compatible sizes, we write $C = A \oplus B$ if $c_{ij} = a_{ij} \oplus b_{ij}$ for all i, j and $C = A \otimes B$ if $c_{ij} = \sum_k^{\oplus} a_{ik} \otimes b_{kj} = \max_k(a_{ik} + b_{kj})$ for all i, j. The iterated product $A \otimes A \otimes ... \otimes A$ in which the symbol A appears k-times will be denoted by $A^{(k)}$.

Let $A = (a_{ij}) \in \overline{\mathbb{R}}^{n \times n}$. We denote $N = \{1, ..., n\}$. The complete arc-weighted digraph associated with A is $D_A = (N, N \times N, a_{ij})$, the finiteness digraph is $F_A = (N, \{(i,j); a_{ij} > -\infty\})$, the zero digraph is $Z_A = (N, \{(i,j); a_{ij} = 0\})$.

1. Links between max-algebraic problems and combinatorial or combinatorial optimisation problems: The set covering - solvability of max-algebraic linear systems, the minimal set covering - unique solvability of a linear system, existence of a directed cycle - strong regularity of a matrix, signnonsingularity or existence of an even directed cycle - regularity of a matrix, maximum cycle mean - eigenvalue, longest-distances vectors - eigenvectors, best principal submatrices - coefficients of a characteristic polynomial, linear assignment problem - permanent of a matrix.

2. Examples of combinatorial optimisation results obtained as a consequence of the max-algebraic theory.

2.a) The (assignment problem) normal form of a matrix and the longest-distances vectors. Let $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ and denote by P_n the set of all permutations of the set N. Then the (*max-algebraic*) *permanent* of A is $maper(A) = \sum_{\pi \in P_n}^{\oplus} \prod_{i \in N}^{\otimes} a_{i,\pi(i)}$. In the conventional notation this reads $maper(A) = \max_{\pi \in P_n} \sum_{i \in N} a_{i,\pi(i)}$ and thus $maper(A)$ is the optimal assignment problem value for A. For $\pi \in P_n$ we denote $w(A, \pi) = \prod_{i \in N}^{\otimes} a_{i,\pi(i)} = \sum_{i \in N} a_{i,\pi(i)}$. The set of all optimal permutations will be denoted by $ap(A)$, that is $ap(A) = \{\pi \in P_n; maper(A) = w(A, \pi)\}$.

A matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is called *normal* [*strictly normal*] if $a_{ij} \leq a_{ii} = 0$ for all $i, j \in N$ [if $a_{ij} < a_{ii} = 0$ for all $i, j \in N, i \neq j$]. A normal matrix can be obtained from any $A \in \mathbb{R}^{n \times n}$ by adding suitable constants to the rows and columns and by permuting the columns (or rows), e.g. using the Hungarian method for solving the assignment problem for A. We say that a matrix A is *equivalent to* a matrix B if A can be obtained from B by adding constants to the rows or columns and by permuting the rows or columns. Thus every matrix is equivalent to a normal matrix. Not every matrix is equivalent to a strictly normal matrix. Note that if A is normal then the set of all optimal permutations of the assignment problem for A can conveniently be described: $ap(A) = \{\pi \in P_n; a_{i,\pi(i)} = 0$ for

all $i \in N\}$. We say that A is *max-algebraically definite* (or, shortly, *definite*) if $a_{ii} = 0$ for all $i \in N$ and D_A contains no positive cycles. Clearly $id \in ap(A)$ if A is definite.

Theorem 1. [2] *Let $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ be definite. If $x = (x_1, ..., x_n)^T \in \mathbb{R}^n$ is a vector of longest distances from all nodes to any fixed node in D_A then the matrix $(a_{ij} + x_j - x_i)$ is normal.*

Theorem 2. [2] *$A \in \mathbb{R}^{n\times n}$ is equivalent to a strictly normal matrix if and only if $|ap(A)| = 1$.*

By V we denote the max-hull of the vectors of longest distances from all nodes to a fixed node in D_A.

Theorem 3. [2] *Let $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ be definite. A is equivalent to a strictly normal matrix if and only if $int(V) \neq \emptyset$. If $x = (x_1, ..., x_n)^T \in int(V)$ then the matrix $(a_{ij} + x_j - x_i)$ is strictly normal.*

2.b) Another link between the assignment problem and the longest-distances problem. It is easily seen that if $id \in ap(A)$ for some matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ then $B = (a_{ij} - a_{ii})$ is definite. We will call B the *definite form* of A. If A is a definite matrix then it can be considered as the direct-distances matrix between all pairs of nodes in D_A. Note that the longest-distances matrix can max-algebraically be expressed as $A^{(n-1)}$.

Theorem 4. [6] *Let $A \in \mathbb{R}^{n\times n}$ and B and C be the definite forms of any two matrices B' and C' arising from A by permuting the columns so that $id \in ap(B') \cap ap(C')$. Then the longest-distances matrices of B and C coincide.*

2.c) The maximum cycle mean. Let $A = (a_{ij}) \in \mathbb{R}^{n\times n}$. If $\sigma = (i_1, ..., i_k, i_1)$ is a cycle in D_A then its *mean* is

$$\mu(\sigma, A) = \frac{a_{i_1 i_2} + a_{i_2 i_3} + ... + a_{i_k i_1}}{k}.$$

The value $\lambda(A) = \max_\sigma \mu(\sigma, A)$ is called the *maximum cycle mean* of A. Next statement is an immediate corollary of the max-algebraic spectral theory.

Theorem 5. *$\lambda(A^{(k)}) = (\lambda(A))^{(k)}$, for all natural k and any $A \in \overline{\mathbb{R}}^{n\times n}$.*

3. Some open problems.

3.a) Even [odd] parity assignment problem (EPAP [OPAP]): Let P_n^+ $[P_n^-]$ be the set of all even [odd] permutations of the set N. Given $A \in \mathbb{R}^{n\times n}$, find a permutation $\pi^* \in P_n^+$ $[\pi^* \in P_n^-]$ such that $w(A, \pi^*) = \max_{\pi\in P_n^+} w(A, \pi)$ $\left[= \max_{\pi\in P_n^-} w(A, \pi)\right]$. Obviously one of these two problems is always solved by solving the assignment problem. No polynomial method is known in general for solving both problems at the same time. The question whether $\max_{\pi\in P_n^+} w(A, \pi) = \max_{\pi\in P_n^-} w(A, \pi)$ is equivalent to the even cycle problem [2] and if this equality holds then optimal solutions to both problems can be found in $O(n^3)$ time. Some polynomially solvable special cases are studied in [3].

3.b) Strong linear independence. This concept is equivalent to the question: Given $A \in \mathbb{R}^{m\times n}$, $m > n$, is there an $n \times n$ submatrix B of A such that $|ap(B)| = 1$? For $m = n$ it reduces to checking $|ap(A)| = 1$, which can be done in $O(n^2)$ time after solving the assignment problem. Also, it is polynomially solvable for $m \times n$ $0-1$ matrices.

3.c) (Max-algebraic) rank. This is a generalisation of the previous problem important for applications in algebraic geometry [7]: Given $A \in \mathbb{R}^{m\times n}$, find the biggest natural number k for which there is a $k \times k$ submatrix B of A such that $|ap(B)| = 1$.

3.d) Coefficients of a (max-algebraic) characteristic polynomial. Given $A \in \overline{\mathbb{R}}^{n\times n}$ and $k < n$, find a $k \times k$ principal submatrix of A whose optimal assignment problem value is maximal (notation δ_k). No polynomial algorithm is known in general. A polynomial randomised algorithm exists [1]. Biggest k for which δ_k is finite can be found in $O(n^3)$ time [1]. An $O(n(m + n\log n))$ algorithm for finding all δ_k corresponding to so called essential terms exists [5] (here m is the number of finite entries of A). The problem arising after removing "principal" is easily solvable in $O(n^3)$ time [1].

3.e) Special case of the previous problem for matrices over $\{0, -\infty\}$. Given a digraph D with n nodes and $k < n$, are there pairwise node-disjoint cycles in D with exactly k nodes in total? No polynomial algorithm is known in general. Polynomially solvable for a number of special cases, including symmetric matrices and k even [4].

References

[1] Burkard, R.E. and P. Butkovič, Finding all essential terms of a characteristic maxpolynomial, *Discrete Applied Mathematics* 130 (2003)367-380.

[2] Butkovič, P., Max-algebra: the linear algebra of combinatorics?, *Linear Algebra and Applications* **367** (2003) 313-335.

[3] Butkovič, P., A note on the parity assignment problem. Submitted to Optimization. Preprint 2004/10, University of Birmingham.

[4] Butkovič, P. and S. Lewis, On the job rotation problem. Submitted to Discrete Optimization. Preprint 2005/17, University of Birmingham.

[5] Gassner, E., Variants of the Assignment and of the Transportation Problem, PhD thesis, Graz University of Technology, 2004.

[6] Sergeev, S., Max-plus definite closures and their eigenspaces, preprint.

[7] Sturmfels, B., On the tropical rank of a matrix, (with F. Santos and M. Develin), to appear in Discrete and Computational Geometry, (eds. J.E. Goodman and J. Pach), Mathematical Sciences Research Institute Publications, Cambridge University Press, 2005.

Deterministic online optical call admission revisited

Elisabeth Gassner
(joint work with Sven O. Krumke)

An instance of the *Online Call Admission Problem in Optical Networks* (OCA) consists of an undirected graph $G = (V, E)$ together with a set of χ eligible wavelengths $W = \{1, \ldots, \chi\}$ and a finite request sequence $\sigma = r_1, r_2, \ldots, r_m$ of calls. Each of the wavelengths in W is available once per edge. A *lightpath* is a pair (P, λ), where P is a path in G and λ is one of the wavelengths in W. In the sequel, we will use the terms wavelength and color interchangeably.

A *call* $r_j = (s_j, t_j)$ specifies the nodes $s_j \in V$ and $t_j \in V$ to be connected. Upon arrival of a new request $r_j = (s_j, t_j)$, an algorithm for OCA must decide whether to route or to reject r_j. If the call is accepted, the algorithm must provide a lightpath, thereby obeying the wavelength conflict constraint. Once accepted, a call can not be preempted: the lightpaths used for the call can not be changed or removed anymore. Each accepted call r_j contributes a benefit of one to the total profit obtained by an algorithm. The overall goal of OCA is to maximize the overall profit, that is, the total accepted demand.

A deterministic online algorithm ALG for OCA is *c-competitive* if for any request sequence σ the inequality $\text{ALG}(\sigma) \geq \frac{1}{c} \cdot \text{OPT}(\sigma)$ holds.

A result by Awerbuch et al. [1] states that a c-competitive algorithm for OCA with one wavelength, i.e., $\chi := |W| = 1$, implies a $(c+1)$-competitive algorithm for general numbers of wavelengths. We present the first deterministic competitive algorithms on the $(n+1)$ node line for OCA with $\chi > 1$ wavelengths which beat the linear competitive ratio that would be obtained by blindly applying the result by Awerbuch. More specifically, we present a $\chi(\sqrt[\chi]{n} + 2)$-competitive algorithm. For any fixed $\chi > 1$, this bound is sublinear in n.

We complement our results in establishing a lower bound of $\chi(\sqrt[\chi]{n} - 1)$ on the competitive ratio of any deterministic algorithm for OCA on the line with $(n+1)$ nodes and χ wavelengths.

An Algorithm with Sublinear Competitive Ratio:

Let $P = (V, E)$ be the node line with $(n+1)$ vertices, $V = \{v_0, \ldots, v_n\}$, and edge set $E = \{[v_i, v_{i+1}] : i = 0, \ldots, n\}$. Moreover, let W with $|W| = \chi$ be the set of wavelengths which we assume to be available on each edge of P. Suppose that $\sigma = r_1, r_2, \ldots, r_m$ is a sequence of requests which are subject to the call-admission problem. Each request $r_i = (s_i, t_i)$ uniquely determines a path in P between s_i and t_i. We will call the length of this path (measured in the number of edges) the *length* of the call length(r_i).

Intuitively, a good online algorithm should try to accept and route preferably "short" calls, since a short call does not block as many potential future calls as a longer one.

Our algorithm GETSHORTYworks as follows: Let r be a new call.

- If r can not be routed on any wavelength we reject r.

- Else, determine the smallest wavelength $\lambda \in W = \{1, \ldots, \chi\}$ such that r can be routed on λ.
- If length$(r) \leq \ell(\lambda)$, accept r and route r on wavelength λ, else reject it.

Our main result of this section is the following theorem:

Theorem 1. *Algorithm* GETSHORTY$_\ell$ *equipped with threshold function* $\ell(\lambda) := n^{\frac{\chi+1-\lambda}{\chi}}$ *achieves a competitive ratio of* $\chi(\sqrt[\chi]{n}+2)$ *for the* OCA *on an* $(n+1)$ *node line with* χ *wavelengths.*

The remainder of this section is dedicated to the proof of Theorem 1.

Fix a request sequence $\sigma = r_1, r_2, \ldots, r_m$ which contains at least one request. We denote by GETSHORTY$[\sigma]$ the set of calls routed by algorithm GETSHORTY$_\ell$ and by GETSHORTY$(\sigma) :=$ |GETSHORTY$[\sigma]$| its cardinality. Also, let OPT be an optimal offline algorithm for OCA.

We say that GETSHORTY$_\ell$ uses a wavelength λ on edge $e \in E$, if a call of σ is routed on e on wavelength λ. For $L \subseteq W = \{1, \ldots, \chi\}$ we denote by E_L the edges in E on which exactly the wavelengths in L are used by GETSHORTY$_\ell$, that is,

$$E_L = \{e \in E : \text{GETSHORTY}_\ell \text{ uses exactly the wavelengths in } L \text{ on } e\}.$$

Let us examine the solution GETSHORTY$[\sigma]$. Fix λ. Then, the total length of calls which are routed by GETSHORTY$_\ell$ on wavelength λ is given by

$$\sum_{L \subseteq W : \lambda \in L} |E_L| \leq n.$$

The first call r_1 in σ can be routed on wavelength 1. For $\lambda = 2, \ldots, \chi$, every call routed by GETSHORTY on wavelength λ has length at most $\ell(\lambda) = n^{(\chi+1-\lambda)/\chi}$, thus we get that

$$\begin{aligned} \text{GETSHORTY}(\sigma) &\geq 1 + \sum_{L \subseteq W} |E_L| \sum_{\lambda \in L : \lambda \neq 1} \frac{1}{\ell(\lambda)}. \\ &=: \sum_{L \subseteq W} b_L \end{aligned}$$

We now consider an optimal solution OPT$[\sigma]$ and partition it into three pairwise disjoint sets: OPT$[\sigma] = X \cup Y \cup Z$ where

- X is the set of calls $r \in$ OPT$[\sigma] \setminus$ GETSHORTY$[\sigma]$ such that r uses only edges of a single set E_L for some $L \subseteq W$.
- Y is the set of calls $r \in$ OPT$[\sigma] \setminus$ GETSHORTY$[\sigma]$ such that r uses edges of at least two sets E_L and $E_{L'}$ for $L, L' \subseteq W$.
- Z is the set of calls $r \in$ OPT$[\sigma] \cap$ GETSHORTY$[\sigma]$.

Lemma 1. *Let* $L \subseteq \{2, \ldots, \chi\}$ *be a subset of wavelengths that does not contain the first wavelength. Then there does not exist any call* $r \in X$ *that uses only edges of* E_L.

In order to bound the number of calls in $Y \cup Z$ we use a charging scheme.

First consider the requests in the set Y, that is, the set of calls $r \in \text{OPT}[\sigma] \setminus \text{GETSHORTY}[\sigma]$ such that r uses edges of at least two sets E_L and $E_{L'}$ for $L, L' \subseteq W$. Since any $r \in Y$ uses at least two different kinds of edges, there exists at least one call $r' \in \text{GETSHORTY}[\sigma]$ such that either the start vertex or the end vertex of r' and the corresponding start or end edge of r' is on the path of r. We assign r to r'.

If $r \in Z$, then r was accepted by GETSHORTY. We assign r to itself, that is, to $r' := r \in \text{GETSHORTY}[\sigma]$ and, again, either the start vertex or the end vertex of r' and the corresponding final edge is on the path of r.

Observe, that there are at most 2χ calls $r \in Y \cup Z$ that can be assigned to a call $r' \in \text{GETSHORTY}[\sigma]$ in the above charging scheme, because there are χ wavelengths and one start- and one end-edge of r'. This allows us to conclude that

$$|Y| + |Z| \leq 2\chi \cdot \text{GETSHORTY}(\sigma)$$

holds.

The remainder of this section is dedicated to bounding the term $\frac{|X|}{\sum_{L \subseteq W} b_L}$ appropriately. For $L \subseteq W$ we denote by x_L the number of calls $r \in X$ that use only edges of set E_L.

For $L = \{1, \ldots, \chi\}$ we can bound x_L from above by $x_L \leq \chi |E_L|$. Now let $L \subset W$ be of the form $L = \{1, \ldots, j\} \cup L'$ where $L' \subseteq \{j+2, \ldots, \chi\}$ for some $j = 1, \ldots, \chi - 1$, i. e., all wavelengths $\lambda = 1, \ldots, j$ are used by calls in $\text{GETSHORTY}[\sigma]$ and $j+1$ is the first wavelength that is not used by any call in $\text{GETSHORTY}[\sigma]$. Hence, wavelength $j+1$ is available on every edge in E_L.

Let $r \in X$ be a call that uses only edges of E_L. Notice that r could be routed by GETSHORTY on wavelength $j+1$. The only reason why GETSHORTY rejected r must be its length. We conclude that $\text{length}(r) > \ell(j+1)$ holds for every $r \in X$ that uses only edges of E_L. This gives us:

$$x_L \leq \chi \frac{|E_L|}{\ell(j+1)} \text{ for } L = \{1, \ldots, j\} \cup L', \text{ where } L' \subseteq \{j+2, \ldots, \chi\}.$$

Some calculations using the bounds on x_L and b_L yield

$$\frac{\text{OPT}(\sigma)}{\text{GETSHORTY}(\sigma)} \leq 2\chi + \frac{\sum_{L \subseteq W} x_L}{\sum_{L \subseteq W} b_L} \leq \chi(\sqrt[\chi]{n} + 2).$$

This completes the proof of the main theorem.

References

[1] B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosen. On-line competitive algorithms for call admission in optical networks. In: *Proceedings of the 4th Annual European Symposium on Algorithms*, volume **1136** (1996), 431–444.

Real-time AGV routing

Björn Stenzel
(joint work with Ewgenij Gawrilow, Ekkehard Köhler, Rolf H. Möhring)

Nowadays automation in logistic systems is very popular. In such an automated logistic system Automated Guided Vehicles (AGVs) are used for transportation tasks. The control of these AGVs is the key to an efficient transportation system.

We consider the online problem where requests appear sequentially and one must answer each request without having any information about later arriving requests. We extend the approaches of Huang, Palekar and Kapoor [2] and Kim and Tanchoco [3], respectively. In particular, we take physical properties of the AGVs into consideration in a more exact and flexible way and use an dynamic routing approach to obtain an efficient algorithm for the underlying problem. The algorithm avoids collisions, deadlocks and livelocks already at the time of route computation (conflict-free routing).

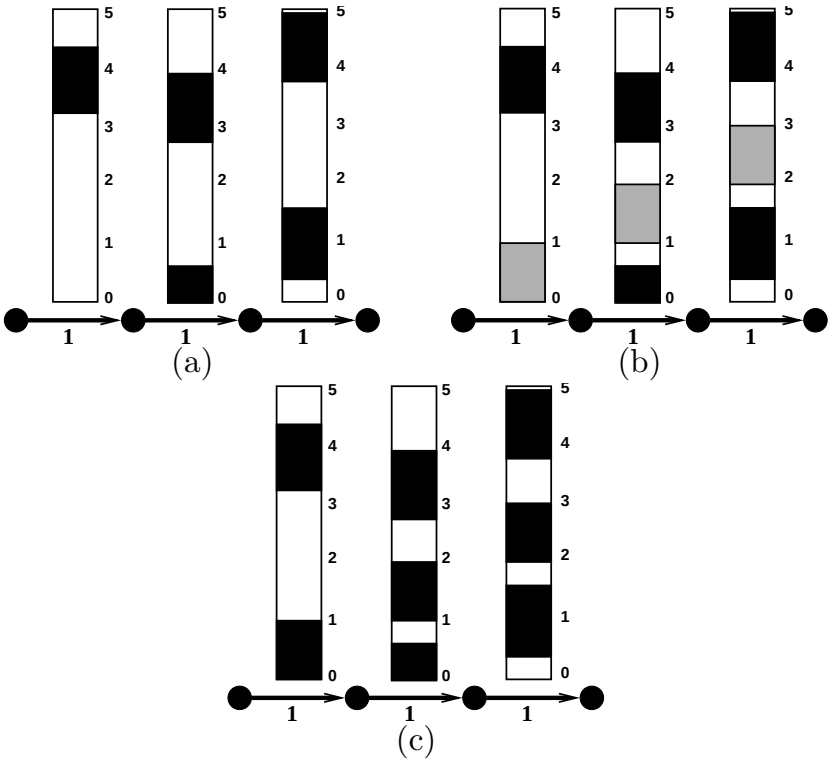


Figure 1. Real-time computation. (a) shows the situation before the new request arrives. There is a graph with some blockings (black) and some time-windows (white). The task is to compute a quickest path that respects the time-windows. This is illustrated in (b). The chosen path is blocked afterwards (see (c)).

The physical properties, i.e. the possible movements and the dimensions of the AGVs, are taken into consideration in a preprocessing step.

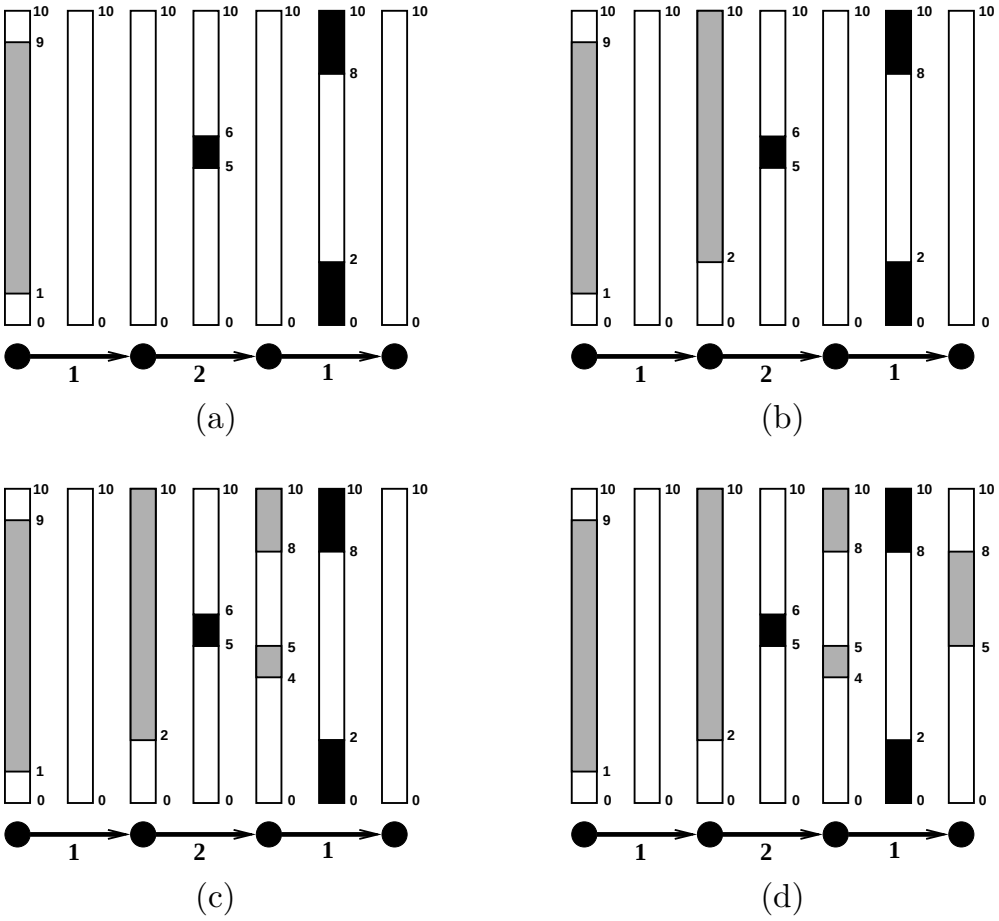


FIGURE 2. Label Expansion. The label intervals are represented by grey bars (nodes). The blockings are colored black (arcs). The white intervals between these blockings are the time-windows. The figures (a) to (d) show the successive expansion of the label intervals.

For the real-time computation we use a dynamic routing approach. We use time-windows on arcs to model the time dependencies in the underlying directed graph. Each time-window represents a free time slot at the corresponding arc depending on the routes of the AGVs that are already computed (see Fig. 1). For each request our algorithm computes a shortest path with respect to the current set of time-windows. In general, the Shortest Path Problem with Time-Windows [1] is NP-hard, but in our setting we get a polynomial time generalized dijkstra algorithm by carrying an interval in each label (the expansion of an label interval is shown in Fig. 2).

Two questions certainly arise concerning the dynamic (conflict-free) routing approach. Does this approach lead to better results than the approaches currently used in praxis? Is the algorithm suitable for real-time computation? Fortunately, we can answer these questions in the affirmative.

TABLE 2. Computational times (in sec.).

Scenarios	Comp. per request		Search		Readjustment	
	maximal	∅	maximal	∅	maximal	∅
25-1G-L (25 AGVs)	0.35	0.10	0.32	0.08	0.04	0.02
25-1G-S (25 AGVs)	0.14	0.06	0.11	0.04	0.03	0.02
25-2G-L (25 AGVs)	0.24	0.06	0.24	0.05	0.03	0.01
25-2G-S (25 AGVs)	0.25	0.06	0.24	0.05	0.02	0.01
25-3G-L (25 AGVs)	0.29	0.06	0.27	0.05	0.04	0.01
25-3G-S (25 AGVs)	0.23	0.06	0.18	0.05	0.04	0.01
25-4G-L (25 AGVs)	0.18	0.04	0.16	0.03	0.03	0.01
25-4G-S (25 AGVs)	0.18	0.05	0.16	0.04	0.02	0.01
50-1G-L (50 AGVs)	0.35	0.10	0.31	0.08	0.04	0.02
50-1G-S (50 AGVs)	0.23	0.07	0.20	0.05	0.04	0.02
50-2G-L (50 AGVs)	0.32	0.06	0.30	0.05	0.04	0.01
50-2G-S (50 AGVs)	0.16	0.06	0.13	0.04	0.04	0.01
100G-L (100 AGVs)	0.26	0.06	0.23	0.05	0.05	0.01
100G-S (100 AGVs)	0.23	0.06	0.20	0.04	0.05	0.01

In order to measure the performance of the computed routes we consider the sum of all transit times (all requests). We compare these overall transit times with a static approach used in Container Terminal Altenwerder (CTA) at Hamburg Harbour. The comparison shows that the conflict-free approach is superior to the static one in scenarios with heavy traffic (many AGVs) which clearly indicates the potential of this approach.

Using goal-oriented search the computation of a route in a graph with about 30.000 arcs and up to 100 AGVs take not more than some hundredth of a second on the average[1] (see Tab. 2) which is suitable for real-time computation.

REFERENCES

[1] Desrosiers et al., (1986) *Methods for routing with time windows*, European Journal of Operations Research **23** (1986), 236–245

[2] Huang, J., Palekar, U. S., Kapoor, S., *A labeling algorithm for the navigation of automated guided vehicles*, Journal of engineering for industry **115** (1993), 315–321

[3] Kim, Ch. W., Tanchoco, J. M. A., *Conflict-free shortest-time bidirectional agv routing*, International Journal of Production Research **29(12)** (1991), 2377–2391

[1]Hardware: AMD-Athlon 2100+ (1,7 Mhz) with 512 MB RAM.

Scheduling an air-taxi service

George Nemhauser

New developments in avionics and airplane manufacturing have brought about a new technology called the very light jet (VLJ). Weighing less than 10,000 pounds, these aircraft can carry 4-5 passengers, fly distances of over a 1,000 miles, reach altitudes of 19,000-30,000 feet, and travel at speeds between 350-390 nautical miles per hour (almost twice the altitude and speed of current turbo-prop airplanes). Priced at approximately 1 million US dollars, these jets cost about one fourth of the price of the cheapest business jets sold today. Additionally, these jets can be operated by a single pilot, which means low operational costs. Several manufacturers are taking orders for VLJs with Eclipse Aviation [7] being the first with planned deliveries in March of 2006.

The availability of relatively cheap small jet aircrafts suggests a new air transportation business: the air taxi, an on-demand service in which travelers call one day or a few days in advance to schedule transportation. The advantages of such a system are obvious. An air-taxi service gives regional travelers the option of hopping aboard small jets that fly to and from less congested outlying airports, without packing parking lots, long lines at security checkpoints, flight delays, and lost luggage, that are closer to where they live and where they want to go. In fact, VLJs could land at any of 14,000 private and public landing strips in the United States. By charging a discount fare for sharing cabin space with other passengers, aggregation can greatly reduce costs while still ensuring a very convenient service.

The idea of an air-taxi service to satisfy regional demand is rapidly becoming a reality. In fact, even though VLJs are not yet available, air taxi services already exist today. Since April 2002, SkyTaxi Inc. [12] has been providing on-demand air transportation in the northwestern United States. Likewise, SATSair [10] provides such a service in the northeast. And recently, DayJet Corporation [5] has announced that they will be providing air-taxi services in the southeast starting in mid-2006 using the Eclipse jet. All of them plan to expand their services to the entire United States. The air-taxi alternative has generated a lot of public interest. For example, The New York Times [8] devoted a major article to it this past July.

To effectively manage day-to-day operations at an air taxi service, several optimization-based scheduling components need to be employed:

(1) an online scheduling system to quickly inform passengers if their air transportation requests can be serviced and at what price,
(2) an off-line scheduling system to construct minimum cost pilot and jet itineraries for the next day once the reservation deadline has passed, and
(3) a disruption management system to re-route jets in case of failures, bad weather, or other unpredictable events.

In this paper, we present an optimization engine that forms a core component of the off-line scheduling system being developed at the DayJet Corporation. This system resembles a dial-a-ride [2, 3] or pick-up and delivery system [11] and is

concerned with the scheduling of a set of requests for air transportation during a single day. A request specifies an origin airport, an earliest acceptable departure time at the origin, a destination airport, a latest acceptable arrival time at the destination, a number of passengers, and a weight. A fleet of jet airplanes, each operated by a single pilot, is available to provide the requested air transportation. Each jet has a home base, a seat capacity limiting the number of passengers that can be accommodated, and a weight capacity limiting the weight that can be accommodated. Each jet is available for a certain period during the day, and has to return to its home base at the end of the day. A set of pilots, stationed at the home bases of the airplanes, is available to fly the jets. A pilot departs from the home base where he is domiciled at the start of his duty and returns to the home base where he is domiciled at the end of his duty. A pilot schedule has to satisfy FAA regulations governing flying hours and duty periods, i.e., a single pilot cannot fly more than 8 hours in a day and his duty period cannot be more than 14 hours. To ensure acceptable service an itinerary for a passenger will involve at most two flights, i.e., only a single intermediate stop is allowed. If there is an intermediate stop, both flights have to be on the same jet, i.e., no airplane changes are allowed. A turnaround time at an airport, i.e., the minimum time between an arrival at an airport and the next departure, is given. The objective is to minimize the costs, while satisfying all requests and respecting all constraints. A dispatcher has to decide which jets and pilots to use to satisfy the requests and what the jet and pilot itineraries will be, i.e., the flight legs and associated departure times.

We describe an integer multi-commodity network flow [1] model for the air taxi scheduling problem. We present techniques for simplifying the network to ensure that the resulting network is of manageable size. We give a large neighborhood search algorithm [9] for extending the methodology to solve problems with up to 300 planes. We also sketch a column generation model [2] that gives a tighter LP bound than the multi-commodity flow model.

References

[1] Ahuja, R.K., T.L. Magnanti, J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey, 1993.

[2] Barnhart, C., E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance. "Branch-and-Price: Column Generation for Solving Huge Integer Programs." *Operations Research*.

[3] Cordeau, J.F. "A Branch-and-Cut Algorithm for the Dial-a-Ride Problem." *Operations Research*, forthcoming.

[4] Cordeam, J.F., G. Laporte. "A Tabu Search Heuristic for the Static Multi-Vehicle Dial-a-Ride Problem." *Transportation Research B*, 37: 579-594, 2003.

[5] DayJet Corporation. www.dayjet.com

[6] Dumas, Y., J. Desrosiers, F. Soumis. "The Pickup and Delivery Problem with Time Windows." *European Journal of Operations Research*, 54: 7-22, 1991.

[7] Eclipse Aviation. www.ecpliseaviation.com

[8] "Fly Me to the Moon? No, but the Next Best Thing." The New York Times, July 10, 2005.

[9] Ropke, S., D. Pisinger. "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows." Technical Report 2004-13. DIKU, University of Copenhagen, 2004.

[10] SATSair. www.satsair.com

[11] Savelsbergh, M.W.P., M. Sol. "The General Pickup and Delivery Problem." *Transportation Science* 29: 17-29. 1995.
[12] Sky Taxi. www.skytaxi.com

Reporters: Johannes Hatzl and Nitin Ahuja

Participants

Prof. Dr. Karen Aardal

CWI - Centrum voor Wiskunde en Informatica
Postbus 94079
NL-1090 GB Amsterdam

Dr. Nitin Ahuja

Institut für Mathematische Optimierung
TU Braunschweig
Pockelsstr. 14
38106 Braunschweig

Prof. Dr. Kurt M. Anstreicher

Department of Management Sciences
The University of Iowa
Iowa City IA 52242-1000
USA

Dr. Sanjeev Arora

Department of Computer Science
Princeton University
35 Olden Street
Princeton, NJ 08544-2087
USA

Prof. Dr. Peter Butkovic

School of Mathematics
The University of Birmingham
Edgbaston
GB-Birmingham B15 2TT

Prof. Dr. Alberto Caprara

DEIS
Universita di Bologna
Viale Risorgimento, 2
I-40136 Bologna

Prof. Dr. Michelangelo Conforti

Dipartimento di Matematica Pura ed Applicata
Universita di Padova
Via Belzoni, 7
I-35131 Padova

Prof. Dr. William J. Cook

Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
USA

Prof. Dr. Gerard Cornuejols

Tepper School of Business
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
USA

Prof. Dr. William H. Cunningham

Dept. of Combinatorics and Optimization
University of Waterloo
200 University Ave.
Waterloo, Ontario N2L 3G1
CANADA

Prof. Dr. Sandor P. Fekete

Institut für Mathematische
Optimierung
TU Braunschweig
Pockelsstr. 14
38106 Braunschweig

Prof. Dr. Andras Frank

Department of Operations Research
Eötvös Lorand University
ELTE TTK
Pazmany Peter setany 1/C
H-1117 Budapest

Prof. Dr. Satoru Fujishige

RIMS
Kyoto University
Sakyo-ku
Kyoto 606-8502
JAPAN

Dr. Elisabeth Gassner

Institut für Mathematik
Technische Universität Graz
Steyrergasse 30
A-8010 Graz

Prof. Dr. Michel Goemans

Department of Mathematics
Massachusetts Institute of
Technology
Cambridge, MA 02139-4307
USA

Prof. Dr. Peter Gritzmann

Zentrum Mathematik
Kombinatorische Geometrie (M9)
Technische Universität München
Boltzmannstr. 3
85747 Garching bei München

Prof. Dr. Bertrand Guenin

Dept. of Combinatorics and
Optimization
University of Waterloo
200 University Ave.
Waterloo, Ontario N2L 3G1
CANADA

Prof. Dr. Horst W. Hamacher

Fachbereich Mathematik
Universität Kaiserslautern
67653 Kaiserslautern

Ronny Hansmann

Institut für Mathematische
Optimierung
TU Braunschweig
Pockelsstr. 14
38106 Braunschweig

Dr. Johannes Hatzl

Institut für Mathematik B
Technische Universität Graz
Steyrergasse 30
A-8010 Graz

Prof. Dr. Christoph Helmberg

Fakultät für Mathematik
Technische Universität Chemnitz
09107 Chemnitz

Benjamin Hiller

Konrad-Zuse-Zentrum für
Informationstechnik Berlin (ZIB)
Takustr. 7
14195 Berlin

Prof. Dr. Dorit S. Hochbaum

Department of Industrial
Engineering and Operations Research
University of California
Etcheverry Hall
Berkeley, CA 94720-1777
USA

Prof. Dr. Michael Jünger

Institut für Informatik
Universität zu Köln
Pohligstr. 1
50969 Köln

Dr. Bettina Klinz

Institut für Mathematik B
Technische Universität Graz
Steyrergasse 30
A-8010 Graz

Prof. Dr.Dr.h.c. Bernhard Korte

Forschungsinstitut für
Diskrete Mathematik
Universität Bonn
Lennestr. 2
53113 Bonn

Stefan Krause

Institut für Mathematische
Optimierung
TU Braunschweig
Pockelsstr. 14
38106 Braunschweig

Retsef Levi

IBM Corporation
Thomas J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598
USA

Prof. Dr. Thomas M. Liebling

EPFl SB IMA-ROSO
MA-C1 547
Station 8
CH-1015 Lausanne

Dr. Marco E. Lübbecke

Institut für Mathematik - Fak. II
Technische Universität Berlin
Sekr. MA 6-1
Straße des 17. Juni 136
10623 Berlin

Prof. Dr. Alexander Martin

FB Mathematik, AG 7
TU Darmstadt
Schlossgartenstr. 7
64289 Darmstadt

Prof. Dr. S. Thomas McCormick

Sauder School of Business
University of British Columbia
2053 Main Mall
Vancouver, B.C. V6T 1Z2
CANADA

Prof. Dr. Rolf Möhring

Institut für Mathematik - Fak. II
Technische Universität Berlin
Sekr. MA 6-1
Straße des 17. Juni 136
10623 Berlin

Prof. Dr. Petra Mutzel

Universität Dortmund
Informatik LS 11
Lehrstuhl für Algorithm Engineering
Josef-von-Fraunhofer-Str.20
44227 Dortmund

Prof. Dr. George L. Nemhauser

Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
USA

Prof. Dr. Gyula Pap

Department of Operations Research
Eötvös Lorand University
ELTE TTK
Pazmany Peter setany 1/C
H-1117 Budapest

Prof. Dr. Ulrich Pferschy

Institut für Statistik und
Operations Research
Universität Graz
Universitätsstraße 15
A-8010 Graz

Prof. Dr. Maurice Queyranne

Sauder School of Business
University of British Columbia
2053 Main Mall
Vancouver, B.C. V6T 1Z2
CANADA

Prof. Dr. Gerhard Reinelt

Institut für Informatik
Ruprecht-Karls-Universität
Heidelberg
Im Neuenheimer Feld 368
69120 Heidelberg

Prof. Dr. Franz Rendl

Institut für Mathematik
Universität Klagenfurt
Universitätsstr. 65-67
A-9020 Klagenfurt

Dr. Günter Rote

Institut für Informatik
Freie Universität Berlin
Takustr. 9
14195 Berlin

Prof. Dr. Alexander Schrijver

Centrum voor Wiskunde en
Informatica
Kruislaan 413
NL-1098 SJ Amsterdam

Prof. Dr. Rüdiger Schultz

FB Mathematik
Universität Duisburg-Essen
Lotharstr. 65
47057 Duisburg

Prof. Dr. Andreas S. Schulz

Sloan School of Management
Massachusetts Institute of
Technology
77, Massachusetts Avenue
Cambridge, MA 02139-4307
USA

Michael Schulz

Institut für Informatik
Universität zu Köln
Pohligstr. 1
50969 Köln

Prof. Dr. David B. Shmoys

Cornell University
232 Frank H.T. Rhodes Hall
Ithaca NY 14853
USA

Prof. Dr. Martin Skutella

Fachbereich Mathematik
Universität Dortmund
44221 Dortmund

Prof. Dr. Clifford Stein

Dept. of IEOR
Columbia University
MC 4704
500 West 120 St.
New York NY 10027-6699
USA

Björn Stenzel

Fachbereich Mathematik - FB 3
Technische Universität Berlin
MA 6-1
Straße des 17. Juni 135
10623 Berlin

Dr. Chaitanya Swamy

Center for the Mathematics of Inf.
338 Moore Laboratory, MC 136-93
Cal Tech
1200E. California Blvd.
Pasadena, CA 91125
USA

Prof. Dr. Jens Vygen

Forschungsinstitut für
Diskrete Mathematik
Universität Bonn
Lennestr. 2
53113 Bonn

Prof. Dr. Robert Weismantel

Fakultät für Mathematik
Otto-von-Guericke-Universität
Magdeburg
Universitätsplatz 2
39106 Magdeburg

Prof. Dr. David P. Williamson

School of Operations Research and
Industrial Engineering
Cornell University
Rhodes Hall
Ithaca, NY 14853
USA

Prof. Dr. Gerhard Woeginger

Faculty of Mathematical Sciences
University of Twente
P.O. Box 217
NL-7500 AE Enschede

Prof. Dr. Laurence A. Wolsey

Center for Operations Research and
Econometrics
Universite de Louvain
34 Voie du Roman Pays
B-1348 Louvain-la-Neuve

Prof. Dr. Uwe Zimmermann

Institut für Mathematische
Optimierung
TU Braunschweig
Pockelsstr. 14
38106 Braunschweig