Mathematisches Forschungsinstitut Oberwolfach

Report No. 51/2008

Combinatorial Optimization

Organised by
William Cook, Atlanta
András Frank, Budapest
Michael Jünger, Köln

November 9th – November 15th, 2008

Abstract. The field of Combinatorial Optimization brings together techniques from a variety of mathematical disciplines to study optimization problems over discrete structures. The field is very active and continues to make strong advances in theory, computation, and applications. This Oberwolfach Workshop in Combinatorial Optimization follows a long tradition of such meetings that have helped to enrich the field and to establish new or deepen existing research collaborations.

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

Introduction by the Organisers

Combinatorial Optimization remains a very lively discipline with strong connections to Combinatorics, Graph Theory, Geometry, and Integer Programming. For over thirty years, Oberwolfach workshops have had a central role in shaping the field, being the unique setting where the entire spectrum of the subject is covered, from fundamental theory to practical computation. The 2008 Combinatorial Optimization Workshop was again a great success. We would like to take this opportunity to thank all speakers for the great care they put into preparing their contributions, and to thank all participants for their active discussions that led to a truly stimulating meeting. Just like our predecessors, we are happy that we could attract many of the strongest theoreticians as well as the most successful practitioners to the workshop.

Even though the team of organizers has dynamically changed over the last thirty years, a format has emerged that has been quite stable in the last decade, namely a mix of a few pre-arranged focus talks and short presentations of the participants that are scheduled during the conference. In the focus talks, selected experts survey the state-of-the-art in current hot topics and point out new mathematical

challenges. While we have largely adopted this format, we have lowered the number of short presentations in favor of more time that can be spent for joint research in small groups, taking up a suggestion made by the director during the previous Combinatorial Optimization Workshop held in 2005. We have had very positive feedback from workshop participants that this measure enhances the conditions for new cooperation between participants that can be expected to lead to continued joint work and publications afterwards. Each day, the program started with a prearranged one hour focus presentation, followed by two thirty-minute presentations in the morning and three thirty-minute presentations before dinner.

This year's focus talks featured topics that bring together techniques from other areas of mathematics to explore new approaches for attacking well-known combinatorial problems. Tibor Jordán led off the workshop with a talk covering methods in network localization, making use of ideas from combinatorial rigidity. Tuesday's talk was by Kazuo Murota, discussing the new area of discrete convex analysis, where combinatorial analogues of basic convex results are being developed, mainly in the Japanese research community. Friedrich Eisenbrand presented new directions in the application of techniques from the geometry of numbers in Wednesday's talk, including several new open problems in the area of integer programming. On Thursday, Monique Laurent gave an in-depth treatment of the use of semidefinite programming techniques applied to coloring problems in graphs. András Sebő gave the final focus talk on Friday, describing the application of polyhedral techniques to graph optimization problems.

A new addition to this year's workshop was an evening session devoted to computational integer programming, featuring discussions led by Robert Bixby, Sanjeeb Dash, and Alexander Martin. The session highlights one of the remarkable achievements of the field in general and the Oberwolfach meetings in particular, namely, a very fast transition form basic theoretical breakthroughs to commercially successful software. Within the course of the three-year cycle of the Oberwolfach meetings, proposed new techniques work their way from academia to the leading mixed-integer programming software, delivered throughout the industrial sector. The broad mix of researchers at the Combinatorial Optimization Workshops directly contributes to this rapid transfer.

Workshop: Combinatorial Optimization

Table of Contents

Abstracts

Uniquely Localizable Sensor Networks

Tibor Jordán
(joint work with Bill Jackson)

In the network localization problem the locations of some nodes (called anchors) of a network as well as the distances between some pairs of nodes are known, and the goal is to determine the location of all nodes. This is one of the fundamental algorithmic problems in the theory of wireless sensor networks, see for example [1, 11].

A natural additional question is whether a solution to the localization problem is unique. The network, with the given locations and distances, is said to be *uniquely localizable* if there is a unique set of locations consistent with the given data. The unique localizability of a two-dimensional network, whose nodes are 'in generic position', can be characterized by using results from graph rigidity theory. In this case unique localizability depends only on the combinatorial properties of the network: it is determined completely by the *distance graph* of the network and the set of anchors, or equivalently, by the *grounded graph* of the network and the number of anchors. The vertices of the distance and grounded graph correspond to the nodes of the network. In both graphs two vertices are connected by an edge if the corresponding distance is explicitly known. In the grounded graph we have additional edges: all pairs of vertices corresponding to anchor nodes are adjacent. The grounded graph represents all known distances, since the distance between two anchors can be obtained from their locations. It is convenient to investigate localization problems with distance information by using frameworks, the central objects of rigidity theory.

A 2-dimensional *framework* is a pair (G,p), where $G=(V,E)$ is a graph and p is a map from V to $\mathbb{R}^2$. We consider the framework to be a straight line *realization* of G in $\mathbb{R}^2$. Two frameworks (G,p) and (G,q) are *equivalent* if corresponding edges have the same lengths, that is, if $||p(u)-p(v)||=||q(u)-q(v)||$ holds for all pairs u,v with $uv\in E$, where $||.||$ denotes the Euclidean norm in $\mathbb{R}^2$. Frameworks (G,p), (G,q) are *congruent* if $||p(u)-p(v)||=||q(u)-q(v)||$ holds for all pairs u,v with $u,v\in V$. This is the same as saying that (G,q) can be obtained from (G,p) by an isometry of $\mathbb{R}^2$. We shall say that (G,p) is *globally rigid*, or that (G,p) is a *unique realization* of G, if every framework which is equivalent to (G,p) is congruent to (G,p).

Proposition 1. [1, 11] *Let N be a network in $\mathbb{R}^2$ consisting of m anchors located at positions $p_1,...,p_m$ and $n-m$ ordinary nodes located at $p_{m+1},...,p_n$. Suppose that there are at least three anchors in general position. Let G be the grounded graph of N and let $p=(p_1,...,p_n)$. Then the network is uniquely localizable if and only if (G,p) is globally rigid.*

It is an NP-hard problem to decide if a given framework is globally rigid. The problem becomes more tractable, however, if we assume that there are no algebraic dependencies between the coordinates of the points of the framework.

A framework (G,p) is said to be *generic* if the set containing the coordinates of all its points is algebraically independent over the rationals. Restricting to generic frameworks gives us two important 'stability properties'. The first is that, if (G,p) is a globally rigid d-dimensional generic framework then there exists an $\epsilon > 0$ such that all frameworks (G,q) which satisfy $||p(v) - q(v)|| < \epsilon$ for all $v \in V$ are also globally rigid. The second is that if some 2-dimensional generic realization of a graph G is globally rigid, then all 2-dimensional generic realizations of G are globally rigid. Rigidity, which is a weaker property of frameworks than global rigidity, plays an important role in the exploration of the structural results of global rigidity. A framework (G,p) is *rigid* if there exists an $\epsilon > 0$ such that, if (G,q) is equivalent to (G,p) and $||p(u) - q(u)|| < \epsilon$ for all $v \in V$, then (G,q) is congruent to (G,p). It is equivalent to saying that (G,p) has no non-trivial continuous deformations which preserve the edge lengths.

Rigidity, like global rigidity, is a generic property of frameworks, that is, the rigidity of a generic realization of a graph G depends only on the graph G and not the particular realization. We say that the graph G is *rigid*, respectively *globally rigid* or *uniquely realizable*, in $\mathbb{R}^2$ if every (or equivalently, if some) generic realization of G in $\mathbb{R}^2$ is rigid, respectively globally rigid. See [3, 12] for a detailed survey of the rigidity of frameworks.

The characterization of rigid graphs in $\mathbb{R}^2$ is a result of Lovász and Yemini [9]. Globally rigid graphs in $\mathbb{R}^2$ are characterized by the following result, which incorporated earlier results from [2, 4] and a new inductive construction for the family of 3-connected redundantly rigid graphs. We say that G is *redundantly rigid* in $\mathbb{R}^2$ if $G - e$ is rigid in $\mathbb{R}^2$ for all edges e of G.

Theorem 2. [5] *Let (G,p) be a 2-dimensional generic framework. Then (G,p) is globally rigid if and only if either G is a complete graph on two or three vertices, or G is 3-connected and redundantly rigid in $\mathbb{R}^2$.*

We may also consider a more general problem in which either distances, directions, or both, are known for some pairs of vertices of a sensor network. In this situation we work with mixed graphs. A *mixed graph* is a graph together with a bipartition $D \cup L$ of its edge set. We refer to edges in D as *direction edges* and edges in L as *length edges*. A *mixed framework* (G,p) is a mixed graph $G = (V; D, L)$ together with a map $p : V \to \mathbb{R}^2$. Two mixed frameworks (G,p) and (G,q) are *equivalent* if $p(u) - p(v)$ is a scalar multiple of $q(u) - q(v)$ for all $uv \in D$ and $||p(u) - p(v)|| = ||q(u) - q(v)||$ for all $uv \in L$. The mixed frameworks (G,p) and (G,q) are *congruent* if there exists a vector $t \in \mathbb{R}^2$ and $\lambda \in \{-1, 1\}$ such that $q(v) = \lambda p(v) + t$ for all $v \in V$. This is equivalent to saying that (G,q) can be obtained from (G,p) by a rotation by 0 or 180 degrees and a translation. The mixed framework (G,p) is *globally rigid* if every framework which is equivalent to (G,p) is congruent to (G,p). It is *rigid* if there exists an $\epsilon > 0$ such that every framework (G,q) which is equivalent to (G,p) and satisfies $\|p(v) - q(v)\| < \epsilon$ for

all $v \in V$, is congruent to (G, p). The rigidity of mixed frameworks is a generic property and we may define a mixed graph G to be *rigid* if every, or equivalently, if some, generic realization of G is rigid. Rigid mixed graphs were characterized by Servatius and Whiteley [10].

The problem of characterizing when a generic mixed framework (G, p) is globally rigid is still an open problem. We have, however, been able to obtain some partial results. In order to state these in terms of graphs we define a mixed graph G to be *globally rigid* if all generic realizations of G are globally rigid. (It is not known whether global rigidity of mixed frameworks is a generic property.) A necessary condition for global rigidity, which is analogous to the '3-connectedness condition' of Theorem 2 is 'direction balancedness', which is defined as follows [8]. Let G be a 2-connected mixed graph. A 2-separation S of G is *direction-balanced* if each 'side' of S contains a direction edge. We say that G is *direction balanced* if all 2-separations of G are direction balanced.

We say that a mixed graph $G = (V; D, L)$ is a *mixed circuit* if $|D|+|L| = 2|V|-1$ and G is redundantly rigid. (These graphs are the circuits of the 'mixed rigidity matroid' of G.) We proved the following characterization of globally rigid mixed circuits.

Theorem 3. [7] *Let G be a mixed circuit. Then G is globally rigid if and only if G is direction balanced.*

For a detailed survey on graph theoretic techniques in the localization problem see [6].

References

[1] J. Aspnes, T. Eren, D.K. Goldenberg, A.S. Morse, W. Whiteley, Y.R. Yang, B.D.O. Anderson, and P.N. Belhumeur, *A theory of network localization*, IEEE Trans. on Mobile Computing **5(12)** (2006), 1663–1678.

[2] R. Connelly, *Generic global rigidity*, Discrete Comput. Geom. **33** (2005), 549–563.

[3] J. Graver, B. Servatius, and H. Servatius, *Combinatorial Rigidity*, AMS Graduate Studies in Mathematics **2** (1993).

[4] B. Hendrickson, *Conditions for unique graph realizations*, SIAM J. Comput. **21(1)** (1992), 65–84.

[5] B. Jackson and T. Jordán, *Connected rigidity matroids and unique realizations of graphs*, J. Combinatorial Theory, Ser. B. **94** (2005), 1–29.

[6] B. Jackson and T. Jordán, *Graph theoretic techniques in the analysis of uniquely localizable sensor networks*, in: *Localization Algorithms and Strategies for Wireless Sensor Networks* (B. Fidan, G. Mao, eds.), IGI Global (2008), to appear.

[7] B. Jackson and T. Jordán, *Globally rigid circuits of the two-dimensional direction-length rigidity matroid*, Egerváry Research Group TR-2008-09, submitted.

[8] B. Jackson and T. Jordán, *Operations preserving global rigidity of generic direction-length frameworks*, Egerváry Research Group TR-2008-08, submitted.

[9] L. Lovász and Y. Yemini, *On generic rigidity in the plane*, SIAM J. Algebraic Discrete Methods **3(1)** (1982), 91–98.

[10] B. Servatius and W. Whiteley, *Constraining plane configurations in CAD: Combinatorics of directions and lengths*, SIAM J. Discrete Math. **12** (1999), 136–153.

[11] A.M. So and Y. Ye, *Theory of semidefinite programming for sensor network localization*, Math. Program., Ser. B **109(2–3)** (2007), 367–384.

[12] W. Whiteley, *Some matroids from discrete applied geometry*, Matroid theory (Seattle, WA, 1995), 171–311, Contemp. Math., 197, Amer. Math. Soc., Providence, RI, 1996.

Separation, Dimension, and Facet Algorithms for Node Flow Polyhedra

S. Thomas McCormick
(joint work with Maren Martens and Maurice Queyranne)

Ball et al. [3] (see also [5]) propose a graph model for *material compatibility* constraints (see, e.g., Balakrishnan and Geunes [2]) for products such as PCs made from components, where each component is a node, and the natural variables are flows on paths (products). When the number of paths is much larger than the number of nodes, it is more convenient to consider the set of node flows in place of that of path flows. It is natural to want to extend this model in three directions: (1) What if we allow the graph to contain directed cycles, which allows for more modeling flexibility? (2) What if we have lower and upper bounds l and u on arc flow and λ and μ on node flow? (3) Can we find efficient algorithms that solve separation, validity, and dimension questions about this *node flow polytope*? This paper extends [3, 5] in all of these directions.

Since we now allow directed cycles, it turns out to be simpler to consider flows on cycles instead of flows on paths. We denote the set of directed cycles by $\mathcal{C}$, and denote the node-cycle incidence matrix by M. A production plan $y \in \mathbb{R}^{\mathcal{C}}$ naturally induces an arc flow f via $f_{ij} = \sum_{C \in \mathcal{C}: i \to j \in C} y_C$ and a node flow x via $x = My$. Given lower and upper bounds l_{ij}, u_{ij} on arc flow, and λ_i, μ_i on node flow with $-\infty \le l \le u \le \infty$ and $-\infty \le \lambda \le \mu \le \infty$, then y is a feasible production plan if its associated f and x satisfy $l_{ij} \le f_{ij} \le u_{ij}$ for all $i \to j \in A$, and $\lambda_i \le x_i \le \mu_i$ for all $i \in N$. (Non-zero arc lower bounds arise in computing face dimension, and so we include them from the start.) Letting $B = (l, u, \lambda, \mu)$ be the vector of bounds, define $Q^B = \{x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^{\mathcal{C}} \text{ s.t. } y \text{ is a feasible production plan and } x = My\}$, so that Q^B is the projection of the polyhedron of feasible production plans in the "big" space $\mathbb{R}^{\mathcal{C}}$ to the "small" space $\mathbb{R}^N$. The node flow polyhedron Q^B is likely to appear as a part of a more general model that is being solved by Branch and Cut. Here is a list of possible tasks that such a code may require, along with our algorithms for solving them:

(1) Separation: Given some $\bar{x} \in \mathbb{R}^n$, either prove that $\bar{x} \in Q^B$, or find some constraint $\alpha^T x \ge \beta$ that is satisfied by every $x \in Q^B$, but violated by $\bar{x}$, i.e., such that $\alpha^T \bar{x} < \beta$.

We solve this by splitting each node i into i and i' with a new arc $i \to i'$ with bounds $l_{ii'} = u_{ii'} = \bar{x}_i$. Then it is easy to see that $\bar{x} \in Q^B$ iff there is a feasible flow in this extended network. This can be decided using one max flow (see, e.g., [1]). When there is no feasible flow then as in Hoffman's Circulation Theorem [4] the associated min cut gives us disjoint subsets (possibly empty) of the (original) nodes J and K, and subsets

of the (original) arcs L and U, defining the *Hoffman cut* $x(J) + l(L) \leq x(K) + u(U)$.

(2) VALIDITY: Given a proposed constraint $\alpha^T x \geq \beta$, either prove that it is valid for all $x \in Q^B$, or find some $\bar{x} \in Q^B$ such that $\alpha^T \bar{x} < \beta$.

We solve this by using the same split-node network. This time we put cost α_i on new arc $i \to i'$ and cost 0 on all other arcs, and we do a min-cost circulation on the network that effectively minimizes $\alpha^T x$. If the optimal value $z^* \geq \beta$, then the cut is valid, else the optimal flow yields an $\bar{x}$ violating the cut.

(3) DIMENSION of Q^B: Compute the dimension of Q^B.

For $\mathrm{cc}(H)$ as the number of connected components induced by arcs H, define $\mathrm{cotree}(H) = \mathrm{cotree}(N_H, A_H) := |A_H| - |N_H| + \mathrm{cc}(H)$, i.e., the number of arcs in the complement of a spanning forest. Also define the *fixed* arcs as those whose flow is fixed to a bound by equal bounds or min cuts, and other arcs as *free*; the new arcs in the node-split graph are *split* arcs, and the others are *real* arcs. Then, since $\mathrm{cotree}(A_{\mathrm{real}\cap\mathrm{free}})$ represents the number of split arcs that must occur in any spanning forest, we show that $\dim(Q^B) = \mathrm{cotree}(A_{\mathrm{free}}) - \mathrm{cotree}(A_{\mathrm{real}\cap\mathrm{free}})$.

(4) FACE DIMENSION: Given some valid constraint $\alpha^T x \geq \beta$, compute the dimension of the face F of Q^B it induces. Solving this allows us to characterize which valid constraints are facets of Q^B.

Here we combine the techniques of the previous two items. We again use the node-split graph, and put cost α on arcs $i \to i'$, costs 0 elsewhere, and do min-cost flow. Reduced costs determine which arcs have their flows fixed to a bound in every optimal solution. By redefining A_{free} to exclude these new fixed arcs and using the algorithm of the previous section, we get the dimension of the face.

(5) SEPARATION TO A FACET: Given some $\bar{x} \notin Q^B$, find a constraint $\alpha^T x \geq \beta$ separating $\bar{x}$ from Q^B that induces a facet of Q^B.

If SEPARATION gives us a Hoffman cut that is not a facet, we use a min-cost flow with 0, ± 1 costs to find a feasible flow "nearest" to $\bar{x}$. By analyzing the connected components of this flow we can manipulate this cut so that it remains separating but has a nice structure. We then show that the only way this can avoid being a facet is if it contains a "blob" that causes extra real arcs to have fixed flow. We then show a final manipulation that preserves separation, and which increases the dimension of the induced face by at least one. The overall time for all these manipulations is $O(mn)$, which is dominated by the min-cost flow.

SEPARATION TO A FACET is a key subroutine needed for Branch and Cut, and turns out to be quite tricky. Note that we solve all problems by giving fast network flow-based algorithms. Since such algorithms are highly developed and very quick in practice (see, e.g., [1]), these results are quite practical.

References

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

[2] A. Balakrishnan and J. Geunes, *Requirements Planning with Substitutions: Exploring Bill-of-Materials Flexibility in Production Planning*, MSOM **2** (2000), 166–185.

[3] M.O. Ball, C.-Y. Chen, and Z.-Y. Zhao, *Material compatibility constraints for make-to-order production planning*, OR Letters **31** (2003), 420–428.

[4] A.J. Hoffman, *Some Recent Applications of the Theory of Linear Inequalities to Extremal Combinatorial Analysis*, in R. Bellman and M. Hall, Jr., (eds.), Proceedings of Symposia in Applied Mathematics, Vol. X, *Combinatorial Analysis*, American Mathematical Society, Providence, RI (1960), 113–127.

[5] S. T. McCormick and M. N. Queyranne, *Le Cône des Flots aux Noeuds dans un Réseau Acyclique*, Proceedings of *Journées sur les Polyèdres en Optimisation Combinatoire* at Clermont-Ferrand (in French) (2003).

[6] M. N. Queyranne, *On the Node Flow Cone of an Acyclic Directed Graph*, Seventh Aussois Conference of Combinatorial Optimization (2003).

Minimal Infeasible Subsystems and Benders Cuts

Matteo Fischetti

(joint work with Domenico Salvagnin and Arrigo Zanette)

1. Introduction

Suppose we are given a MIP problem $\min\{c^T x + d^T y : Ax \geq b, Tx + Qy \geq r, x \in \mathbb{Z}^n_+,\ y \in \mathbb{R}^t_+\}$, where matrix Q has m rows. Classical Benders decomposition states that solving such a problem is equivalent to solving:

$$
\begin{aligned}
&\min c^T x + \eta \\
&Ax \geq b \\
&\eta \geq u^T(r - Tx), \quad u \in \text{VERT} \qquad (1)\\
&v^T(r - Tx) \leq 0, \quad v \in \text{RAY} \\
&x \geq 0,\ x \text{ integer}
\end{aligned}
$$

where the additional variable η takes into account the objective function term $d^T y$, while sets VERT and RAY contain the vertices and extreme rays of the polyhedron $D = \{\ \pi \in \mathbb{R}^m_+ : \pi^T Q \leq d^T\}$, respectively.

Formulation (1) has exponentially many inequalities, so an iterative solution approach based on cutting planes is needed, that can be outlined as follows.

(1) Solve the *master* problem: $\min\{c^T x + \eta :\ A'x \geq b', \eta \in \mathbb{R}, x \in \mathbb{Z}^n_+\}$ where $A'x \geq b'$ includes $Ax \geq b$ plus (some of) the Benders cuts generated so far (none at the very beginning). Let (x^*, η^*) be an optimal solution of the master problem.

(2) Solve the *dual slave* problem: $\max\{\pi^T(r - Tx^*) : \pi^T Q \leq d^T, \pi \in \mathbb{R}^m_+\}$

(3) If the dual slave problem is unbounded, choose any unbounded extreme ray $\overline{v}$, add the *Benders feasibility cut* $\overline{v}^T(r - Tx) \leq 0$ to the master, and go to Step 1. Otherwise, let the optimal value and an optimal vertex be z^* and $\overline{u}$, respectively. If $z^* \leq \eta^*$ then stop. Otherwise, add the so-called *Benders optimality cut* $\eta \geq \overline{u}^T(r - Tx)$ to the master problem, and go to Step 1.

According to our computational experience, *optimality cuts* and *feasibility cuts* behave quite differently. Indeed, for many problems where term $d^T y$ gives a significant contribution to the overall optimal value, optimality cuts can be much more effective in moving the bound than feasibility cuts. Moreover, optimality cuts are typically worse from a numerical point view.

As a consequence, it is important to have some control on the kind (and quality) of Benders cuts generated at each iteration. Unfortunately, Benders decomposition—as it is typically implemented in the literature—is heavily biased toward feasibility cuts. As a matter of fact, as long as a violated feasibility cut exists, the dual slave is unbounded and hence no optimality cut is generated. As noted by Benders himself [3], however, if we solve the dual slave with the primal simplex method, then when we discover an unbounded ray we are "sitting on a vertex" of polyhedron D, and thus we can generate also an optimality cut with no additional computational effort. A main drawback of this approach is that optimality cut produced is not guaranteed to be violated, and in any case its discovery is quite "random" as the corresponding vertex is by no mean a one maximizing a certain quality index such as cut violation, depth, etc.

The lack of control on the *quality* of the Benders cuts is even more striking when feasibility cuts are generated, since the textbook method does not give any rule to choose among the unbounded rays.

The considerations above prompted us to introduce an effective criterion for choosing among violated (optimality or feasibility) Benders cuts, very much in the spirit of disjunctive cut generation that is also based on Cut Generation LP (CGLP); see [2].

2. Benders cuts and Minimal Infeasible Subsystems

The CGLP for Benders cuts can always be seen as a feasibility problem: given a master solution (x^*, η^*), it is possible to generate a violated cut if and only if the following primal slave problem is infeasible:

$$d^T y \leq \eta^*,\ Qy \geq r - Tx^*,\ y \geq 0 \tag{2}$$

or equivalently, by LP duality, if the dual slave problem $\max\{\pi^T(r - Tx^*) - \pi_0\eta^* : \pi^T Q \leq \pi_0 d^T, (\pi, \pi_0) \in \mathbb{R}^{m+1}_+\}$ is unbounded. If separation is successful, for an unbounded ray $(\overline{\pi}, \overline{\pi}_0)$ the generated cut reads $\overline{\pi}^T(r - Tx) - \overline{\pi}_0\eta \leq 0$.

In practice, one is interested in detecting a "minimal source of infeasibility" of (2), so as to detect a small set of rows that allow to cut the master solution. According to Gleeson and Ryan [4], the rows of any *Minimal* (with respect to set

inclusion) *Infeasible Subsystem* (MIS) of (2) are indexed by the support of the vertices of the following polyhedron, sometimes called the *alternative polyhedron*:

$$AP = \{(\pi, \pi_0) \in \mathbb{R}^{m+1}_+ : \ \pi^T Q \leq \pi_0 d^T, \ \pi^T(r - Tx^*) - \pi_0\eta^* = 1\} \tag{3}$$

where the unbounded objective function—i.e., the cut violation to be maximized—has been fixed to a normalization positive value. By choosing an appropriate objective function it is therefore possible to optimize over AP, thus selecting a violated cut corresponding to a MIS of (2) with certain useful properties. A natural objective function whose purpose is to try to minimize the cardinality of the support of the optimal vertex (and hence to find a small-cardinality MIS), is

$$\min \sum_{i=1}^{m} \pi_i + \pi_0 \tag{4}$$

As we are only interested in solutions with a positive cut violation, and since $\{(\pi, \pi_0) \in \mathbb{R}^{m+1}_+ : \ \pi^T Q \leq \pi_0 d^T\}$ is a cone, we can swap the role of the objective function (4) and of the normalization condition in (3), yielding the following equivalent CGLP akin to the one used for disjunctive cuts in [2]:

$$\max\{\pi^T(r - Tx^*) - \pi_0\eta^* : \ \pi^T Q \leq \pi_0 d^T, \ \sum_{i=1}^{m} \pi_i + \pi_0 = 1, \ (\pi, \pi_0) \in \mathbb{R}^{m+1}_+\} \tag{5}$$

It is worth noting that the feasible solution set of the above CGLP is never empty nor unbounded, so a violated cut can be generated if and only if the CGLP has a strictly positive optimal value. The latter formulation is preferable from a computational point because the normalization constraint $\sum_{i=1}^{m} \pi_i + \pi_0 = 1$, though very dense, is numerically more stable than its "cut violation" counterpart $\pi^T(r - Tx^*) - \pi_0\eta^* = 1$. Moreover, at each iteration only the CGLP objective function is affected by the change in the master solution (x^*, y^*), hence its re-optimization with the primal simplex method is usually quite fast.

3. Computational results

The effectiveness of our CGLP formulation has been tested on a collection of problems from the MIPLIB 2003 library.

We implemented the classical (textbook) Benders method, as well as two variants of our MIS-based CGLP. The textbook (*tb*) implementation is the original method as proposed by Benders [3]: if the dual slave problem is bounded, we generate one optimality cut, otherwise we generate both a feasibility and an optimality cut (the optimality cut being added to the master problem only if it is violated by the current master solution). *tb_noopt* is instead a typical Benders implementation where only one cut per iteration is generated—in case of unboundedness, only the feasibility cut associated with the unbounded dual-slave ray detected by the LP solver is added to the master. *mis* is our basic MIS-based method. It uses the CGLP (5) to solve the separation problem, hence it generates only one cut per iteration. *mis2* is a modified version of *mis*: after having solved the CGLP, if the generated cut is an optimality one, the generation of an additional feasibility cut is

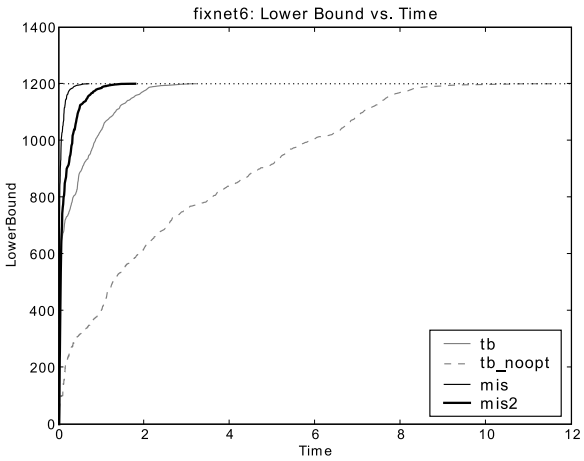


FIGURE 1. Lower bound growth vs. time (CPU seconds) with different separation methods on instance `fixnet6`. The dotted line is the known optimal value.

enforced by imposing condition $\pi_0 = 0$. The quality of the generated Benders cuts is measured in terms of "percentage gap closed" at the root node, as customary in cutting plane methods. A typical behavior is illustrated in Figure 1.

Overall, the *tb_noopt* performance is always very poor, whereas *tb* turns out to be the most efficient method only in 1 out of 11 instances, and only with little advantage over the competitors. *mis* and *mis2* are much more effective on 10 out of 11 instances, with speedups of 1 to 2 orders of magnitude. As expected, the average density of the cuts generated by *mis* and *mis2* is considerably smaller than *tb* and *tb_noopt*. This has a positive effect on the rate of growth of the master solution time as a function of the number of iterations.

A comparison between *mis* and *mis2* shows that *mis* qualifies as the method of choice, as it is usually faster than *mis2* due to the extra computing time spent by the latter in generating the additional feasibility cut. Nevertheless, there are instances where the extra separation effort is rewarded by a significant improvement of the overall performance; see [1] for more details.

REFERENCES

[1] M. Fischetti, D. Salvagnin, and A. Zanette, *Minimal Infeasible Subsystems and Benders cuts*, Technical Report, DEI, University of Padova (2008).

[2] E. Balas, S. Ceria, and G. Cornuéjols, *Mixed 0-1 programming by lift-and-project in a branch-and-cut framework*, Management Science **42** (1996), 1229–1246.

[3] J.F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik **4** (1962), 238–252.

[4] J. Gleeson and J. Ryan, *Identifying minimally infeasible subsystems of inequalities*, ORSA Journal on Computing **2(1)** (1990), 61–63.

Single-source 2-splittable Min-cost Flows

Martin Skutella
(joint work with Fernanda Salazar)

We study a relaxation of the following flow problem introduced by Kleinberg [5]:

Single-source unsplittable min-cost flow problem

Given: Digraph $G = (V, E)$ with capacities $u = (u_e)_{e \in E}$ and costs $c = (c_e)_{e \in E}$; source node $s \in V$ and p sink nodes $t_1, \ldots, t_p \in V$ with demands $d = (d_1, \ldots, d_p) \in \mathbb{R}^p_{\geq 0}$.

Task: Find a flow $(y_e)_{e \in E}$ with $y \leq u$ of minimum cost $c(y) = \sum_{e \in E} c_e y_e$ and with a path decomposition $(y_{P_i})_{i=1,\ldots,p}$ such that P_i is an s-t_i-path and $y_{P_i} = d_i$ for $i = 1, \ldots, p$.

Thus the demand d_i of each commodity i must be routed along a single s-t_i-path. Any such flow y is called an *unsplittable flow satisfying demands d*. Already the problem of deciding whether an unsplittable flow satisfying demands d and obeying capacity constraints $y \leq u$ exists is NP-complete [5]. It contains several well-known NP-complete problems as special cases, such as, for example, Partition, Bin Packing, or even scheduling parallel machines with makespan objective. On the other hand, if we drop the constraint on y to be unsplittable, what remains is a classical minimum cost flow problem that can be solved efficiently.

Let $d_{\max} := \max_i d_i$ denote the maximum demand value. A popular assumption in the context of unsplittable flow problems is the *no bottleneck condition* which says that no demand may exceed the capacity of any arc, that is,

$$d_{\max} \leq u_e \qquad \text{for all } e \in E. \tag{1}$$

Our work is motivated by the following conjecture.

Conjecture 1 (Goemans [4])**.** *For any flow x satisfying demands d, there is an unsplittable flow y satisfying demands d with*

$$y_e \leq x_e + d_{\max} \qquad \textit{for all } e \in E \tag{2}$$

and $c(y) \leq c(x)$.

Dinitz, Garg, and Goemans [2] prove that the conjecture without costs (i.e., removing the bound $c(y) \leq c(x)$) is true and provide an efficient algorithm that computes y.

The *congestion* of a given flow y is the minimum value $\alpha \geq 1$ with $y \leq \alpha u$. In particular, a flow of congestion 1 obeys the capacity constraints. The first approximation results for the min-congestion version of the single-source unsplittable flow problem (without costs) is given by Kleinberg [6]. Since a flow x satisfying

demands d with minimum congestion can be computed with classic network flow techniques, the result of Dinitz et al. [2] implies the existence of a 2-approximation algorithm for the problem without costs. For this and all further approximation results mentioned below we assume that the no-bottleneck condition (1) holds.

Kolliopoulos and Stein [8] prove the weaker version of Conjecture 1 where condition (2) is replaced by

$$y_e \;\leq\; 2x_e + d_{\max} \qquad \text{for all } e \in E \tag{3}$$

and with the relaxed cost bound $c(y) \leq 2c(x)$. Their result implies the existence of a bicriteria $(3, 2)$-approximation algorithm for congestion and cost. Improving upon this result, Skutella [10] gives a $(3, 1)$-approximation algorithm. He proves Conjecture 1 with (2) replaced by (3) but with the original cost bound $c(y) \leq c(x)$. Notice that an efficient algorithm that computes an unsplittable flow y as in Conjecture 1 would yield a $(2, 1)$-approximation algorithm. On the negative side, Erlebach and Hall [3] prove that, for arbitrary $\varepsilon > 0$, there is no $(2 - \varepsilon, 1)$-approximation algorithm, unless P=NP.

Kolliopoulos and Stein [8] and Skutella [10] both build upon the result that Conjecture 1 holds for the special case where all demand values are powers of 2. In [8] the case of general demands is handled by *rounding up* demand values to the nearest power of 2. This yields an increase in cost by a factor of at most 2. In contrast to this, the improved result in [10] is achieved by *rounding down* demand values to the nearest power of 2 and carefully adjusting the given flow x.

We mention the following reformulation of Conjecture 1 stated in [9].

Conjecture 2 ([9])**.** *Any flow x satisfying demands d can be written as a convex combination of unsplittable flows y^ℓ, $\ell \in L$, with property* (2) *and satisfying demands d.*

It is not difficult to observe that Conjecture 2 is equivalent to Conjecture 1. Building upon [8] and [10], Martens, Salazar, and Skutella [9] prove the following result.

Theorem 3 ([9])**.** *Conjecture 2 holds if all demands are powers of* 2*. Moreover, the family of unsplittable flows y^ℓ, $\ell \in L$, can be obtained in polynomial time.*

In particular, the cardinality of L is polynomially bounded in the input size. More precisely, it is at most $|E| + 1$ (by Carathéodory's Theorem).

Baier, Köhler, and Skutella [1] introduce the following relaxation of unsplittable flows. For a given $k \geq 1$, a *k-splittable flow* must route each commodity along at most k paths. In particular, 1-splittable flows are unsplittable flows. We consider the case $k = 2$. The resulting relaxation of our unsplittable flow problem is the single-source 2-splittable min-cost flow problem.

Kolliopoulos [7] presents an efficient algorithm that, given a flow x satisfying demands d, finds a 2-splittable flow y satisfying demands d with $y_e \leq \frac{4}{3}x_e + \frac{2}{3}d_{\max}$ for all $e \in E$ and $c(y) \leq c(x)$. This yields a $(2, 1)$-approximation algorithm for the single-source 2-splittable min-cost flow problem. The main idea behind this result

is to *round down* the demand values to the nearest sum of two powers of 2 and to carefully adjust the given flow x (as in [10]).

Inspired by the work of Kolliopoulos [7], we present the following improved result yielding a $(\frac{11}{6}, 1)$-approximation algorithm for the single-source 2-splittable min-cost flow problem.

Theorem 4. *For any flow x satisfying demands d, there is a 2-splittable flow y satisfying demands d with*

$$y_e \leq \frac{4}{3} x_e + \frac{d_{\max}}{2} \qquad \text{for all } e \in E \tag{4}$$

and $c(y) \leq c(x)$. Moreover, such a flow y can be found in polynomial time.

In order to achieve this result, we build upon results mentioned above and introduce several new ideas and techniques. The history of bicriteria approximations for unsplittable flows outlined above suggests that rounding down demands to powers of 2 (as in [10]) leads to superior results compared to rounding up (as in [8]). Consequently, Kolliopoulos [7] also uses rounding-down in his algorithm. Surprisingly, the algorithm behind Theorem 4 is based on rounding-up. We also obtain a new algorithm and proof for the best known result for the unsplittable min-cost flow problem given in [10]. This implies, in particular, that the original idea of Kolliopoulos and Stein to round up demands to powers of 2 can still lead to unsplittable flows y that are no more expensive than the given flow x. This insight also sheds new light on Conjecture 1.

Moreover, in contrast to earlier approximation results, in our result we use a more sophisticated technique based upon Theorem 3. That is, for the problem with rounded demands we compute an entire family of 2-splittable flows which contain an accordingly rounded version of the given flow x in their convex hull. We emphasize that, in our approach, it is not sufficient to only compute a member of this family that has minimum cost. Only after going back to the original demands and rounding down all 2-splittable flows we can check which members of the family do not violate the cost bound.

References

[1] G. Baier, E. Köhler, and M. Skutella, *On the k-splittable flow problem*, Algorithmica **42** (2005), 231–248.

[2] Y. Dinitz, N. Garg, and M. X. Goemans, *On the single source unsplittable flow problem*, Combinatorica **19** (1999), 17–41.

[3] T. Erlebach and A. Hall, *NP-hardness of broadcast scheduling and inapproximability of single-source unsplittable min-cost flow*, Journal of Scheduling **7** (2004), 223–241.

[4] M. X. Goemans, January 2000, Personal communication (also cited in [10]).

[5] J. M. Kleinberg, *Approximation algorithms for disjoint paths problems*, Ph.D. thesis, M.I.T., 1996.

[6] J. M. Kleinberg, *Single-source unsplittable flow*, Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (Burlington, VT) (1996), 68–77.

[7] S. G. Kolliopoulos, *Minimum-cost single-source 2-splittable flow*, Information Processing Letters **94** (2005), 15–18.

[8] S. G. Kolliopoulos and C. Stein, *Approximation algorithms for single-source unsplittable flow*, SIAM Journal on Computing **31** (2002), 919–946.
[9] M. Martens, M. F. Salazar, and M. Skutella, *Representing single source multicommodity flows as convex combinations of unsplittable flows*, Algorithms — ESA '07 (L. Arge, M. Hoffmann, and E. Welzl, eds.), LNCS **4698**, Springer (2007), 395–406.
[10] M. Skutella, *Approximating the single source unsplittable min-cost flow problem*, Mathematical Programming **91** (2002), 493–514.

The Alcuin Number of a Graph

Gerhard J. Woeginger
(joint work with Péter Csorba and Cor A.J. Hurkens)

Alcuin's river crossing problem. The Anglo-Saxon monk Alcuin (735–804 A.D.) was one of the leading scholars of his time. He served as head of Charlemagne's Palace School at Aachen, he developed the Carolingian minuscule (a script which has become the basis of the way the letters of the present Roman alphabet are written), and he wrote a number of elementary texts on arithmetic, geometry, and astronomy. His book *"Propositiones ad acuendos iuvenes"* (Problems to sharpen the young) is perhaps the oldest collection of mathematical problems written in Latin. It contains the following well-known problem.

> A man had to transport to the far side of a river a wolf, a goat, and a bundle of cabbages. The only boat he could find was one which would carry only two of them. He sought a plan which would enable them all to get to the far side unhurt. Let him, who is able, say how it could be possible to transport them safely?

In a safe transportation plan, neither wolf and goat nor goat and cabbage can be left alone together. Alcuin's river crossing problem differs significantly from other mediaeval puzzles, since it is neither geometrical nor arithmetical but purely combinatorial. In fact it might be one of the oldest combinatorial problems in the history of mathematics.

Graph-theoretic model. We consider the following generalization of Alcuin's problem to arbitrary graphs $G = (V, E)$: The man has to transport a set V of items/vertices across the river. Two items are connected by an edge in E, if they are *conflicting* and thus cannot be left alone together without human supervision. The available boat has capacity $b \geq 1$, and thus can carry the man together with any subset of at most b items. A *feasible schedule* is a finite sequence of triples (L_1, B_1, R_1), (L_2, B_2, R_2), $\ldots$, (L_s, B_s, R_s) of subsets of the item set V that satisfies the following conditions (FS1)–(FS3). The odd integer s is called the *length* of the schedule.

(FS1) For every k, sets L_k, B_k, R_k form a partition of V. The sets L_k and R_k are stable sets, and B_k contains at most b elements.
(FS2) The sequence starts with $L_1 \cup B_1 = V$ and $R_1 = \emptyset$, and the sequence ends with $L_s = \emptyset$ and $B_s \cup R_s = V$.

(FS3) For even $k \geq 2$: $B_k \cup R_k = B_{k-1} \cup R_{k-1}$ and $L_k = L_{k-1}$.
For odd $k \geq 3$: $L_k \cup B_k = L_{k-1} \cup B_{k-1}$ and $R_k = R_{k-1}$.

Here the kth triple encodes the kth boat trip: L_k contains the items on the left bank, B_k the items in the boat, and R_k the items on the right bank. Odd indices correspond to forward boat trips from left to right, and even indices correspond to backward trips from right to left.

We are interested in the smallest possible capacity of a boat for which a graph $G = (V, E)$ possesses a feasible schedule; this capacity is called the *Alcuin number* $\textsc{Alcuin}(G)$ of the graph. A natural problem variant puts a hard constraint on the length of the schedule: Let $t \geq 1$ be an odd integer. The smallest possible capacity of a boat for which G possesses a feasible schedule with at most t boat trips is called the *t-trip constrained* Alcuin number $\textsc{Alcuin}_t(G)$. In our graph-theoretic model Alcuin's river crossing problem corresponds to the path P_3 with three vertices w(olf), g(oat), c(abbage) and two edges $[w, g]$ and $[g, c]$. It can be seen that $\textsc{Alcuin}(P_3) = 1$, that $\textsc{Alcuin}_1(P_3) = 3$, that $\textsc{Alcuin}_t(P_3) = 2$ for $t \in \{3, 5\}$, and that $\textsc{Alcuin}_t(P_3) = 1$ for all $t \geq 7$.

Known results. The idea of generalizing Alcuin's problem to arbitrary conflict graphs goes back (at least) to Prisner [3] and Bahls [1]: Prisner introduced it in 2002 in his course on Discrete Mathematics at the University of Maryland, and Bahls discussed it in 2005 in a talk in the Mathematics Seminar at the University of North Carolina.

Bahls [1] (and later Lampis & Mitsou [2]) observed that it is NP-hard to compute the Alcuin number exactly; Lampis & Mitsou [2] also showed that the Alcuin number is hard to approximate. These negative results follow quite easily from the close relationship between the Alcuin number and the vertex cover number. The papers [1, 2] provide a complete analysis of the Alcuin number of trees. Finally, Lampis & Mitsou [2] proved that the computation of the trip constrained Alcuin number $\textsc{Alcuin}_3(G)$ is NP-hard.

Our results. Our main result is the following structural characterization of the Alcuin number, which also yields an NP-certificate: A graph $G = (V, E)$ possesses a feasible schedule for a boat of capacity $b \geq 1$, if and only if there exist five subsets X_1, X_2, X_3, Y_1, Y_2 of V that satisfy the following four conditions.

(i) The three sets X_1, X_2, X_3 are pairwise disjoint. Their union $X := X_1 \cup X_2 \cup X_3$ forms a stable set in G.
(ii) The (not necessarily disjoint) sets Y_1, Y_2 are non-empty subsets of the set $Y := V - X$, which satisfies $|Y| \leq b$.
(iii) $X_1 \cup Y_1$ and $X_2 \cup Y_2$ are stable sets in G.
(iv) $|Y_1| + |Y_2| \geq |X_3|$.

Furthermore, in case these four conditions are satisfied, then there exists a feasible schedule of length at most $2|V| + 1$. This bound $2|V| + 1$ is the best possible (for any $|V| \geq 3$).

For any graph G, the Alcuin number either equals the vertex cover number (in which case the graph is a *small-boat* graph) or equals the vertex cover number

plus one (in which case the graph is a *big-boat* graph). We derive a number of combinatorial lemmas around the division line between these two classes, which all fall out quite easily from our structural characterization.

From the complexity point of view, computing the Alcuin number is an NP-hard problem. On the positive side, this problem belongs to the class FPT of fixed-parameter tractable problems. Furthermore, distinguishing small-boat graphs from big-boat graphs is NP-hard, and also computing the t-trip constrained Alcuin number $\textsc{Alcuin}_t(G)$ is NP-hard for every fixed value $t \geq 3$.

Finally, we derive a number of results for graphs from specially structured graph classes.

- If a chordal graph is not a split graph, then it is a small-boat graph.
- A split graph is a small-boat graph if and only if one of the following holds: The split graph can be partitioned into a maximum stable set and a clique, such that there exist two (not necessarily distinct) vertices in the clique that have at most two common neighbors in the stable set.
- If a tree is not a star with at least three leaves, then it is a small-boat graph.
- The Alcuin number of a bipartite graph can be computed in polynomial time.
- Every planar graph with vertex cover number at least five is a small-boat graph.
- It is NP-hard to compute the Alcuin number of a planar graph.
- It can be decided in polynomial time, whether a planar graph is a small-boat graph. (Hence: It is hard to compute the Alcuin number of a planar graph, and it is hard to compute its vertex cover number. But it is easy to decide whether these two numbers coincide.)

Open problems. The central open problem concerns perfect graphs: Is there a polynomial time algorithm for computing the Alcuin number of a perfect graph?

Also the computational complexity of recognizing small-boat graphs remains unclear. Is the problem of recognizing small-boat graphs contained in NP? We have proved that this problem is NP-hard, but there is no reason to assume that it lies in NP: To demonstrate that a graph is small-boat in a straightforward way, we have to show that its Alcuin number is small (NP-certificate) and that its vertex cover number is large (coNP-certificate). This mixture of NP- and coNP-certificates suggests that the problem might be located in one of the complexity classes above NP; the complexity class DP might be a reasonable guess.

References

[1] P. Bahls, *The wolf, the goat, and the cabbage: A modern twist on a classical problem*, Unpublished manuscript, University of North Carolina at Asheville (2005).

[2] M. Lampis and V. Mitsou, *The ferry cover problem*, Proceedings of FUN'2007, LNCS **4475**, Springer (2007), 227–239.

[3] E. Prisner, *Generalizing the wolf-goat-cabbage problem*, Electronic Notes in Discrete Mathematics **27** (2006), 83.

Whitney Type Theorems for Signed Graphs and Grafts

Bertrand Guenin
(joint work with Irene Pivotto and Paul Wollan)

A *cycle* of G is a subgraph where all vertices have even degree. An inclusion-wise minimal cycle is a *circuit*. The set of all cycles (resp. all cuts) of a graph G forms a vector space over the two element field (where addition is the symmetric difference). We denote by $\text{cycle}(G)$ and $\text{cut}(G)$ these vector spaces. Note that $\text{cycle}(G)$ and $\text{cut}(G)$ are orthogonal spaces. In particular, $\text{cycle}(G) = \text{cycle}(G')$ if and only if $\text{cut}(G) = \text{cut}(G')$.

A *Whitney-flip* consists of decomposing a graph G along a two vertex cutset s and t into parts G_1 and G_2 and then recombining the two parts by identifying the vertex s (resp. t) of G_1 with vertex t (resp. s) of G_2. Whitney [1] proved the following seminal result,

Theorem 1. *Let G and G' be graphs. Then* $\text{cycle}(G) = \text{cycle}(G')$ *(resp.* $\text{cut}(G) = \text{cut}(G')$*) if and only if G' is obtained from G by a sequence of Whitney-flips.*

Our main result is a generalization of the previous theorem to respectively signed graphs and grafts. A *signed graph* is a pair (G, Σ) where G is a graph and $\Sigma \subseteq EG$. We say that $B \subseteq EG$ is even (resp. odd) if $|B \cap \Sigma|$ is even (resp. odd). The set of all even cycles of (G, Σ) forms a vector space which we denote $\text{ecycle}(G, \Sigma)$. Note that an (inclusion-wise) minimal even cycle of (G, Σ) is either an even circuit of (G, Σ) or the disjoint union of two odd circuits of (G, Σ) that share at most one vertex.

Even Cycle Isomorphism Problem: Let (G, Σ) and (G', Σ') be signed graphs. When is the relation $\text{ecycle}(G, \Sigma) = \text{ecycle}(G', \Sigma')$ satisfied?

Note that Theorem 1 answers the question for the case where $\Sigma = \Sigma' = \emptyset$. Gerards [2] gives a survey about what was known about this problem.

A *graft* is a pair (G, T) where G is a graph and $T \subseteq VG$ where $|T|$ is even. A cut $\delta_G(U)$ of G is *even* if $|U \cap T|$ is even. The set of all even cuts of (G, T) forms a vector space which we denote $\text{ecut}(G, T)$.

Even Cut Isomorphism Problem: Let (G, T) and (G', T') be grafts. When is the relation $\text{ecut}(G, T) = \text{ecut}(G', T')$ satisfied?

Note that Theorem 1 answers the question for the case where $T = T' = \emptyset$.

We first show that the two aforementioned isomorphism problems are in fact equivalent,

Theorem 2. *Let G and G' be graphs with same edge sets. Then there exists a pair $\Sigma, \Sigma' \subseteq EG$ such that* $\text{ecycle}(G, \Sigma) = \text{ecycle}(G', \Sigma')$ *if and only if there exists a pair $T \subseteq VG, T' \subseteq VG'$ (where $|T|, |T'|$ even) such that* $\text{ecut}(G, T) = \text{ecut}(G', T')$.

Thus to solve the Even Cut Isomorphism Problem it suffices to solve the Even Cycle Isomorphism Theorem.

We will now briefly outline our solution for that problem. If $\text{ecycle}(G,\Sigma) = \text{ecycle}(G,\Gamma)$ then we call the operation that consists of replacing Σ by Γ *resigning*. It is easy to verify that we must have $\Gamma = \Sigma \triangle B$ where $B \in \text{cut}(G)$. Consider a signed graph (G,Σ). Suppose that there are vertices v_1, v_2 of G such that $\Sigma \subseteq \delta_G(v_1) \cup \delta_G(v_2)$. We can construct a new signed graph (G',Σ) as follows:

(i) turn odd edges with ends v_1, v_2 into loops;
(ii) turn odd loops into edges with ends v_1, v_2;
(iii) replace end v_1 of odd edges by end v_2;
(iv) replace end v_2 of odd edges by end v_1.

Note we apply (iii) and (iv) to edges which are not in (i) or (ii). We say that (G',Σ) is obtained from (G,Σ) by a *Lovász-flip*. It can be readily checked that $\text{ecycle}(G,\Sigma) = \text{ecycle}(G',\Sigma')$.

Given a pair of signed graphs (G,Σ) and (G',Σ'), we show that after applying a sequence of Whitney-flips, Lovász-flips, and resigning to (G,Σ) and (G',Σ') one of the following cases must occur:

Case 1: We can decompose (G,Σ) and (G',Σ') along small separations to create (G_1,Σ_1), (G_2,Σ_2) and (G'_1,Σ'_1), (G'_2,Σ'_2) such that

$$\text{ecycle}(G,\Sigma) = \text{ecycle}(G',\Sigma')$$

if and only if

$$\text{ecycle}(G_1,\Sigma_1) = \text{ecycle}(G'_1,\Sigma'_1) \quad \text{and}$$
$$\text{ecycle}(G_2,\Sigma_2) = \text{ecycle}(G'_2,\Sigma'_2)$$

Moreover, $|VG_1|$, $|VG_2| < |VG|$ and $|VG'_1|$, $|VG'_2| < |VG'|$.

Case 2: The pairs (G,Σ) and (G',Σ') are either *graphic* or *co-graphic*.

Case 3: (G',Σ') obtained from (G,Σ) by applying once, one of three possible operations: a *Shuffle*, a *Tilt*, or a *Twist*.

Let us now discuss Case 1-Case 3 in more details.

Case 1: We say that a signed graph (G,Σ) is *graphic* if $\text{ecycle}(G,\Sigma) = \text{cycle}(\tilde{G})$ for some graph $\tilde{G}$. The following theorem of Gerards, Lovász, Schrijver, Seymour, Shih, and Truemper solves the Even Cycle isomorphism problem for the graphic case [2]:

Theorem 3. *Suppose* $(G,\Sigma),(G',\Sigma')$ *are graphic and* $\text{ecycle}(G,\Sigma) = \text{ecycle}(G',\Sigma')$. *Then* (G',Σ') *can be obtained from* (G,Σ) *by a sequence of resigning, Lovász-flips, and Whitney-flips.*

We say that a graft (G,T) is *co-graphic* if $\text{ecut}(G,T) = \text{cut}(\tilde{G})$ for some graph $\tilde{G}$. For instance (G,T) is co-graphic when $|T| \leq 2$. Suppose we have a pair of co-graphic grafts $(G,T),(G',T')$ such that $\text{ecut}(G,T) = \text{ecut}(G',T')$. Then theorem 2 implies that there exists Σ,Σ' such that $\text{ecycle}(G,\Sigma) = \text{ecycle}(G',\Sigma')$

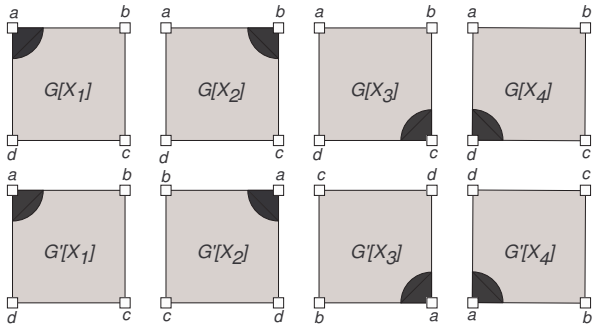
moreover Σ, Σ' are essentially unique. We then say that $(G, \Sigma), (G', \Sigma')$ are co-graphic as well. The co-graphic case can be related to the graphic case and then solved using Theorem 3.

Case 2: We decompose G and G' along special two and three separations. Note that since each of the parts of the decomposition is smaller than the original graphs, this occurs only a finite number of time.

Case 3: We describe the shuffle operation here and omit discussing the Twist and Tilt operations. Consider (G, Σ, T) with four special vertices a, b, c, d. Suppose that EG can be partitioned into $X_1, \ldots, X_4$ (not necessarily all non-empty) such that for all $i, j \in [4]$, $i \neq j$, $VG[X_i] \cap VG[X_j] \subseteq \{a, b, c, d\}$ (where $VG[X_i]$ is the set of vertices of the subgraph of G induced by X_i). For all $i \in [4]$ denote by a_i (resp. b_i, c_i, d_i) the copy of vertex a (resp. b, c, d) of $G[X_i]$. Construct G' by:

(i) identifying vertices a_1, b_2, c_3, d_4 to vertex say a';
(ii) identifying vertices b_1, a_2, d_3, c_4 to vertex say b';
(iii) identifying vertices d_1, c_2, b_3, a_4 to vertex say c';
(iv) identifying vertices c_1, d_2, a_3, b_4 to vertex say d'.

Suppose $\Sigma = \delta_{G'}(a')$ and let $\Sigma' := \delta_G(a)$. Then we say that (G, Σ) and (G', Σ') are *related by a shuffle*. In the figure, the top four squares represent the parts



$G[X_1], G[X_2], G[X_3], G[X_4]$ of G. The graph G is obtained from by superimposing each of these parts and identifying the vertices at each of the corners. The bottom four squares represent the parts $G[X_1], G[X_2], G[X_3], G[X_4]$ of G but some of the squares have been flipped from the top squares so that their corners do not correspond the original corners. The graph G' is obtained by superimposing each of these parts and identifying the vertices at each of the corners. The shaded areas near some of the corner on the top (resp. bottom) row of squares represent the edges in Σ (resp. Σ').

References

[1] H. Whitney, *2-Isomorphic graphs*, Am. J. Math. **55** (1933), 245–254.
[2] B. Gerards, *A few comments on isomorphism of even cycle spaces*, `http://homepages.cwi.nl/~bgerards/`

Discrete Convex Analysis—Basics and Topics

Kazuo Murota

Discrete convex analysis is a theory of L-convex and M-convex functions, aiming at a discrete analogue of convex analysis for nonlinear discrete optimization (Murota [6, 7, 9], Fujishige [1, Chap. VII]).

$L^\natural$-convex functions, an equivalent variant of L-convex functions, are defined in terms of a discrete version of the mid-point convexity, as follows. We first observe that a convex function g on $\mathbb{R}^n$ satisfies

$$g(p) + g(q) \geq g\left(\frac{p+q}{2}\right) + g\left(\frac{p+q}{2}\right) \qquad (p, q \in \mathbb{R}^n). \tag{1}$$

This property, called *midpoint convexity*, is known to be equivalent to convexity if g is a continuous function. For a function $g : \mathbb{Z}^n \to \overline{\mathbb{R}}$ in discrete variables, where $\overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$, the above inequality does not always make sense, since the midpoint $\frac{p+q}{2}$ of two integer vectors p and q may not be integral. Instead we simulate (1) by

$$g(p) + g(q) \geq g\left(\left\lceil\frac{p+q}{2}\right\rceil\right) + g\left(\left\lfloor\frac{p+q}{2}\right\rfloor\right) \qquad (p, q \in \mathbb{Z}^n), \tag{2}$$

where, for $z \in \mathbb{R}$ in general, $\lceil z \rceil$ denotes the smallest integer not smaller than z (rounding-up to the nearest integer) and $\lfloor z \rfloor$ the largest integer not larger than z (rounding-down to the nearest integer), and this operation is extended to a vector by componentwise applications. We refer to (2) as *discrete midpoint convexity*, and say that a function $g : \mathbb{Z}^n \to \overline{\mathbb{R}}$ is *$L^\natural$-convex* if it satisfies this.

$M^\natural$-convex functions, an equivalent variant of M-convex functions, are defined in terms of a discrete version of the equi-distance convexity, as follows. We first observe that a convex function f on $\mathbb{R}^n$ satisfies the inequality

$$f(x) + f(y) \geq f(x - \alpha(x-y)) + f(y + \alpha(x-y)) \tag{3}$$

for every $\alpha \in \mathbb{R}$ with $0 \leq \alpha \leq 1$. The inequality (3) says that the sum of the function values evaluated at two points, x and y, does not increase if the two points approach each other by the same distance on the line segment connecting them. We refer to this property as *equidistance convexity*. For a function $f : \mathbb{Z}^n \to \overline{\mathbb{R}}$ in discrete variables we simulate equidistance convexity (3) by moving a pair of points (x, y) to another pair (x', y') along the coordinate axes rather than on the connecting line segment. To be more specific, we consider two kinds of possibilities

$$(x', y') = (x - \chi_i, y + \chi_i) \quad \text{or} \quad (x', y') = (x - \chi_i + \chi_j, y + \chi_i - \chi_j) \tag{4}$$

with indices i and j such that $x_i > y_i$ and $x_j < y_j$. For a vector $z \in \mathbb{R}^n$ in general, define the *positive* and *negative supports* of z as

$$\mathrm{supp}^+(z) = \{i \mid z_i > 0\}, \qquad \mathrm{supp}^-(z) = \{j \mid z_j < 0\}.$$

Then the expression (4) can be rewritten compactly as $(x', y') = (x - \chi_i + \chi_j, y + \chi_i - \chi_j)$ with $i \in \mathrm{supp}^+(x-y)$ and $j \in \mathrm{supp}^-(x-y) \cup \{0\}$, where χ_0 is defined to be the zero vector. As a discrete analogue of equidistance convexity (3) we

consider the following condition: For any $x, y \in \mathrm{dom} f$ and any $i \in \mathrm{supp}^+(x-y)$, there exists $j \in \mathrm{supp}^-(x-y) \cup \{0\}$ such that

$$f(x) + f(y) \geq f(x - \chi_i + \chi_j) + f(y + \chi_i - \chi_j), \tag{5}$$

which is referred to as the *exchange property*. A function $f : \mathbb{Z}^n \to \overline{\mathbb{R}}$ having this exchange property is called *$M^\natural$-convex*.

For a function $f : \mathbb{Z}^n \to \overline{\mathbb{Z}}$ with $\mathrm{dom} f \neq \emptyset$, a discrete version of the Legendre transformation is defined by

$$f^\bullet(p) = \sup\{\langle p, x\rangle - f(x) \mid x \in \mathbb{Z}^n\} \qquad (p \in \mathbb{Z}^n), \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product. Under the discrete Legendre transformation (6), the class of integer-valued $L^\natural$-convex functions and the one of $M^\natural$-convex functions are in one-to-one correspondence. For an $M^\natural$-convex function f, $f^\bullet$ is an $L^\natural$-convex function and $(f^\bullet)^\bullet = f$; and similarly for an $L^\natural$-convex function f (Conjugacy Theorem).

A *discrete separation theorem* means a statement like:

> For any $f : \mathbb{Z}^n \to \overline{\mathbb{R}}$ and $h : \mathbb{Z}^n \to \underline{\mathbb{R}}$ belonging to certain classes of functions, if $f(x) \geq h(x)$ for all $x \in \mathbb{Z}^n$, then there exist $\alpha^* \in \mathbb{R}$ and $p^* \in \mathbb{R}^n$ such that
>
> $$f(x) \geq \alpha^* + \langle p^*, x\rangle \geq h(x) \qquad (\forall x \in \mathbb{Z}^n).$$
>
> Moreover, if f and h are integer-valued, there exist integer-valued $\alpha^* \in \mathbb{Z}$ and $p^* \in \mathbb{Z}^n$.

Discrete separation theorems often capture deep combinatorial properties in spite of the apparent similarity to the separation theorem in convex analysis. A discrete separation theorem holds for $L^\natural$-convex functions, which generalizes Frank's discrete separation theorem for submodular set functions. Another discrete separation theorem holds for $M^\natural$-convex functions, which generalizes Frank's weight splitting theorem for weighted matroid intersection.

Finite metric space is closely related to discrete convexity in several aspects. Distance functions satisfying triangle inequality are in one-to-one correspondence with positively homogeneous M-convex functions, and tree metrics are the same as valuated matroids of rank two. Furthermore, the Bandelt–Dress split decomposition and the Buneman construction can be derived as decompositions of polyhedral convex functions (Hirai [2], Koichi [5]).

The concept of M-convex functions can be extended to functions on constant-parity jump systems (Murota [8]). For $x, y \in \mathbb{Z}^n$ we call $s \in \mathbb{Z}^n$ an (x, y)-increment if $s = \chi_i$ for some $i \in \mathrm{supp}^+(y-x)$ or $s = -\chi_i$ for some $i \in \mathrm{supp}^-(y-x)$. We call $f : \mathbb{Z}^n \to \overline{\mathbb{R}}$ an M-convex function (on a constant-parity jump system) if it satisfies the following exchange property: For any $x, y \in \mathrm{dom} f$ and any (x, y)-increment s, there exists an $(x + s, y)$-increment t such that

$$f(x) + f(y) \geq f(x + s + t) + f(y - s - t).$$

It then follows that $\mathrm{dom} f$ is a constant-parity jump system. Operations such as infimal convolution can be generalized (Kobayashi, Murota and Tanaka [3]).

Odd-cycle symmetry condition in the even factor problem can be formulated as M-convexity on jump systems (Kobayashi and Takazawa [4]).

References

[1] S. Fujishige, *Submodular Functions and Optimization*, 2nd ed., Annals of Discrete Mathematics **58**, Elsevier (2005).

[2] H. Hirai, *A geometric study of the split decomposition*, Discrete and Computational Geometry **36** (2006), 331–361.

[3] Y. Kobayashi, K. Murota, and K. Tanaka, *Operations on M-convex functions on jump systems*, SIAM Journal on Discrete Mathematics **21** (2007), 107–129.

[4] Y. Kobayashi and K. Takazawa, *Even factors, jump systems, and discrete convexity*, Journal of Combinatorial Theory, Series B, to appear.

[5] S. Koichi, *The Buneman index via polyhedral split decomposition*, METR 2006-57, Department of Mathematical Informatics, University of Tokyo (2006).

[6] K. Murota, *Discrete convex analysis*, Mathematical Programming **83** (1998), 313–371.

[7] K. Murota, *Discrete Convex Analysis*, SIAM Monographs on Discrete Mathematics and Applications **10**, Society for Industrial and Applied Mathematics, Philadelphia (2003).

[8] K. Murota, *M-convex functions on jump systems: A general framework for minsquare graph factor problem*, SIAM Journal on Discrete Mathematics **20** (2006), 213–226.

[9] K. Murota, *Recent developments in discrete convex analysis*, in: W. Cook, L. Lovasz and J. Vygen, eds., *Research Trends in Combinatorial Optimization*, Chapter 11, Springer (2008), 219–260.

Weighted Fractional Matroid Matching

Gyula Pap

(joint work with Dion Gijswijt)

In the matroid matching problem, we are given a matroid $M = (S, r)$ with rank function r over a groundset S, which is partitioned into subsets of size 2, called lines, denoted by E. Our goal is to find a maximum number of lines so that their union is an independent set. We assume matroids come with an independence testing oracle. An equivalent version of matroid matching was originally introduced by Lawler, motivated by the applications such as graph matching and matroid intersection. The bad news is that matroid matching is exponential under the oracle framework, and also shown to be NP-hard, see Jensen, Korte [6] and Lovász [7]. On the other hand, Lovász [7] proved that it is tractable for a linearly represented matroid, later generalized to some broader classes of matroids. Thus it became a powerful tool with a growing number of applications, such as disjoint A-paths [8], maximum genus graph embedding [4], parity constrained orientations [3], and rigid planar pinning down.

There are possibilities to further generalize the concept of matroid matching, and one of them is polymatroid matching, where a partition-type condition is proved necessary and sufficient for the class polymatroids without so-called double cycles [9]. Another possibility is the weighted version, when we are given non-negative weights for the lines, and our goal is to find a maximum weight matroid matching. This problem is open even for linearly represented matroids. Note that the fractional relaxation of matching problems usually are easier to solve, thus,

before trying to solve the weighted matroid matching problem, we should try its fractional relaxation, called weighted fractional matroid matching. But what is the proper definition of a fractional matroid matching?

It was Vande Vate [12] who originally came up with the non-trivial definition of fractional matroid matchings. We are looking for an LP that is a fractional relaxation defined by "nice" inequalities, i.e. its integral solutions are exactly the 0-1 incidence vectors of matchings, and we also liked to have nice applications. Before Vande Vate's good definition, here's one that is not so good for our purposes. Consider a vector $x : E \to \mathbb{R}$, and impose the constraints $x \geq 0$, and $x(X) \leq r(\cup X)/2$ for all $X \subseteq E$. This certainly is a fractional relaxation of matroid matching, but unluckily, it is not interesting, since all one can say follows from properties of a 2-polymatroid. There is, however, another clever definition, one that is similar to the definition of Vande Vate, that fits better in the picture with matroid matching.

Consider a finite or infinite matroid $M = (S, r)$, and let E be a set of subsets of S of rank 1 or 2 called "lines". Note that this definition of a line is equivalent to, but technically more general than the one above. For $K \subseteq S$ we define $d_K : E \to \{0, 1, 2\}$ for $e \in E$ by

$$d_K(e) := \begin{cases} 0 & \text{if } e \cap K = \emptyset \\ 2 & \text{if } e \subseteq K \\ 1 & \text{otherwise} \end{cases}$$

Then the solutions $x : E \to \mathbb{R}$ to the following system are called fractional matroid matchings:

$$x \geq 0 \tag{1}$$

$$\sum_{e \in E} d_K(e) \cdot x(e) \leq r(K) \qquad \text{for all } K \subseteq S \tag{2}$$

One should think of the constraint (2) as the analogue of the degree bound defining the fractional matching polytope of a graph. Actually, if M is the free matroid, then (2) is equivalent with the degree bound, thus we end up with the fractional matching polytope of the graph (S, E). A crucial difference is that here, in general, we may have an exponential number of these degree constraints (2). Note that (1)-(2) is defined by linear inequalities with coefficients 0,1,2, thus the set of feasible solutions – i.e. the set of fractional matroid matchings – is a polytope. Remark that, even if M is an infinite matroid, we obtain a polytope. However simple the system may look like, it is not straightforeward to separate or optimize over the polytope, actually, it is not even simple for some given vector x to test feasibility.

The maximum weight fractional matroid matching problem is that, given weights $w \in \mathbb{R}^E_+$, maximize $w \cdot x$ subject to (1)-(2). The maximum fractional matriod matching problem is to maximize $\mathbf{1} \cdot x$ subject to (1)-(2), thus it is the special case with weights all one. Vande Vate proved the following min-max result for maximum fractional matroid matching.

Theorem 1 (Vande Vate, 1992, [12]). $\max\{\mathbf{1} \cdot x : x : E \to \mathbb{R}$ *satisfying* (1)-(2)$\} = \min\{\frac{1}{2}r(K) + \frac{1}{2}r(L) : K, L \subseteq S$ *such that* $\frac{1}{2}d_K + \frac{1}{2}d_L \geq \mathbf{1}\}$. *Moreover, the maximum attains with a fractional matroid matching that is half-integral.*

This result is essentially equivalent to the existence of a half-integral primal and dual optima, for all-one weights. Recently, Dion Gijswijt observed that (1)-(2) is actually a half-TDI system, meaning that for any weights, the dual system has a half-integral optimum. His simple proof is based on uncrossing, and fractional matching in an auxiliary graph.

Theorem 2 (Gijswijt, 2006). (1)-(2) *is half-TDI.*

The half-TDI-ness of (1)-(2) also implies that the polytope is half-integral, that is, all its vertices are half-integral, and thus, the maximum attains with a half-integral x for any weights w. This, for $w = \mathbf{1}$, implies Vande Vate's result. Let us mention that the independent set polytope, the matroid intersection polytope, and the graph fractional matching polytope arise as special cases of this one. As a further appliaction one can prove that the problem of fractionally packing A-paths is also equivalent with an instance of maximum fractional matroid matching, see [10].

Knowing the min-max result, Chang et al. [1, 2] worked on the complexity of finding a maximum fractional matroid matching matching, and constructed a polynomial time algorithm. That algorithm is a bit technical, and heavily relies on the all-one weights.

Theorem 3 (Chang, Llewellyn, Vande Vate, 2001, [1, 2]). *There is a polynomial time algorithm for maximum fractional matroid matching.*

Although linear programming duality and half-integrality works out just the same for arbitrary weights, the complexity of solving weighted fractional matroid matching remained an open question. Our goal with Dion Gijswijt was to settle this by constructing a strongly polynomial time algorithm, which we recently completed in [11]. Our algorithm is based on the concept of the Hungarian Method, and uses Chang et al.'s unweighted algorithm as a subroutine. We actually solve the problem of finding a maximum weight *prefect* fractional matching, which is equivalent to the original one. In the algorithm we maintain a dual feasible solution the support of which is a chain of flats, and check if there is a perfect fractional matching satisfying certain slackness conditions. This is done by running Vande Vate et al.'s algorithm for a different matroid and a subset of lines. If there is one, then that turns out to be a maximum weight perfect fractional matching, and the algorithm terminates. Otherwise, using the dual optimum provided by Vande Vate et al.'s algorithm, we perform a dual update, that improves on the dual objective value. This already implies a finite running time, since the dual objective improves by at least one half in every iteration. To prove a strongly polynomial running time, we need that Vande Vate et al.'s algorithm returns a special dual optimum what they call a "dominant cover", which can be used to prove that not only that the dual value improves in every iteration, but there is another, quite technical,

combinatorial measure of improvement, that guarantees a polynomial bound on the running time.

Theorem 4 (Gijswijt, Pap, 2008, [11])**.** *There is a polynomial time algorithm for maximum weight fractional matroid matching.*

Conclusions and remarks. The most surprising fact here is that, while matroid matching is tractable only for certain restricted classes of matroids, fractional matroid matching, even its weighted version, is tractable for arbitrary matroids. It is unclear why there is this distinction. Also note that one could pick any instance of matroid matching, and plug it in the fractional relaxation. There is a lot of room for further inverstigation of these plugging-ins, and thus learning about more applications.

References

[1] S. Chang, D. Llewellyn, and J. Vande Vate, *Matching 2-lattice polyhedra: finding a maximum vector*, Discrete Mathematics **237** (2001), 29–61.

[2] S. Chang, D. Llewellyn, and J. Vande Vate, *Matching 2-lattice polyhedra: duality and extreme points*, Discrete Mathematics **237(1–3)** (2001), 63–95.

[3] A. Frank, T. Jordán, and Z. Szigeti, *An orientation theorem with parity conditions*, Discrete Applied Mathematics **115** (2001) 37–47.

[4] M.L. Furst, J.L. Gross, and L.A. McGeoch, *Finding a maximum-genus graph embedding*, J. Assoc. Comput. Mach. **35(3)** (1988), 523–534.

[5] H.N. Gabow and M. Stallmann, *An augmenting paths algorithm for linear matroid parity*, Combinatorica **6(2)**, (1986), 123–150.

[6] P. Jensen and B. Korte, *Complexity of matroid property algorithms*, SIAM Journal on Computing **11** (1982), 184–190.

[7] L. Lovász, *The matroid matching problem*, in: Algebraic Methods in Graph Theory, Vol II (Szeged 1978; L.Lovász, V.T. Sós, eds.), Colloq. Math. Soc. János Bolyai, North-Holland (1981), 495–517.

[8] L. Lovász, *Matroid matching and some applications*, Journal of Combinatorial Theory, Series B **28** (1980), 208–236.

[9] M. Makai, G. Pap, and J. Szabó, *Matching Problems in Polymatroids Without Double Circuits*, IPCO XII, [in: Integer Prog. and Comb. Opt., M. Fischetti, D.P. Williamson eds.], LNCS **4513**, Springer (2007), 167-181.

[10] G. Pap, *A constructive approach to matching and its applications*, Ph.D. Thesis, Eötvös University, 2007

[11] D. Gijswijt and G. Pap, *An algorithm for weighted fractional matroid matching*, Egerváry Technical Report, EGRES TR-2008-11

[12] J. Vande Vate, *Fractional matroid matchings*, Journal of Combinatorial Theory, Series B **55** (1992), 133–145.

A Simple Combinatorial Algorithm for Submodular Function Minimization

Satoru Iwata
(joint work with James B. Orlin)

This paper presents a new simple algorithm for minimizing submodular functions. Let V be a finite nonempty set of cardinality n. A function f defined on the subsets of V is *submodular* if it satisfies

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y), \qquad \forall X, Y \subseteq V.$$

Submodular functions are discrete analogues of convex functions. Examples include cut capacity functions, matroid rank functions, and entropy functions. The first polynomial algorithm for submodular function minimization due to Grötschel, Lovász, and Schrijver [1] is based on the ellipsoid method. Recently, combinatorial polynomial algorithms have been developed [3, 5], and the current best weakly and strongly polynomial bounds [2, 4] are $O((n^4\mathrm{EO} + n^5)\log M)$ and $O(n^5\mathrm{EO} + n^6)$, where EO is the time for function evaluation and M is the maximum absolute value of the function values.

The initial variant of our algorithm minimizes integer valued submodular functions in $O(n^6\mathrm{EO}\log nM)$ time. The algorithm achieves this complexity without relying on the scaling technique nor on Gaussian elimination. With the aid of the Gaussian elimination procedure, the algorithm can be improved to run in $O((n^4\mathrm{EO} + n^5)\log nM)$ time. The strongly polynomial version of this faster algorithm runs in $O((n^5\mathrm{EO} + n^6)\log n)$ time for real valued general submodular functions. These are comparable to the best known running time bounds for submodular function minimization.

The new algorithm as well as its strongly polynomial version can be turned into fully combinatorial algorithms, which use only additions, subtractions, comparisons, and the oracle calls for function evaluation. The running time bounds of the resulting algorithms are $O(n^6(\mathrm{EO}+\log nM)\log nM)$ and $O((n^7\mathrm{EO}+n^8)\log n)$. These are the first fully combinatorial algorithms that do not rely on the scaling method. Moreover, the latter algorithm improves the best previous bound by a factor of n.

References

[1] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, **1** (1981), 169–197.

[2] S. Iwata, *A faster scaling algorithm for minimizing submodular functions*, SIAM Journal on Computing, **32** (2003), 833–840.

[3] S. Iwata, L. Fleischer, and S. Fujishige, *A combinatorial strongly polynomial algorithm for minimizing submodular functions*, Journal of the ACM, **48** (2001), 761–777.

[4] J. B. Orlin, *A faster strongly polynomial time algorithm for submodular function minimization*, Mathematical Programming, to appear.

[5] A. Schrijver, *A combinatorial algorithm minimizing submodular functions in strongly polynomial time*, Journal of Combinatorial Theory B, **80** (2000), 346–355.

Combinatorial Mixed-Integer Programming

Michele Conforti

We study the convex hull of a mixed-integer set S by expressing each continuous variable as the average of k integral variables. This allows us to model S as a pure integer set. We use the additional integral variables to strengthen the inequalities that describe S.

We concentrate on a mixed-integer set defined as follows: Given a bipartite graph $G = (U \cup V, E)$, a set $I \subseteq U \cup V$ and rational numbers b_{ij}, $ij \in E$, let

$$S^{(G,I)} = \{x \in \mathbb{R}^{U \cup V} \mid x_i + x_j \geq b_{ij} \quad ij \in E;\ x_i \in \mathbb{Z} \quad i \in I\}.$$

We show that the set $S^{(G,I)}$ is equivalent to the "network dual" set introduced and studied recently by Conforti, Di Summa, Eisenbrand and Wolsey [1]. Conforti et al. give an extended formulation for the polyhedron $\text{conv}(S^{(G,I)})$ and discuss cases in which the formulation is compact.

An *extended formulation* of a polyhedron P in the x-space is an inequality description $Ax + B\mu \geq d$ of a polyhedron Q in the (x, μ)-space such that P is the projection of Q in the x-space. We are interested in polyhedra P that are convex hulls of mixed integer sets such as $\text{conv}(S^{(G,I)})$. We say that an extended formulation is *compact* if the size of the system that describes Q is polynomially bounded in the size of the system of inequalities that describes the mixed integer set.

The smallest integer k such that $kb_{ij} \in \mathbb{Z}$ for all $ij \in E$ is important in the study of the polyhedron $\text{conv}(S^{(G,I)})$. Indeed the size of the extended formulation of $\text{conv}(S^{(G,I)})$ grows with k. For the case $k = 2$, Conforti, Gerards and Zambelli [3] have given an explicit description of $\text{conv}(S^{(G,I)})$ in the space of the x variables by computing the projection of the extended formulation. However the essential inequalities that describe $\text{conv}(S^{(G,I)})$ become more and more complicated as k grows.

Our goal is to describe the polyhedron $\text{conv}(S^{(G,I)})$ in the space of the x variables. To this end we derive an extended formulation for $\text{conv}(S^{(G,I)})$ that differs slightly from that proposed in [1] but whose projection cone is easier to analyze. Our principal result is a characterization of the structure of facet-defining inequalities when the graph G is a tree. In several cases in which such a characterization is known, including the mixing set studied by Günlük and Pochet [4] and the continuous mixing set studied by Van Vyve [5], this gives further insight into the inequalities, and it leads to the first explicit description for the continuous mixing set with flows treated by Conforti, Di Summa and Wolsey [2]. It also suggests the following conjecture, that we describe next.

The main open problem is the complexity of optimizing a linear function over $S^{(G,I)}$, which is equivalent to the membership problem for $\text{conv}(S^{(G,I)})$. More formally.

Problem 1. *Given $\bar{x} \in \mathbb{R}^{V(G)}$, decide whether $\bar{x}$ is in $\text{conv}(S^{(G,I)})$ or not.*

The problem is clearly in NP, since the fact that $\bar{x} \in \text{conv}(S^{(G,I)})$ admits a short certificate, namely at most $|V(G)|+1$ vectors in $S^{(G,I)}$ such that $\bar{x}$ is in their convex hull. In [1] it has been shown that, when G is a tree, optimizing a linear function over $S^{(G,I)}$ and the membership problem can be solved in polynomial time.

Denote by $\mathcal{T}$ the family of subtrees T of G such that $I_T = V(T) \cap I$ coincides with the set of leaves of T. We propose the following conjecture.

Conjecture 2.

$$\text{conv}(S^{(G,I)}) = \bigcap_{T \in \mathcal{T}} \text{conv}(S^{(T,I_T)}).$$

Conjecture 2, if true, would imply that Problem 1 is in coNP. Indeed, a certificate that $\bar{x} \notin \text{conv}(S^{(G,I)})$ would be given by a tree T such that $\bar{x} \notin \text{conv}(S^{(T,I_T)})$. By the above discussion, the latter fact can be decided in polynomial time.

For the case of $k = 2$, the validity of Conjecture 2 follows from the characterization of the facet defining inequalities of $\text{conv}(S^{(G,I)})$ [3]. However the conjecture is open for any $k \geq 3$.

References

[1] M. Conforti, M. Di Summa, F. Eisenbrand, and L.A. Wolsey, *Network formulations of mixed-integer programs*, accepted in Mathematics of Operations Research (2006).

[2] M. Conforti, M. Di Summa, and L. A. Wolsey, *The intersection of continuous mixing polyhedra and the continuous mixing polyhedron with flows*, in Proceedings of the 12th IPCO Conference, LNCS **4513** (2007), 352–366.

[3] M. Conforti, B. Gerards and G. Zambelli, *Mixed-integer vertex covers in bipartite graphs*, in Proceedings of the 12th IPCO Conference, LNCS **4513** (2007), 324–336.

[4] O. Günlük and Y. Pochet, *Mixing mixed integer inequalities*, Mathematical Programming **90** (2001), 429–457.

[5] M. Van Vyve, *The Continuous Mixing Polyhedron*, Mathematics of Operations Research **30** (2005), 441–452.

An Analysis of Mixed Integer Sets Based on Lattice Point Free Convex Sets

Robert Weismantel

(joint work with Kent Andersen and Quentin Louveaux)

We consider a polyhedron in $\mathbb{R}^n$ of the form

$$P := \text{conv}(\{v^i\}_{i \in V}) + \text{cone}(\{r^j\}_{j \in E}), \tag{1}$$

where V and E are finite index sets, $\{v^i\}_{i \in V}$ denotes the vertices of P and $\{r^j\}_{j \in E}$ denotes the extreme rays of P. We assume P is rational, *i.e.*, we assume $\{r^j\}_{j \in E} \subset \mathbb{Z}^n$ and $\{v^i\}_{i \in V} \subset \mathbb{Q}^n$.

We are interested in points in P that have integer values on certain coordinates. For simplicity assume the first $p > 0$ coordinates must have integer values, and let $q := n - p$. The set $N_I := \{1, 2, \ldots, p\}$ is used to index the integer constrained

variables and the set $P_I := \{x \in P : x_j \in \mathbb{Z} \text{ for all } j \in N_I\}$ denotes the mixed integer points in P.

The following concepts from convex analysis are needed (see [5] for a presentation of the theory of convex analysis). For a convex set $C \subseteq \mathbb{R}^n$, the interior of C is denoted $\text{int}(C)$, and the relative interior of C is denoted $\text{ri}(C)$ (where $\text{ri}(C) = \text{int}(C)$ when C is full dimensional).

We consider the generalization of *split sets* (see [3]) to lattice point free convex sets (see [4]). A split set is of the form $S^{(\pi,\pi_0)} := \{x \in \mathbb{R}^p : \pi_0 \leq \pi^T x \leq \pi_0 + 1\}$, where $(\pi, \pi_0) \in \mathbb{Z}^{p+1}$ and $\pi \neq 0$. Clearly a split set does not have integer points in its interior. In general, a lattice point free convex set is a convex set that does not contain integer points in its relative interior. Lattice point free convex sets that are maximal wrt. inclusion are known to be full dimensional rational polyhedra that can be written as the sum of a polytope $\mathcal{P}$ and a linear space $\mathcal{L}$. For simplicity, and to emphasize that lattice point free convex sets generalize split sets, we call lattice point free convex sets that are maximal wrt. inclusion for *split bodies*.

A lattice point free convex set is an object that assumes integrality of *all* coordinates. For *mixed* integrality in $\mathbb{R}^{p+q}$, we use a lattice point free convex set $C^x \subset \mathbb{R}^p$ to form a *mixed integer* lattice point free convex set $C \subset \mathbb{R}^n$ of the form $C := \{(x, y) \in \mathbb{R}^p \times \mathbb{R}^q : x \in C^x\}$. A *mixed integer split body* is then a mixed integer lattice point free convex set of the form $L := \{(x, y) \in \mathbb{R}^p \times \mathbb{R}^q : x \in L^x\}$, where L^x is a split body in $\mathbb{R}^p$.

An important measure in our analysis of the size of a mixed integer split body L is the *facet width* of L. The facet width measures how wide a mixed integer split body is parallel to a given facet. Specifically, given any facet $\pi^T x \geq \pi_0$ of a mixed integer split body L, the width of L along π is defined to be the number $w(L, \pi) := \max_{x \in L} \pi^T x - \min_{x \in L} \pi^T x$. The *max-facet-width* of a mixed integer split body L measures how wide L is along any facet of L, *i.e.*, the max-facet-width $w_f(L)$ of L is defined to be the largest of the numbers $w(L, \pi)$ over *all* facet defining inequalities $\pi^T x \geq \pi_0$ for L.

Any mixed integer lattice point free convex set $C \subseteq \mathbb{R}^n$ gives a relaxation of $\text{conv}(P_I)$

$$R(C, P) := \text{conv}(\{x \in P : x \notin \text{ri}(C)\})$$

that satisfies $\text{conv}(P_I) \subseteq R(C, P) \subseteq P$. The set $R(C, P)$ might exclude fractional points in $\text{ri}(C) \cap P$ and give a tighter approximation of $\text{conv}(P_I)$ than P.

Mixed integer split bodies L are the mixed integer lattice point free convex sets that give the tightest relaxations of P_I of the form above. Specifically, if $C, C' \subseteq \mathbb{R}^n$ are mixed integer lattice point free convex sets that satisfy $C \subseteq C'$, then $R(C', P) \subseteq R(C, P)$. For a general mixed integer lattice point free convex set C, the set $R(C, P)$ may not be a polyhedron. However, it is sufficient to consider mixed integer split bodies, and we show $R(L, P)$ is a polyhedron when L is a mixed integer split body.

Observe that the set of mixed integer split bodies with max-facet-width equal to one are exactly the split sets $S^{(\pi,\pi_0)} = \{x \in \mathbb{R}^n : \pi_0 \leq \pi^T x \leq \pi_0 + 1\}$, where

$(\pi, \pi_0) \in \mathbb{Z}^{n+1}$, $\pi_j = 0$ for $j > p$ and $\pi \neq 0$. In [3], Cook et. al. considered the set of split sets

$$\mathbf{L}^1 := \{L \subseteq \mathbb{R}^n : L \text{ is a mixed integer split body satisfying } w_f(L) \leq 1\}$$

and showed that the *split closure*

$$\mathrm{SC}^1 := \cap_{L \in \mathbf{L}^1} R(L, P)$$

is a polyhedron. A natural generalization of the split closure is to allow for mixed integer split bodies that have max-facet-width larger than one. For any $w > 0$, define the set

$$\mathbf{L}^w := \{L \subseteq \mathbb{R}^n : L \text{ is a mixed integer split body satisfying } w_f(L) \leq w\}$$

of mixed integer split bodies with max-facet-width at most w. We define the w^{th} *split closure* to be the set

$$\mathrm{SC}^w := \cap_{L \in \mathbf{L}^w} R(L, P).$$

We prove that the w^{th} split closure is a polyhedron. The proof is based on an analysis of cutting planes from an inner representation of the linear relaxation P. In fact, our proof does not use an outer description of P at all. Many of our arguments are obtained by generalizing results of Andersen et. al. [1] from the first split closure to the w^{th} split closure.

Given a family $\{(\delta^l)^T x \geq \delta_0^l\}_{i \in I}$ of cutting planes, we provide a sufficient condition for the set $\{x \in P : (\delta^l)^T x \geq \delta_0^l \text{ for all } l \in I\}$ to be a polyhedron. This condition concerns the number of intersection points between hyperplanes defined from the inequalities $\{(\delta^l)^T x \geq \delta_0^l\}_{i \in I}$ and line segments either of the form $\{v^i + \alpha r^j : \alpha \geq 0\}$, or of the form $\{\beta v^i + (1 - \beta) v^k : \beta \in [0, 1]\}$, where $i, k \in V$ denote two vertices of P and $j \in E$ denotes an extreme ray of P. We then show that this sufficient condition is satisfied by any family of split cuts that have bounded max-facet-width from which it follows that the w^{th} split closure is a polyhedron.

Finite cutting plane proofs for the validity of an inequality for P_I can be designed by using mixed integer split bodies. A measure of the complexity of a finite cutting plane proof is then the max-facet-width of the mixed integer split body with the largest max-facet-width in the proof. A measure of the complexity of a valid inequality $\delta^T x \geq \delta_0$ for P_I is the smallest integer $w(\delta, \delta_0)$ for which there exists a finite cutting plane proof of validity of $\delta^T x \geq \delta_0$ for P_I that only uses split bodies with max-facet-width at most $w(\delta, \delta_0)$. We give a formula for $w(\delta, \delta_0)$ that explains geometrically why mixed integer split bodies of large width size can be necessary.

References

[1] K. Andersen, G. Cornuéjols, and Y. Li, *Split closure and intersection cuts*, Mathematical Programming A **102** (2005), 457–493.

[2] E. Balas, *Disjunctive Programming*, Annals of Discrete Mathematics **5** (1979), 517–546.

[3] W.J. Cook, R. Kannan, and A. Schrijver, *Chvátal closures for mixed integer programming problems*, Mathematical Programming **47** (1990), 155–174.

[4] L. Lovász, *Geometry of numbers and integer programming*, Mathematical Programming, Recent Developments and Applications (1989).
[5] R.T. Rockafellar, *Convex Analysis*, Princeton Landmarks in Mathematics and Physics (1970).

Inverse 1-median Problems

Rainer E. Burkard
(joint work with Mohammadreza Galavii and Elisabeth Gassner)

Given n points $P_1, P_2, ..., P_n$ in a metric space (X, d) and positive weights $w_1, w_2, ..., w_n$ the 1-*median problem* asks for a point $P \in X$ which minmizes

$$\sum_{i=1}^{n} w_i d(P_i, P).$$

The 1-median problem plays an important role in location theory, e.g. for locating a supply center such that the distribution cost to the customers $P_1, P_2, ..., P_n$ becomes minimum.

In the *inverse* 1-*median problem* a point P_0 is given in addition to the points $P_1, P_2, ..., P_n$. The weight of these points have to be modified within given bounds $[\underline{w}_i, \overline{w}_i]$ such that P_0 becomes a 1-median and the sum of weight changes [or: the cost for the weight changes] is as small as possible. The inverse 1-median problem in trees with positive weights can be solved by a continuous knapsack algorithm in $O(n)$ time. The inverse 1-median problem on a cycle can be solved in $O(n^2)$ time by applying mehtods from computational geometry, see Burkard, Pleschiutschnig and Zhang [2]. The same authors developed also a greedy-like $O(n \log n)$ time algorithm for the inverse 1-median problem in the plane provided the distances between the points are measured in the Manhattan or maximum metric, see [1]. In the following we outline a fast method for solving the inverse 1-median problem in the Euclidean plane.

Concerning the Euclidean 1-median problem the resultant $R(P)$ of point P in the plane is defined by

$$R(P) := \sum_{i=1}^{n} \frac{w_i}{d(P_i, P)}(P_i - P) \quad \text{if} \quad P \neq P_j, \ j = 1, 2, ..., n$$

and for $P = P_j$ for some $j = 1, 2, ..., n$,

$$R(P) := \max(||R_j|| - w_j, 0)\frac{R_j}{||R_j||}$$

where

$$R_j := \sum_{\substack{i=1 \\ i \neq j}}^{n} \frac{w_i}{d(P_i, P_j)}(P_i - P_j).$$

Differential calculus tells that P_0 is a 1-median iff the resultant $R(P_0) = 0$.

In case that $P \neq P_j$ we can assume that P_0 lies in the interior of the convex hull of the points P_i, $i = 1, 2, ..., n$ and that P_0 is the origin. Next we project the given points on the unit circle. Note that possibly two or more different points may coincide on the unit cycle. The point $P_0 = (0, 0)$ is a Euclidean 1-median if and only if

$$R_x(w) := \sum_{i=1}^{n} w_i x_i = 0, \tag{1}$$

$$R_y(w) := \sum_{i=1}^{n} w_i y_i = 0. \tag{2}$$

Since the Euclidean distance is invariant with respect to rotation and reflection, we can always assume that

$$R_x(w) = 0 \quad \text{and} \quad R_y(w) \leq 0. \tag{3}$$

If $R_y(w) = 0$, then the weights w_i, $i = 1, 2, ..., n$, provide an optimal solution. Therefore we assume in the following $R_y(w) < 0$. We call $|R_y(w)|$ the *optimality gap* $G(w)$. The solution method is based on a sequence of weight changes. If by chance one of the given points coincides with $A := (0, 1)$ or $B := (0, -1)$, then we can decrease $G(w)$ by changing the weight of this point without violating $R_x(w) = 0$. Otherwise we reduce the optimality gap by simultaneously changing the weights of two points, say point P_s and point P_t. If we want to decrease the optimality gap by δ, the weight change δ_s of point P_s and δ_t of point P_t have to fulfill according to (1) and (2):

$$x_s \delta_s + x_t \delta_t = 0, \tag{4}$$

$$y_s \delta_s + y_t \delta_t = \delta. \tag{5}$$

Then Cramer's rule yields

$$\delta_s = -\frac{x_t}{x_s y_t - x_t y_s}\delta, \tag{6}$$

$$\delta_t = \frac{x_s}{x_s y_t - x_t y_s}\delta, \tag{7}$$

δ is to be chosen as large as possible such that $\delta \leq G(w)$ and the weight bounds for w_s and w_t are fulfilled. The maximal possible value of δ is called the augmentation value δ_{st}. If $\delta_{st} > 0$, we call (P_s, P_t) an *augmenting pair*. The cost of an augmentation δ by the pair (P_s, P_t) is given by $|\delta_s| + |\delta_t|$. We can evaluate the efficiency of the weight change incurred by the augmenting pair (P_s, P_t) by defining the *efficiency* e_{st} as fraction of the gain in closing the optimality gap divided by costs:

$$e_{st} := \frac{\delta}{|\delta_s| + |\delta_t|}.$$

An augmenting pair with maximum efficiency is called *maximal augmenting pair*. For each point $P_i = (x_i, y_i)$ let $\frac{y_i}{x_i}$ denote the *slope* of P_i. Then (P_s, P_t) is a maximal augmenting pair, if P_s has maximum slope and P_t has a minimum slope. If an augmenting pair with maximal efficiency is used for a pair exchange then there

exists an optimal solution which can be obtained without revoking the weight increase or decrease of this weight transformation. This leads to a greedy-like algorithm. The algorithm terminates after at most n weight exchanges. It yields a solution where at most two of the changed weights lie strictly between their lower and upper bound.

This method solves the unit cost model. Successively choosing maximal augmenting pairs in case of arbitrary linear cost for the weight changes, however, does not yield an optimal solution.

Now we consider the case that the prespecified point coincides with one of the given points. According to the optimality condition point P_0 is 1-median if and only if

$$R_x^2(w) + R_y^2(w) \leq w_0^2$$

holds. This condition does not lead to a convex problem. However, it is possible to fix the optimal weight w_0^* in advance to

$$w_0^* = \min\left\{\overline{w}_0, \sqrt{R_x^2(w) + R_y^2(w)}\right\}.$$

The remaining problem is convex and can be solved by any algorithm for convex programming.

References

[1] R.E. Burkard, C. Pleschiutschnig, and J.Z. Zhang, *Inverse median problems*, Discrete Optimization **1** (2004), 23–39.

[2] R.E. Burkard, C. Pleschiutschnig, and J.Z. Zhang, *The inverse* 1*-median problem on a cycle*, Discrete Optimization **5** (2008), 242–253.

Computational Progress in Linear and Mixed Integer Programming: 1988–2008

Robert E. Bixby

We begin by reviewing the improvements in linear programming (LP) software from 1988 to 2004, concluding that the combined algorithmic (3300x) and hardware (1600x) improvements have yielded approximately a 5 million fold speedup. This is the good news. The bad news is that relatively little has happened in the four years since 2004. Indeed there is both a real opportunity and a need for further research in this important area.

Finally we review computational progress in mixed integer programming (MIP) from 1990 to 2008. We do this by reporting on extensive testing with the well-known CPLEX mixed integer solver, using a test set of 1852 real-world models and employing 12 successive CPLEX releases. We conclude, among other things, that algorithmic improvements alone have accounted for a remarkable 30000x speedup over this period, with most of that effect coming from the use of the extensive cutting-plane and other mathematical techniques that have been developed by the combinatorial optimization community in the last half century. Moreover, in

contrast to LP, research in MIP and the resulting computational improvements seem to be continuing unabated.

On the Separation of Rank-1 Gomory Mixed-Integer Cuts

Sanjeeb Dash
(joint work with Marcos Goycoolea)

In this talk, we describe a method to generate useful rank-1 Gomory mixed-integer cuts from different tableaus of the initial linear progamming (LP) relaxation of a mixed-integer program (MIP). Gomory mixed-integer (GMI) cutting planes (or *cuts*), introduced by Gomory [14], were not widely used to solve general MIPs till Balas, Ceria, Cornuéjols, and Natraj [2] showed how to use them in an effective manner. Subsequent computational studies [5, 6] confirmed the usefulness of GMI cuts in solving practical MIPs, and they are now considered to form the most important class of general cutting planes for solving MIPs. Following these papers, GMI cuts are typically generated from rows of an optimal simplex tableau associated with the LP relaxation of an MIP, usually in rounds. A *round* of cuts consists of generating GMI cuts from the rows of the optimal tableau of the current LP relaxation, and augmenting the current LP relaxation with the cuts violated by the basic solution defined by the optimal tableau. Some factors which limit very extensive generation of GMI cuts in solving practical MIPs are that after many rounds of cuts from optimal tableau rows, one often obtains high-rank GMI cuts which are invalid [16] (because of floating point-computations with limited accuracy), or which are very dense or have large variation in coefficient magnitudes, thus leading to hard-to-solve LP relaxations when these cuts are added.

Marchand and Wolsey [15] showed that GMI cuts based on linear combinations of constraints which are different from optimal tableau rows can be useful. Fischetti and Lodi [11] recently showed that strong lower bounds for practical pure integer programs (with an objective function to be minimized) in the MIPLIB 3.0 library [4] can be obtained by optimizing over the Chvátal closure, i.e., by iteratively generating violated rank-1 Gomory-Chvátal cuts derived from arbitrary combinations of the linear constraints defining the integer programs. Balas and Saxena [3], and Dash, Günlük, and Lodi [10] subsequently showed that one can obtain strong lower bounds for the MIP instances in the MIPLIB 3.0 library by approximately optimizing over the GMI closure. Approximate optimization over the GMI closure associated with an MIP is performed by iteratively generating violated rank-1 GMI cuts based on general linear combinations of the inequalities defining the MIP. However, given an arbitrary point, finding a violated GMI cut is NP-hard [7]. In the papers of Balas and Saxena, and Dash, Günlük, and Lodi, an auxiliary MIP is solved to find a violated GMI cut; this process can be very time-consuming, and therefore the overall approach to obtaining strong lower bounds for MIP instances in these papers is unlikely to lead to a practical method for solving MIPs.

The papers discussed above show that rank-1 GMI cuts can be used to obtain strong lower bounds for practical MIPs. We show that the family of rank-1 GMI cuts derived from rows of non-optimal tableaus of the initial LP relaxation of an MIP form a very useful subclass of all rank-1 GMI cuts. Further, we give a heuristic to find a violated GMI cut from this subclass, given an arbitrary point to be separated from the GMI closure. In particular, given an input vector x which lies on some face f of the initial LP relaxation of an MIP, our method explores both feasible and infeasible tableaus of the LP relaxation associated with f, and adds all GMI cuts based on rows of these tableaus which are violated by x. We iterate this process, and generate multiple rounds of rank-1 GMI cuts for an MIP. The effect of multiple rounds of GMI cuts has been studied in the literature (for example by Balas et. al. [2], and in [8]); however GMI cuts generated in these papers are not restricted to be rank-1 cuts beyond the GMI cuts generated in the first round. As discussed earlier, high-rank GMI cuts generated via multiple rounds of cuts often lead to numerical difficulties, some of which are avoided by our method. Our method is also much faster than the MIP-based separation methods in the papers of Balas and Saxena, and Dash, Günlük and Lodi, but is not much weaker for MIPLIB 3.0 problems in terms of the quality of lower bounds obtained by iterated generation of violated GMI cuts. More specifically, our heuristic closes 59.80% of the integrality gap, on the average; the corresponding numbers in the papers of Dash, Günlük, and Lodi [10] and Balas and Saxena [3] are 62.53% and 76.52%, respectively. A single round of GMI cuts closes 25.29% of the integrality gap. Further, we are able to apply our heuristic to the larger MIPs in the MIPLIB 2003 library [1] and close 40.21% of the integrality gap, on the average, as opposed to 23.32% if one round of GMI cuts is used.

References

[1] T. Achterberg, T. Kock, and A. Martin, MIPLIB 2003, ZIB Technical report 05-28 (2005).

[2] E. Balas, S. Ceria, G. Cornuejols, and N. Natraj, *Gomory cuts revisited*, Operations Research Letters **19** (1996), 1–9.

[3] E. Balas and A. Saxena, *Optimizing over the split closure*, Mathematical Programming **113** (2008), 219–240.

[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] R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling, *Mip: Theory and practice–closing the gap*, in *System Modelling and Optimization* (1999), 19–50.

[6] R. Bixby, Z. Gu, E. Rothberg, and R. Wunderling, *The sharpest cut: The Impact of Manfred Padberg and his work*, chapter 18 (Mixed-Integer Programming: A Progress report), MPS-SIAM Series on Optimization (2004), 309–323.

[7] A. Caprara and A. Letchford, *On the separation of split cuts and related inequalities*, Mathematical Programming **94** (2003) 279–294.

[8] W. Cook, S. Dash, M. Goycoolea and R. Fukasawa, *Numerically safe Gomory mixed-integer cuts*, Manuscript (2008).

[9] S. Dash and O. Günlük, *On the strength of Gomory mixed-integer cuts as group cuts*, Mathematical Programming **115** (2008), 387–407.

[10] S. Dash, O. Günlük, and A. Lodi, *MIR closures of polyhedral sets*, Mathematical Programming (2008).

[11] M. Fischetti and A. Lodi, *Optimizing over the first Chvátal closure*, Mathematical Programming B **110** (2007) 3–20.
[12] M. Fischetti and C. Saturni, *Mixed integer cuts from cyclic groups*, Mathematical Programming **109** (2007), 27–53.
[13] R. Fukasawa and M. Goycoolea, *On the exact separation of mixed-integer knapsack cuts*, in Proceedings of IPCO 2007, LNCS **4513** (2007), 225–239.
[14] R. E. Gomory, *An algorithm for the mixed integer problem*, RM-2597, The Rand Corporation (1960).
[15] H. Marchand and L.A. Wolsey, *Aggregation and mixed integer rounding to solve MIPs*, Operations Research **49** (2001), 363–371.
[16] F. Margot, *Testing cut generators for MILP*, Optima **77** (2008), 6–9.
[17] G. L. Nemhauser and L. A. Wolsey, *A recursive procedure for generating all cuts for 0-1 mixed integer programs*, Mathematical Programming **46** (1990), 379–390.

MIP Computations on Networks with PDEs

Alexander Martin
(joint work with Björn Geißler and Antonio Morsi)

We consider the optimization of dynamical transport networks. For a network to be a transport network, we assume some flow through the components of the network, which should be optimized under the consideration of several side constraints. In one of the simplest cases, this may lead to a minimum cost flow problem, where a cost is assigned for the use of each connection and the goal is to achieve a certain flow between distinct nodes of the network at lowest cost. Since this problem can be described in terms of a mathematical model involving only linear constraints and a linear objective function, it can be solved by using algorithms from linear optimization, which are proven to be fast and to guarantee to find the global optimum. The situation gets more complex in cases where the network contains some kinds of switching components, i.e., components which can be in different discrete states, such as valves, pumps or compressors in water or gas networks. In order to reflect discrete processes in a mathematical model, one typically introduces binary variables and additional linear constraints involving these variables. This leads to a mixed integer linear optimization problem, which in general is harder to solve than a linear program. Fortunately, there are still algorithms available which guarantee to solve these kinds of problems to global optimality. If we consider a situation where the transported entities underly nonlinear dynamics, typically according to physical laws, it seems like the guarantee of global optimality, provided by continuous and mixed integer programming solvers, is no longer at hand. Moreover, classical algorithms for nonlinear optimization problems are not able to handle discrete variables and constraints in an efficient way. Our approach is based on piecewise linearizing of nonlinearities and the modeling of piecewise linear functions in terms of linear constraints. This brings us into a situation where we can apply global optimization algorithms from mixed integer linear programming to the optimization of dynamical transport networks. Many engineering problems such as production planning problems, traffic problems, the problem of gas network optimization and the problem of water supply network

optimization can be stated as an optimization problem on a dynamic transport network.

In this talk we exactly study problems arising from these four application areas. We introduce a network model suitable for the general dynamics occurring in these transport networks. The dynamics in these networks are mainly described by a coupled set of (hyperbolic) partial differential equations leading to a PDE optimization problem. For the PDE equations, we also give a discretization scheme in order to come up with an appropriate mixed integer linear model. We computationally compare both models for each of the four application areas (production planning, traffic, water and gas network optimization problem). It turns out that the characteristics of the PDE constraints lead to different approximation qualities of the MIP models. For the production planning problem, where the flow basically depends linear on the pressure, the MIP model achieves exactly the same solution as the PDE model [2]. From the traffic, via the water to the gas model the dynamics increase. It comes at no surprise that the discretization and thus the piecewise linearization must be finer the more the dynamics increase in order to achieve a reasonable good approximation quality [1, 3]. The computational results show that this is possible, although the running times increase partially significantly, even when applying the newest techniques for piecewise linearizing nonlinear functions [4]. On the other hand applying MIP techniques allows to compute globally optimal solutions. And indeed even for very small symmetric gas network instances we show that nonlinear optimization methods fail to find the global optimum, whereas the MIP approach does. This shows that there is a lot of potential for MIP techniques to help PDE optimization to achieve better or even optimal solutions. The future lies in the combination of both techniques to solve larger real sized problems. To achieve this goal a long way needs to be gone and many problems on the border between the PDE and MIP optimization worlds need to be solved.

References

[1] A. Fügenschuh, M. Herty, A. Klar, and A. Martin, *Combinatorial and continuous models for the optimization of traffic flows on networks*, SIAM Journal on Optimization **16** (2006), 1155–1176.

[2] A. Fügenschuh, S. Göttlich, M. Herty, A. Klar, and A. Martin, *A discrete optimization approach to large scale supply networks based on partial differential equations*, SIAM Journal on Scientific Computing **30** (2008), 1490–1507.

[3] A. Martin, M. Möller, and S. Moritz, *Mixed integer models for the stationary case of gas network optimization*, Mathematical Programming B **105** (2006), 563–582.

[4] J.P. Vielma, S. Ahmed and G.L. Nemhauser, *Mixed-integer models for nonseparable piecewise linear optimization: unifying framework and extensions.* Submitted for publication (2008).

Integer Programming and Geometry of Numbers

Friedrich Eisenbrand

Integer programming is not only related to Combinatorics but also to Number Theory, especially the Algorithmic Geometry of Numbers, as we shortly explain.

The *greatest common divisor* of two integers $a_0, a_1 \in \mathbb{N}_+$ is the largest integer, which divides both a_0 and a_1. The greatest common divisor can be efficiently computed with the *Euclidean algorithm*, see, e.g. [6, 1]. It computes the remainder sequence $a_0, a_1, \ldots, a_{k-1}, a_k \in \mathbb{N}_+$, where $a_i,\ i \geq 2$ is given by $a_{i-2} = a_{i-1}q_{i-1} + a_i$, $q_i \in \mathbb{N},\ 0 < a_i < a_{i-1}$, and a_k divides a_{k-1} exactly. Then $a_k = \gcd(a_0, a_1)$. The Euclidean algorithm can be interpreted as a reduction algorithm. The 2-dimensional basis reduction algorithm of Lagrange works along the same lines.

Now, the connection of integer linear programming with algorithmic number theory reveals itself already in the following theorem, which is proved at the beginning of every course on elementary number theory, see, e.g. [9].

> Let $a, b \in \mathbb{Z}$ be integers which are not both equal to zero. The greatest common divisor $\gcd(a, b)$ is the smallest integer in the set

$$\min\{a\,x + b\,y \colon x, y \in \mathbb{Z},\ a\,x + b\,y \geq 1\}. \tag{1}$$

The problem to find the minimum in (1) can be modeled as an integer program in two variables and one constraint.

$$\begin{array}{ll} \min & a\,x + b\,y \\ & a\,x + b\,y \geq 1 \\ & x, y \in \mathbb{Z}. \end{array}$$

This already shows that efficient methods for integer programming with a fixed number of variables, must incorporate *reduction techniques*, which prototypical appear already in the Euclidean algorithm. This is also the case for Lenstra's algorithm [8], which solves integer programming problems in fixed dimension in polynomial time.

In this talk we review some classical results and outline recent results on this interplay between Combinatorics, Complexity and Geometry of Numbers. We begin with Minkowski's theorem and Dirichlets theorem on simultaneous Diophantine approximation. Then we consider the complexity of simultaneous Diophantine approximation. Lagarias [7] showed that simultaneous Diophantine approximation is intractable. This was strengthened to an inapproximability result by Rössner and Seifert [10]. We then mention a recent result of Rothvoß [11] who extends these results to to directed Diophantine approximation, which was for example considered in [4]. With this result one can prove that optimizing over the Mixing set is NP-hard. Polynomial time special cases of Mixing set were previously presented in [12, 2].

Finally we mention recent results on parametric integer programming [3]. This is concerned with the following problem: Given a rational matrix $A \in \mathbb{Q}^{m \times n}$ and a rational polyhedron $Q \subseteq \mathbb{R}^{m+p}$, decide if for all vectors $b \in \mathbb{R}^m$, for which there exists an integral $z \in \mathbb{Z}^p$ such that $(b, z) \in Q$, the system of linear inequalities

$Ax \leq b$ has an integral solution. We show that there exists an algorithm that solves this problem in polynomial time if p and n are fixed. This extends a result of Kannan [5], who established such an algorithm for the case when, in addition to p and n, the affine dimension of Q is fixed.

We pose several open problems, one of which is the following. Given a matrix $A \in \mathbb{Z}^{m \times n}$, where n is fixed and a polyhedron $Q \subseteq \mathbb{R}^m$, find a b such that the number integer points in the polyhedron $\{x \in \mathbb{R}^n \colon Ax \leq b\}$ is minimal. Is this problem in P?

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley (1974).

[2] M. Conforti, M. Di Summa, and L. A. Wolsey, *The mixing set with divisible capacities*, in *IPCO* (2008), 435–449.

[3] F. Eisenbrand and G. Shmonin, *Parametric integer programming in fixed dimension*, Mathematics of Operations Research (2008), to appear.

[4] M. Henk and R. Weismantel, *Diophantine approximations and integer points of cones*, Combinatorica **22(3)** (2002), 401–407.

[5] R. Kannan, *Lattice translates of a polytope and the Frobenius problem*, Combinatorica **12(2)** (1992), 161–177.

[6] D.E. Knuth, *The art of computer programming*, volume 2, Addison-Wesley (1969).

[7] J. C. Lagarias, *The computational complexity of simultaneous diophantine approximation problems*, SIAM Journal on Computing **14(1)** (1985), 196–209.

[8] H. W. Lenstra, *Integer programming with a fixed number of variables*, Mathematics of Operations Research **8(4)** (1983), 538–548.

[9] I. Niven, H. S. Zuckerman, and H. L. Montgomery, *An Introduction to the Theory of Numbers*, Wiley (1991).

[10] C. Rössner and J.-P. Seifert, *Approximating good simultaneous Diophantine approximations is almost NP-hard*, in *Mathematical foundations of computer science 1996 (Cracow)*, LNCS **1113** (1996), 494–505.

[11] T. Rothvoß, *Hardness of mixing set*, Manuscript (2008).

[12] M. Zhao and I. R. de Farias, Jr, *The mixing-MIR set with divisible capacities*, Mathematical Programming A **115(1)** (2008), 73–103.

Exact and Approximation Results for Crossing Minimization

Petra Mutzel

(joint work with Christoph Buchheim, Markus Chimani, Carsten Gutwenger, Peter Hliněný, Michael Jünger, Immanuel Bomze)

The crossing number problem asks for the minimum number of edge crossings $cr(G)$ which can be achieved by drawing a given graph in the 2-dimensional plane. It has originally been stated by Turan during the second world war. Since then, a number of interesting theoretical results has been obtained. Despite the many published papers on this problem and its variants, the crossing number is only known for very few graph classes, e.g., even for complete graphs it is only known for graphs with up to 12 vertices. Garey and Johnson [6] have shown that the decision version of the crossing number problem is NP-complete. No polynomial

time algorithm is known for approximating $cr(G)$ for general graphs within some non-trivial factor. For a bibliography of the crossing number, see [9]. In this talk, we will present (1) some recent approximation results for certain graph classes as well as (2) exact results on the computation of $cr(G)$ for general (sparse) graphs.

(1) The approximation results concern the graph classes of *almost planar graphs* and *apex graphs*. A graph $G = (V, E)$ is called *almost planar* if G is non-planar, but there does exist an edge $e \in E$ so that $G - \{e\}$ is planar. Given a planar embedding Π of the remaining graph $G - \{e\}$, the edge e can be re-inserted with the minimal number of crossings via a shortest path in the extended geometric dual graph of Π. Gutwenger, Mutzel, and Weiskircher[7] have presented a linear time algorithm (based on the data structure of SPQR-trees) which is able to find the optimal embedding Π_0 of $G - \{e\}$, so that inserting e into Π_0 leads to a crossing minimum drawing over the set of all possible planar embeddings Π. The natural question arises, if this approach does approximate the crossing number $cr(G)$ by some small factor. And indeed, Hliněný and Salazar [8] have shown that the above algorithm provides crossing numbers of at most $\Delta(G - \{e\})cr(G)$, where $\Delta(G)$ denotes the maximum degree of G. This number has later been improved to $(\Delta(G - \{e\})/2)cr(G)$ by Cabello and Mohar [2]. This provides the first constant approximation algorithm for nearly planar graphs with bounded degree graphs. Very recently, these results could be generalized to apex-graphs. A graph $G = (V, E)$ is called an *apex graph* if G is non-planar, but there does exist a vertex $v \in V$ so that $G - \{v\}$ is planar. Chimani, Gutwenger, Mutzel, and Wolf [3] have shown that v and all its incident edges can be re-inserted into an optimal embedding Π_0 of $G - \{v\}$ (which the algorithm will identify) with the minimum number of crossings in polynomial time. Chimani, Hliněný, and Mutzel [4] have shown that this algorithm will find solutions which are at most a factor of $deg(v)\Delta(G - \{v\})/2$ away from the optimum solution $cr(G)$. Both approximation results are (almost) tight: for almost planar graphs, there is an example showing that the approximation factor can be reached, while for apex graphs the example is still a factor of 2 away. Some open questions arise:

- Is there a polynomial time algorithm for computing the crossing number $cr(G)$ for almost planar graphs? Cabello and Mohar [2] have shown that the weighted version is NP-hard.
- Can the above results be generalized, e.g., for graphs that are planar after deleting a fixed number of edges?

(2) The exact results concern two approaches for computing the exact crossing number for general graphs. The approaches are based on two integer linear programming (ILP) formulations of the crossing number problem that can be solved by branch-and-cut algorithms. The first ILP formulation, called the *subdivision crossing minimization approach* (SCM), optimizes over the set of all simple drawings [1]. A drawing is called *simple* if every edge is only crossed at most once. In order to provide an optimal solution for $cr(G)$, we need to subdivide all edges in G into a path of length $|E|$. The variables $x_{e,f}$ are associated with all non-adjacent edge pairs $(e, f) \in E^2$. The constraints come essentially from KuratowskiÂ´s

theorem stating that a graph G is planar if and only if it does not contain a subdivision of $K_{3,3}$ or K_5. The ILP contains 0/1-constraints, Kuratowski-constraints, and simple constraints. Our second approach [5], the ordering-based ILP model (OCM) is not restricted to simple drawings, so we do not need to subdivide the edges. However, we introduce additional linear ordering variables y_{efg} for each edge $e \in E$ that may be crossed more than once in the crossing minimum solution of $G = (V, E)$. As constraints we have 0/1-constraints, linear ordering constraints, Kuratowski constraints, and linking constraints between the x and y variables.

We solve both ILP models with branch-and-cut algorithms. In order to get these algorithms to work in practice, we needed to come up with new preprocessing techniques as well as new combinatorial column generation methods. Our computational experiments on a benchmark set of about 11,000 graphs show that we can compute the exact crossing numbers for general sparse graphs with up to 100 vertices and crossing number up to 37 within 30 minutes. It seems that the second formulation based on linear ordering dominates our first ILP model. For most instances, we need far more variables in our SCM model than in the OCM model. We find it surprising that in the OCM model only very few y-variables are needed in order to find the optimum solution.

We also did experiments for special graph classes. While we could verify the crossing number of the complete graph on 11 (and 12) vertices (with an alternative optimum solution), one more vertex is still a challenge. On the other hand, we are able to compute the crossing number of generalized Petersen graphs $P_{n,4}$ up to $n = 44$ which was unknown before. Based on our computed results, we came up with a conjecture of the crossing number of this graph class. Moreover, we are confident that we are about to be able to compute the crossing number of the smallest toroidal grid graph $T_{8,8}$ whose crossing number is still unknown (and conjectured to be 48).

References

[1] C. Buchheim, M. Chimani, D. Ebner, C. Gutwenger, M. Jünger, G. W. Klau, P. Mutzel, and R. Weiskircher, *A branch-and-cut approach to the crossing number problem*. In E. Balas, A. Hoffman, and T. McCormick, editors, *Special issue dedicated to the memory of George B. Dantzig*, Discrete Optimization **5(2)** (2008), 373–388.

[2] S. Cabello, B. Mohar, *Crossing and weighted crossing number of near planar graphs*, In Graph Drawing 2008; LNCS, Springer, to appear.

[3] M. Chimani, C. Gutwenger, P. Mutzel, C. Wolf, *Inserting a vertex into a planar graph*. Symposium on Discrete Algorithms SODA 2009, SIAM (2009), to appear.

[4] M. Chimani, P. Hliněný, P. Mutzel, *Vertex insertion approximates the crossing number for apex graphs*. submitted.

[5] M. Chimani, P. Mutzel, I. Bomze. *Solving the crossing number problem by a linear-ordering-based ILP*. in: Halperin, D., Mehlhorn, K. (eds.), 16th Annual European Symposium on Algorithms (ESA), LNCS 5193, Springer (2008), 284-296.

[6] M.R. Garey, D.S. Johnson, *Crossing number is NP-complete*, SIAM J. Algebraic Discrete Methods **4** (1983), 312–316.

[7] C. Gutwenger, P. Mutzel, R. Weiskircher, *Inserting an edge into a planar graph*. Algorithmica **41** (2005), 289–308.

[8] P. Hliněný and G. Salazar, *On the crossing number of almost planar graphs.* In Graph Drawing 2006; LNCS 4372, Springer (2007), 162–173.
[9] I. Vrťo, Bibliography on crossing numbers. Available online at `http://www.ifi.savba.sk/~imrich/crobib.ps`.

Disjoint Arborescences Spanning Convex Sets in a Graph

Satoru Fujishige

Recently Kamiyama, Katoh, and Takizawa [3] have shown a theorem (KKT theorem for short in the sequel) on packing arc-disjoint arborescences that is a proper extension of Edmonds' theorem [2] on disjoint spanning branchings, which is described as follows. (The precise definitions of terms used here will be given later.)

Let $G = (V, A)$ be a directed graph with a vertex set V and an arc set A. For any vertex $v \in V$ we denote by $R^+_G(v)$ the set of vertices reachable from v by directed paths in G. Given roots $r_i \in V$ $(i \in I)$, KKT theorem gives a characterization of the existence of a set of arc-disjoint arborescences H_i $(i \in I)$ such that for each $i \in I$ arborescence H_i has a root r_i and exactly spans $R^+_G(r_i)$.

In this talk we show a further extension of KKT theorem, which makes clear an essential rôle played by a reachability condition in the theorem. The right concept required for the further extension is "convexity" instead of "reachability."

For more information about disjoint arborescences, their extensions, and related topics see [4, Part V] and [1].

A *branching* in G is a subgraph $H = (U, B)$ of G without any cycle such that every vertex u in U has in-degrees at most one in H. Each connected component of branching H has a unique vertex, called a *root*, that has the in-degree equal to zero in H. A connected branching is called an *arborescence*, which has a single root.

For any vertex $v \in V$ we denote by $R^+_G(v)$ the set of vertices reachable from v by directed paths in G and by $R^-_G(v)$ the set of vertices from which v is reachable by a directed path in G. Also define for any $W \subseteq V$

$$R^+_G(W) = \bigcup\{R^+_G(v) \mid v \in W\}, \quad R^-_G(W) = \bigcup\{R^-_G(v) \mid v \in W\}.$$

A vertex subset W is called a *convex set* in G if we have $W = R^+_G(W) \cap R^-_G(W)$, i.e., for every directed (posiibly closed) path P from a vertex in W to a vertex in W all the intermediate vertices of P also lie in W. The concept of convexity plays an essential rôle in our result, which replaces the rôle of reachability from roots in KKT theorem [3]. It should be noted that for any convex set U in G and the vertex set W of any strongly connected component of G that satisfy $U \cap W \neq \emptyset$, we must have $U \supseteq W$.

Suppose that we are given a finite index set I and, for each $i \in I$, a specified vertex $r_i \in V$. Here we may allow $r_i = r_j$ for some distinct $i, j \in I$. For each $i \in I$ we are also given a convex set $U_i \subseteq V$ such that $r_i \in U_i$. For any $v \in V$ define

$$I(v) = \{i \in I \mid v \in U_i\}.$$

We assume that $I(v) \neq \emptyset$ for all $v \in V$.

Now we are ready to state our main theorem, which is an extension of KKT theorem. It should be noted that replacing U_i by $R^+_G(r_i)$ for all $i \in I$ in our theorem yields KKT theorem. Our proof employs KKT theorem recursively.

Theorem: *The following four statements are equivalent.*

(a) *There exist arc-disjoint arborescences $H_i = (U_i, B_i)$ $(i \in I)$ such that for each $i \in I$ arborescence H_i has a root r_i.*

(b) *for each $v \in V$ there exist arc-disjoint directed paths P_i $(i \in I(v))$ such that for each $i \in I(v)$ path P_i is from r_i to v.*

(c) *For any vertex subset $Z \subset V$*

$$|\Delta^- Z| \geq |\{i \in I(Z) \mid r_i \notin Z\}|,$$

where $\Delta^- Z$ denotes the set of arcs $a = (u, v)$ such that $u \notin Z$ and $v \in Z$.

(d) *There exist spanning trees $T_i = (U_i, E_i)$ of $G[U_i]$ $(i \in I)$ such that E_i $(i \in I)$ are pairwise disjoint and every vertex $v \in V$ has in-degree equal to $|I'(v)|$ in the union of T_i $(i \in I)$ (as a subgraph $H = (V, \cup_{i \in I} E_i)$ of G), where we define $I'(v) = \{i \in I(v) \mid r_i \neq v\}$ for all $v \in V$.*

Because of (d) a weighted version of the problem can also be solved in polynomial time.

References

[1] K. Bérczi and A. Frank, *Variations for Lovász' submodular ideas*, Technical Report TR-2008-7, The Egerváry Research Group on Combinatorial Optimization (2008).

[2] J. Edmonds, *Edge-disjoint branchings*, in *Combinatorial Algorithms* (B. Rustin, ed.), Academic Press (1973), 91–96.

[3] N. Kamiyama, N. Katoh, and A. Takizawa, *Arc-disjoint in-trees in directed graphs*, Proceedings of the 19th annual ACM-SIAM symposium on Discrete Algorithms (2008), 518–526; also to appear in Combinatorica.

[4] A. Schrijver, *Combinatorial Optimization—Polyhedra and Efficiency*, Springer (2003).

Semidefinite Programming Bounds for Stable Sets and Coloring

Monique Laurent

Given a graph $G = (V, E)$, a set $S \subseteq V$ is *stable* if it contains no edge and a *coloring* is an assignment of colors to the nodes so that adjacent nodes receive distinct colors, in other words a partition of V into stable sets. The maximum cardinality of a stable set is denoted $\alpha(G)$, the *stability number* of G, and the minimum number of colors in a proper coloring as $\chi(G)$, its *coloring (chromatic) number*. As is well known, $\alpha(G)$ and $\chi(G)$ are hard to compute and to approximate. Thus it is of interest to find good bounds for $\alpha(G)$ and $\chi(G)$ that can be computed in polynomial time. The celebrated *theta number*:

$$\vartheta(G) = \max \langle J, X \rangle \text{ s.t. } \mathrm{Tr}(X) = 1,\ X_{ij} = 0\ (ij \in E), X \succeq 0, \tag{1}$$

was introduced by Lovász [14], as upper bound on the Shannon capacity of G. Here $X \succeq 0$ means that X is a symmetric positive semidefinite matrix. As it

is defined via a semidefinite program it can be computed in polynomial time (to any precision). The following 'sandwich inequalities' show that the theta number approximates both $\alpha(G)$ and $\chi(G)$, as well as the fractional chromatic number $\chi_f(G)$:

$$\alpha(G) \leq \vartheta(G) \leq \chi_f(\bar{G}) \leq \chi(\bar{G}), \tag{2}$$

where $\bar{G}$ is the complementary graph of G; the fractional chromatic number of G being defined as

$$\chi_f(G) := \min \sum_S \lambda_S \ \text{ s.t. } \ \sum_S \lambda_S \mathbf{1}_S = \mathbf{1}_V, \ \lambda_S \geq 0$$

where the summation is over all stable sets and $\mathbf{1}_S$ is the incidence vector of $S \subseteq V$.

When $\alpha(G) = \chi(\bar{G})$, there is equality throughout in (2), both parameters are equal to $\vartheta(G)$ and thus can be computed in polynomial time. Therefore, the stability number and the chromatic number can be computed in polynomial time for the class of perfect graphs (see e.g. [1]). This is the first application of semidefinite programming to combinatorial optimization and, as of today, this is still the only known efficient algorithm for this problem.

A natural question is how to strengthen the theta number toward $\alpha(G)$ or $\chi(G)$. Several methods have been proposed for designing tractable LP or SDP relaxations of 0/1 LP problems, thus in particular for the stable set problem; in particular the matrix-cut method of Lovász-Schrijver [15] and the method of Lasserre [10] based on moment matrices (see e.g. [11, 13] for a detailed survey). Both give a hierarchy of SDP relaxations finding $\alpha(G)$ in $\alpha(G)$ steps. The Lasserre construction gives a finer hierarchy, however at a high computational cost, since it involves matrices of order $O(n^{2t})$ at the relaxation order t. As shown in [3] this hierarchy can easily be adapted to give a hierarchy of lower bounds for $\chi_f(G)$, converging to it in $\alpha(G)$ steps. This raises the following questions:

- How to get bounds for the chromatic number that go beyond the fractional chromatic number?
- How to define bounds via compact SDP's, thus more amenable to practical computation for interesting graphs?

To answer the first question we propose to use the well known reduction of the chromatic number to the stability number:

$$\chi(G) = \min \ t \ \text{ s.t. } \ \alpha(G_t) = |V(G)|,$$

where G_t is the Cartesian product of G with the complete graph K_t. Now if β is a graph parameter nested between $|V|/\chi$ and $\bar{\chi}$, then the graph parameter Ψ_β defined by

$$\Psi_\beta(G) := \min \ t \ \text{ s.t. } \ \beta(G_t) = |V(G)|$$

is nested between the clique number ω and the chromatic number χ. Obviously, if β is polynomial-time computable then the same holds for Ψ_β. Here are some properties of the operator Ψ: Ψ is monotone non-increasing; $\Psi_\alpha = \chi$, $\Psi_\vartheta = \lceil \bar{\vartheta} \rceil$; Ψ maps a hierarchy converging to α to a hierarchy converging to χ; Ψ maps the

whole interval $[|V|/\omega, \bar{\chi}]$ to ω. As a direct application, a graph parameter in this interval cannot be computed in polynomial time if P$\neq$NP.

To address the second question above, we propose to exploit symmetry. The basic idea being that if we have a SDP with symmetry structure, we can restrict its solutions to the invariant ones (which permits to decrease the number of variables), and we can replace the SDP by an equivalent one with a block-diagonal structure (thus easier to solve). (See e.g. [6, 17] for a survey.) Symmetry can be present in the graph G, which is the case e.g. for Hamming or Kneser graphs, which have a large automorphism group. Another possible source of symmetry is to design the hierarchy in such a way that it involves matrices with a specific symmetry structure. This is precisely what is done in [5] where we design a new, block-diagonal hierarchy which, while being weaker than the Lasserre hierarchy, still refines the hierarchy of Lovász-Schrijver, however at a lower computational cost. While the order 1 relaxation gives back the theta number, the order 2 relaxation gives an improved bound for $\alpha(G)$, which has been succesfully tested in particular in [16, 8, 12] for Hamming graphs. We also test the order 3 bound for Paley graphs in [5]. We have also tested these bounds for the chromatic number in [2, 4] for Hamming and Kneser graphs.

We also propose new compact SDP bounds for the chromatic number of a graph with no apparent symmetry, that improve on several instances of the DIMACS instances (the random DSJC graphs) the best known lower bounds and sometimes permit to determine the exact value of the chromatic number.

We also discuss the link to the hierarchies of SDP bounds based on copositive coprogramming for the stability number (in [7]) and for the fractional chromatic number (in [9]).

This lecture is based on joint work with Nebojša Gvozdenović.

References

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

[2] N. Gvozdenović. *Approximating the Stability Number and the Chromatic Number of a Graph via Semidefinite Programming.* Ph.D. thesis, Univ. Amsterdam (2008).

[3] N. Gvozdenović and M. Laurent, *The operator Ψ for the Chromatic Number of a Graph*, SIAM Journal on Optimization **19(2)** (2008), 572–591.

[4] N. Gvozdenović and M. Laurent, *Computing semidefinite programming lower bounds for the (fractional) chromatic number via block-diagonalization*, SIAM Journal on Optimization **19(2)** (2008), 592–615.

[5] N. Gvozdenović, M. Laurent, and F. Vallentin, *Block-diagonal semidefinite programming hierarchies for 0/1 programming*, Operations Research Letters, to appear.

[6] E. de Klerk, *Exploiting structure in semidefinite programming: A survey of theory and applications*, Optimization Online, October 2008.

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

[8] E. de Klerk and D.V. Pasechnik, *A note on the stability number of an orthogonality graph*, European Journal of Combinatorics **28** (2007), 1971–1979.

[9] I. Dukanovic and F. Rendl, *Copositive programming motivated bounds on the clique and the chromatic number*, Mathematical Programming, Online publication, 4 June 2008.

[10] J.B. Lasserre, *An explicit exact SDP relaxation for nonlinear* $0-1$ *programs*, in K. Aardal and A.M.H. Gerards, eds., LNCS **2081** (2001), 293–303.
[11] M. Laurent, *A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming*, Mathematics of Operations Research **28(3)** (2003), 470–496.
[12] M. Laurent, *Strengthened semidefinite programming bounds for codes*, Mathematical Programming **109(2-3)** (2007), 239–261.
[13] M. Laurent and F. Rendl, *Semidefinite Programming and Integer Programming*, in *Handbook on Discrete Optimization*, K. Aardal, G. Nemhauser, R. Weismantel (eds.), Elsevier (2005), 393–514.
[14] L. Lovász, *On the Shannon capacity of a graph*, IEEE Transactions on Information Theory **IT-25** (1979), 1–7.
[15] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* $0-1$ *optimization*, SIAM Journal on Optimization **1** (1991), 166–190.
[16] A. Schrijver, *New code upper bounds from the Terwilliger algebra and semidefinite programming*, IEEE Transactions on Information Theory **51** (2005), 2859–2866.
[17] F. Vallentin, *Symmetry in semidefinite programs*, Optimization Online, June 2007.

Faster Min-Max Resource Sharing and Application

Jens Vygen
(joint work with Dirk Müller)

The problem of sharing a set of limited resources between users (customers) in an optimal way is fundamental. The common mathematical model has been called the min-max resource sharing problem. Well-studied special cases are the fractional packing problem and the maximum concurrent flow problem. The only known exact algorithms for these problems use general linear (or convex) programming. Sharokhi and Matula [10] were the first to design a combinatorial approximation scheme for the maximum concurrent flow problem. Subsequently, this result was improved, simplified, and generalized many times.

This work is a further step on this line. In particular we provide a simple algorithm and a simple proof of the best performance guarantee in significantly smaller running time. Moreover, we implemented the algorithm and show experimental results for an application to global routing of VLSI chips.

The problem. The Min-Max Resource Sharing Problem is defined as follows. Given finite sets $\mathcal{R}$ of *resources* and $\mathcal{C}$ of *customers*, a convex set $\mathcal{B}_c$, called *block*, of feasible solutions for customer c (for $c \in \mathcal{C}$), and a nonnegative continuous convex function $g_c : \mathcal{B}_c \to \mathbb{R}_+^{\mathcal{R}}$ for $c \in \mathcal{C}$ specifying the *resource consumption*, the task is to find $b_c \in \mathcal{B}_c$ $(c \in \mathcal{C})$ approximately attaining

$$\lambda^* := \inf \left\{ \max_{r \in \mathcal{R}} \sum_{c \in \mathcal{C}} (g_c(b_c))_r \,\middle|\, b_c \in \mathcal{B}_c \,(c \in \mathcal{C}) \right\}, \tag{1}$$

i.e., approximately minimizing the largest resource consumption.

We assume that g_c can be computed efficiently and we have a constant $\epsilon_0 \geq 0$ and oracle functions $f_c : \mathbb{R}_+^{\mathcal{R}} \to \mathcal{B}_c$, called *block solvers*, which for $c \in \mathcal{C}$ and $\omega \in \mathbb{R}_+^{\mathcal{R}}$ return an element $b_c \in \mathcal{B}_c$ with $\omega^\top g_c(b_c) \leq (1+\epsilon_0)\mathrm{opt}_c(\omega)$, where $\mathrm{opt}_c(\omega) :=$

$\inf_{b\in\mathcal{B}_c} \omega^\top g_c(b)$. Block solvers are called *strong* if $\epsilon_0 = 0$ or $\epsilon_0 > 0$ can be chosen arbitrarily small, otherwise they are called *weak*.

Note that previous authors often required that $\mathcal{B}_c$ is compact, but we do not need this assumption. Some algorithms require *bounded block solvers*: for $c \in \mathcal{C}$, $\omega \in \mathbb{R}^{\mathcal{R}}_+$, and $\mu > 0$, they return an element $b_c \in \mathcal{B}_c$ with $g_c(b_c) \le \mu\mathbf{1}$ and $\omega^\top g_c(b_c) \le (1+\epsilon_0)\inf\{\omega^\top g_c(b) \mid b \in \mathcal{B}_c, g_c(b) \le \mu\mathbf{1}\}$ (by $\mathbf{1}$ we denote the all-one vector). They can also be strong or weak.

All algorithms that we consider are fully polynomial approximation schemes relative to ϵ_0, i.e., for any given $\epsilon > 0$ they compute a solution $b_c \in \mathcal{B}_c$ ($c \in \mathcal{C}$) with $\max_{r\in\mathcal{R}} \sum_{c\in\mathcal{C}} (g_c(b_c))_r \le (1+\epsilon_0+\epsilon)\lambda^*$, and the running time depends polynomially on ϵ^{-1}. By θ we denote the time for an oracle call (to the block solver). Moreover, we write $\rho := \sup\{\frac{(g_c(b))_r}{\lambda^*} \mid r \in \mathcal{R}, c \in \mathcal{C}, b \in \mathcal{B}_c\}$.

Previous work. Grigoriadis and Khachiyan [4] were the first to present such an algorithm for the general Min-Max Resource Sharing Problem. Their algorithm uses $O(|\mathcal{C}|^2 \log|\mathcal{R}|(\epsilon^{-2} + \log|\mathcal{C}|))$ calls to a strong bounded block solver. They also have a faster randomized version.

In [5] they proposed an algorithm which needs $O(|\mathcal{C}||\mathcal{R}|(\epsilon^{-2}\log\epsilon^{-1} + \log|\mathcal{R}|))$ calls to a strong, but not bounded, block solver. They also showed that $O(|\mathcal{C}|^2 \log|\mathcal{R}|(\epsilon^{-2} + \log|\mathcal{R}|))$ calls to a strong bounded block solver suffice.

Jansen and Zhang [6] generalized this and allowed *weak block solvers*. Their algorithm needs $O(|\mathcal{C}||\mathcal{R}|(\log|\mathcal{R}| + \epsilon^{-2}\log\epsilon^{-1}))$ calls to a block solver.

	block solver	running time
Grigoriadis, Khachiyan [4]	strong, bounded	$\tilde{O}(\epsilon^{-2}\|\mathcal{C}\|^2\theta)$
Grigoriadis, Khachiyan [5]	strong, unbounded	$\tilde{O}(\epsilon^{-2}\|\mathcal{C}\|\|\mathcal{R}\|\theta)$
Jansen, Zhang [6]	weak, unbounded	$\tilde{O}(\epsilon^{-2}\|\mathcal{C}\|\|\mathcal{R}\|\theta)$
our algorithm	weak, unbounded	$\tilde{O}(\epsilon^{-2}\rho\|\mathcal{C}\|\theta)$
our algorithm	weak, bounded	$\tilde{O}(\epsilon^{-2}\|\mathcal{C}\|\theta)$

Table 1. Approximation algorithms for the Min-Max Resource Sharing Problem. Running times are shown for fixed $\epsilon_0 \ge 0$, and logarithmic terms are omitted.

Fractional packing. The special case where the functions g_c ($c \in \mathcal{C}$) are linear is often called the Fractional Packing Problem (although sometimes this name is used for different problems). For this special case faster algorithms using unbounded block solvers are known. Plotkin, Shmoys and Tardos [9] require a strong block solver and $O(\epsilon^{-2}\rho|\mathcal{C}|\theta(\log|\mathcal{R}|+\epsilon^{-1}))$ oracle calls to solve the feasibility version ($\lambda^* = 1$ is known). Young's algorithm [12] needs $O(\epsilon^{-2}\rho|\mathcal{C}|(1+\epsilon_0)^2\theta\ln|\mathcal{R}|)$ calls to a weak block solver. Charikar et al. extended the result of [9] to weak block solvers resulting in $O(\epsilon^{-2}\rho|\mathcal{C}|(1+\epsilon_0)^2\theta\log(\rho(1+\epsilon_0)\epsilon^{-1}))$ oracle calls.

Bienstock and Iyengar [2] managed to reduce the dependence on ϵ from $O(\epsilon^{-2})$ to $O(\epsilon^{-1})$. Their algorithm does not call a block solver, but requires the resource consumption functions to be explicitly specified by a $|\mathcal{R}| \times \dim(\mathcal{B}_c)$-matrix G_c for

each $c \in \mathcal{C}$. So their algorithm does not apply to the general MIN-MAX RESOURCE SHARING PROBLEM, but to an interesting special case which includes the MAXIMUM CONCURRENT FLOW PROBLEM. The algorithm solves $O(\epsilon^{-1}\sqrt{Kn}\log|\mathcal{R}|)$ separable convex quadratic programs, where $n := \sum_{c\in\mathcal{C}} \dim(\mathcal{B}_c)$, and $K := \max_{1\le i\le|\mathcal{R}|} \sum_{c\in\mathcal{C}} k_i^c$, with k_i^c being the number of nonzero entries in the i-th row of G_c.

	block solver	running time
Plotkin, Shmoys, Tardos [9] $*$	strong, unbounded	$\tilde{O}(\epsilon^{-2}\rho\|\mathcal{C}\|\theta)$
Young [12]	weak, unbounded	$\tilde{O}(\epsilon^{-2}\rho\|\mathcal{C}\|\theta)$
Charikar et al. [3] $*$	weak, unbounded	$\tilde{O}(\epsilon^{-2}\rho\|\mathcal{C}\|\theta)$
Bienstock, Iyengar [2]	—	$\tilde{O}(\epsilon^{-1}\sqrt{Kn}T_{QP})$

TABLE 2. Approximation algorithms for the fractional packing problem. Entries with $*$ refer to the feasibility version ($\lambda^* = 1$). Running times are shown for fixed $\epsilon_0 \ge 0$, and logarithmic terms are omitted. T_{QP} is the time for solving a convex separable quadratic program over $\mathcal{B}_{c_1} \times \ldots \times \mathcal{B}_{c_{|\mathcal{C}|}}$.

Our results. We have a new algorithm for the general MIN-MAX RESOURCE SHARING PROBLEM. It uses ideas of Grigoriadis and Khachiyan [5], Young [12], Albrecht [1], and Vygen [11]. The same algorithm and a quite simple analysis yields two results: With a weak unbounded block solver we obtain a running time of $O(|\mathcal{C}|\theta\rho(1+\epsilon_0)^2\log|\mathcal{R}|(\log|\mathcal{R}| + \epsilon^{-2}(1+\epsilon_0)))$. This generalizes several results for the linear case and improves on results for the general case for moderate values of ρ. With a weak bounded block solver the running time is $O(|\mathcal{C}|\theta(1+\epsilon_0)^2\log|\mathcal{R}|(\log|\mathcal{R}| + \epsilon^{-2}(1+\epsilon_0)))$. This improves on previous results by roughly a factor of $|\mathcal{C}|$ or $|\mathcal{R}|$. The running times are summarized in Tables 1 and 2.

Application. Our main motivation to study this problem was an application to VLSI design. In global routing, instances of the (nonlinear) MIN-MAX RESOURCE SHARING PROBLEM occur naturally when dealing with today's constraints and objectives. We show how to model timing constraints, yield, and power consumption. We incorporate a speed-up technique that drastically decreases the number of oracle calls in practice. We generalize the randomized rounding paradigm to our problem and obtain an improved bound.

Our experimental results for instances from current chips show that the algorithm is applicable in practice, with millions of customers and resources. We show that such problems can be solved efficiently, in less than an hour of computing time with eight processors. This generalizes and improves previous work by [1], [11], and [7].

The paper is avaliable as technical report [8] but not published yet.

References

[1] C. Albrecht, *Global routing by new approximation algorithms for multicommodity flow*, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems **20** (2001), 622–632.

[2] D. Bienstock and G. Iyengar, *Concurrent Flows in $O^*(\frac{1}{\epsilon})$ iterations*, Proceedings of the 26th Symposium on Theory of Computing (2004), 146–155.

[3] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin, *Approximating a finite metric by a small number of tree metrics*, Proceedings of the 39th Annual IEEE Symposium on the Foundations of Computer Science (1998), 379–388.

[4] M. D. Grigoriadis and L. D. Khachiyan, *Fast approximation schemes for convex programs with many blocks and coupling constraints*, SIAM Journal on Optimization **4** (1994), 86–107

[5] M. D. Grigoriadis and L. D. Khachiyan, *Coordination complexity of parallel price-directive decomposition*, Mathematics of Operations Research **21** (1996), 321–340.

[6] K. Jansen and H. Zhang, *Approximation algorithms for general packing problems and their application to the multicast congestion problem*, Mathematical Programming A **114** (2008), 183–206.

[7] D. Müller, *Optimizing yield in global routing*, Proceedings of the IEEE International Conference on Computer-Aided Design (2006), 480–486.

[8] D. Müller and J. Vygen, *Faster min-max resource sharing and applications*, Technical Report No. 08987, Research Institute for Discrete Mathematics, University of Bonn, 2008.

[9] S. A. Plotkin, D. B. Shmoys, and É. Tardos, *Fast approximation algorithms for fractional packing and covering problems*, Mathematics of Operations Research **2** (1995), 257–301.

[10] F. Shahrokhi and D. W. Matula, *The maximum concurrent flow problem*, Journal of the ACM **37** (1990), 318–334.

[11] J. Vygen, *Near-optimum global routing with coupling, delay bounds, and power consumption*, in: Integer Programming and Combinatorial Optimization; Proceedings of the 10th International IPCO Conference; LNCS **3064** (G. Nemhauser, D. Bienstock, eds.), Springer (2004), 308–324.

[12] N. E. Young, *Randomized rounding without solving the linear program*, Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms (1995), 170–178.

The Virtual Private Network Design Problem with Concave Costs

Dirk Oliver Theis

(joint work with Samuel Fiorini, Gianpaolo Oriolo, and Laura Sanità)

The symmetric Virtual Private Network Design (VPND) problem is concerned with buying capacity on links (edges) in a communication network such that certain traffic demands can be met. The precise definition is below. It was shown by Fingerhut, Suri and Turner [3] and later, independently, by Gupta, Kleinberg, Kumar, Rastogi and Yener [8] that VPND can be solved in polynomial time if it has the so-called tree routing property, that is, each instance has an optimal solution whose support is a tree. It was conjectured that VPND has the tree routing property, see, e.g., Erlebach and Rüegg [2], Italiano, Leonardi and Oriolo [12] and Hurkens, Keijsper and Stougie [11]. The conjecture was recently solved affirmatively by Goyal, Olver and Shepherd [5] by settling an equivalent conjecture, due to Grandoni, Kaibel, Oriolo and Skutella [7], claiming that another problem called the Pyramidal Routing (PR) problem has the tree routing property. This fact had previously been established only for cycles [7, 11] and outerplanar graphs [4].

In recent work we have investigated a natural generalization of VPND where the cost per unit of capacity may decrease if a larger amount of capacity is reserved (economies of scale principle). The growth of the cost of capacity is modelled by a non-decreasing concave function f. We call the problem the concave symmetric Virtual Private Network Design (cVPND) problem. This problem is APX-hard for general f due to the fact that it contains the minimum Steiner tree problem as a special case.

Our main contributions are as follows. First, as a corollary to Goyal *et al.* [5], we show that a generalization of the PR problem which we call Concave Routing (CR) problem, and hence also cVPND, has the tree routing property. Second, we study approximation algorithms for cVPND. For general f, using known results on the so-called Single Source Buy at Bulk problem by Grandoni and Italiano [6], we give a randomized 24.92-approximation algorithm.

Detailed description of the problems. We now describe the *symmetric Virtual Private Network design* (VPND) problem and its generalization with concave costs, the *concave symmetric Virtual Private Network Design* (cVPND) problem.

The problems have as input a simple, undirected, connected graph $G = (V, E)$ which represents a communication network; a vector $c \in \mathbb{R}_+^E$ describing the edge costs; and a vector $b \in \mathbb{Z}_+^V$ of maximum cumulative demands. A vertex v with $b_v > 0$ is referred to as a *terminal*. We denote the set of terminals by W.

The vertices of G want to communicate with each other. The exact amount of traffic between pairs of vertices is not known. However, for each vertex v the cumulative amount of traffic that v can send or receive is at most its maximum cumulative demand b_v. The aim is to install, at minimum cost, a so-called *virtual private network,* as a "sub-network" of G. A virtual private network consists of a set of paths $\mathcal{P}$ containing exactly one u–v path P_{uv} in G for each unordered pair $\{u, v\}$ of terminals, and a vector $\gamma \in \mathbb{R}_+^E$ describing the capacity to be installed on each edge. The virtual private network should allow to route any feasible matrix of traffic demands, i.e., whenever d_{uv} is given for each unordered pair $\{u, v\}$ of terminals, such that the upper bounds $\sum_{u \in W} d_{uv} \leq b_v$ for the cumulative demands are satisfied, then for each edge e, the amount of traffic which is routed over edge e is at most the installed capacity:

$$\gamma_e \geq \sum_{\{u,v\} \subseteq W : e \in P_{uv}} d_{uv} \qquad \text{for all edges } e \in E.$$

In the VPND problem, the cost of installing capacity γ_e on edge e is $c_e \gamma_e$, while in the cVPND problem, it is $c_e f(\gamma_e)$. In both cases, the cost of a virtual private network is the sum of the installation of capacity on the edges.

We also use what we call the *Concave Routing* (CR) problem, which is a straightforward generalization of the so-called *Pyramidal Routing* (PR) problem defined in [7]. It, too, is concerned with finding a set of paths satisfying an optimality condition. We do not define it here.

Tree routings. A feasible solution to one of the problems described above is a *tree solution* if the support of the capacity vector γ or, equivalently, the union of the paths in $\mathcal{P}$ induces a tree in G. We say that a problem has the *tree routing property* if, for any instance, there is always a tree solution among the optimal solutions.

In the case of the VPND, the tree routing property, the fact that one can restrict the search for a minimum cost virtual private network to those networks with tree support (tree solutions), established in [5] relying on work in [7], is the key to the polynomial time algorithm. In the non-linear case, the tree routing property is important in our design of approximation algorithms. While acknowledging that the tree routing property for cVPND can be proven directly from the tree routing property for VPND, we offer an elegant geometric proof for the tree routing property of the Concave Routing problem, which then implies the tree routing property of cVPND in much the same way as the tree routing property for PR implies that for VPND.

Approximation. Our approximation algorithms for cVPND rely on the tree routing property. We reduce the cVPND to an intermediate problem which might be thought of as an undirected uncapacitated minimum concave-cost single-source flow problem.

In the special case, when the function f has the form $f(x) = \min(\mu x, M)$, this flow problem is known as the Single Source Rent or Buy problem, for which a randomized 2.92-approximation algorithm exists, as was shown by Eisenbrand, Grandoni, Rothvoß, and Schäfer [1], improving on results by Gupta, Kumar, Pàl, and Roughgarden [10], which can be derandomized with the factor deteriorating to 3.28.

For general functions f, the situation is a bit more subtle. Here we use a theorem of Grandoni & Italiano [6] an the analysis of their (randomized) approximation algorithm for the so-called Single Source Buy at Bulk problem. This problem is concerned with minimum-cost installations of cables on edges. The cables can be selected from a list of cable types, which have characteristic capacities and installation costs. After breaking the function f up into a polynomial sized list of cable types satisfying certain scaling properties, we are able to use said theorem in [6], to show that a polynomial time, randomized 24.92-approximation algorithm exists for the afore-mentioned flow problem, and hence also for the cVPND.

References

[1] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer, *Approximating connected facility location problems via random facility sampling and core detouring*, S-T Huang (ed.) *Proceedings of the XIX Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008* (2008), 1174–1183.

[2] T. Erlebach and M. Rüegg, *Optimal Bandwidth Reservation in Hose-Model VPNs with Multi-Path Routing*, in *Proceedings of the Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 4* (2004), 2275–2282.

[3] J.A. Fingerhut, S. Suri, and J.S. Turner, *Designing least-cost nonblocking broadband networks*, Journal of Algorithms **24(2)** (1997), 287–309.

[4] S. Fiorini, G. Oriolo, L. Sanità, and D.O. Theis, *The Virtual Private Network Design Tree Routing Conjecture for Outerplanar Networks*, Technical Report, `arXiv:0711.2623` (2007).
[5] N. Goyal, N. Olver, and B. Shepherd, *The VPN conjecture is true*, to appear in *Proceedings of the ACM Symposium on Theory of Computing* (2008).
[6] F. Grandoni and G. Italiano, *Improved Approximation for Single-Sink Buy-at-Bulk*, International Symposium on Algorithms and Computation (ISAAC) (2006), 111–120.
[7] F. Grandoni, V. Kaibel, G. Oriolo, and M. Skutella, *A short proof of the VPN tree routing conjecture on ring networks*, to appear in *Operation Research Letters*.
[8] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, *Provisioning a virtual private network: A network design problem for multicommodity flow*, in *Proceedings of the 33rd ACM Symposium on Theory of Computing*, ACM Press (2001), 389–398.
[9] A. Gupta and A. Kumar, Personal communication (2002).
[10] A. Gupta, A. Kumar, M. Pàl and T. Roughgarden, *Approximation via Cost Sharing: Simpler and Better Approximation Algorithms for Network Design*, Journal of the ACM **54(3)** (2007), 11.
[11] C. A. J. Hurkens, J. C. M. Keijsper, and L. Stougie, *Virtual private network design: A proof of the tree routing conjecture on ring networks*, SIAM Journal on Discrete Mathematics **21** (2007), 482–503.
[12] G. Italiano, S. Leonardi, and G. Oriolo, *Design of trees in the hose model: the balanced case*, Operation Research Letters **34(6)** (2006), 601–606.
[13] D. R. Karger and M. Minkoff, *Building Steiner trees with incomplete global knowledge*, Proceedings of the Forty-First IEEE Symposium on Foundations of Computer Science (FOCS) (2006), 613–623.
[14] G. Robins and A. Zelikovsky, *Tighter bounds for graph Steiner tree approximation*, SIAM Journal on Discrete Mathematics **19(1)** (2005), 122–134.

Decorous Lower Bounds for Minimum Linear Arrangement

Alberto Caprara

Given a graph $G = (V, E)$, with $V = \{1, \ldots, n\}$, an *arrangement* (also called a *permutation*, *labelling*, *ordering* or *layout*) is a one-to-one function $\psi : V \to V$. If we view ψ as a placing of the vertices on points $1, \ldots, n$ along the real line, the quantity $|\psi(i) - \psi(j)|$ corresponds to the Euclidean distance between vertices i and j. Several important combinatorial optimization problems, collectively known as *graph layout problems*, call for an arrangement minimizing a function of these distances (see the survey [4]). Here, we are concerned with the *Minimum Linear Arrangement Problem* (MinLA for short, using a common acronym in the literature), in which the objective is to minimize the sum of the pairwise distances between all vertices joined by an edge, namely $\sum_{\{i,j\} \in E} |\psi(i) - \psi(j)|$.

MinLA was originally proposed in [11] back in 1964. It was proven to be strongly NP-hard in [10], and this was later shown to hold even when G is bipartite [9]. For general graphs, the exact algorithm with the best worst-case running time is based on dynamic programming and runs in $O(2^n|E|)$ time [12]. However, MinLA is known to be solvable in polynomial time on trees [19], outerplanar graphs [8] and certain Halin graphs [5].

On the theoretical side, some recent progress has been made on the approximability of MinLA. Approximation algorithms with performance guarantee $O(\log n)$ were introduced in [1, 16]. Very recently, an $O(\sqrt{\log n} \log \log n)$ approximation

algorithm was found [2, 7]. It has been conjectured, however, that MinLA cannot be approximated to within a constant factor in polynomial time [3].

On the practical side, the problem appears to be extremely challenging. On the one hand, several heuristics have been proposed for the problem [12, 14, 15, 17, 18] and tested on a well-established collection of benchmark instances [13]. On the other, it was so far impossible to certify that their solutions are close to optimal since, with the exception of three instances whose optimum is known by construction, the best lower bounds on the optimum are generally some *orders of magnitude* smaller than the heuristic values. This situation is illustrated in [15], which, among other things, provides a clear picture of the situation concerning the practical solvability of MinLA.

Considering that, at present, the best practical method to solve MinLA to proven optimality appears to be (by far) the one based on dynamic programming mentioned above (which rules out the possibility to solve instances with more than, say, 30 nodes), in this paper we focus our attention on the computation of lower bounds for the problem. Our main final contribution is to show that the best-known heuristic solutions are indeed not far from optimal for the benchmark instances with up to 5,000 edges.

Our approach is based on the solution of a suitable *Linear Program* (LP), which involves variables that represent distances between vertices in the layout and contains exponentially many constraints, handled through separation. In itself, this is a very natural idea that has already been exploited in the literature. However, the key to our approach is, on the one hand, to limit to $|E|$ the number of variables, and, on the other, to work with "strong" constraints that arise from the projection into our variable space of natural inequalities in the space of dimension $\binom{n}{2}$ associated with the distances for all vertex pairs. This idea is implicit in [6], in which such an LP was introduced uniquely for theoretical purposes. In this paper, we significantly extend and explore computationally this idea. In doing so, we analyze the structure of the underlying polyhedron, deriving several classes of valid inequalities, proving that they are facet defining, and discussing the associated separation problems.

References

[1] C. Bornstein and S. Vempala, *Flow metrics*, Theor. Comp. Sci. **321** (2004), 13–24.

[2] M. Charikar, M.T. Hajiaghayi, H. Karloff, and S. Rao, *l_2^2 spreading metrics for vertex ordering problems*, in *Proc. of the 17th annual ACM-SIAM Symposium on Discrete Algorithms* (2006), 1018–1027. (Journal version submitted to Algorithmica.)

[3] N.R. Devanur, S.A. Khot, R. Saket, and N.K. Vishnoi, *On the hardness of minimum linear arrangement*, Working paper, College of Computing, Georgia Tech (2006).

[4] J. Díaz, J. Petit, and M. Serna, *A survey of graph layout problems*, ACM Computing surveys **34** (2002), 313–356.

[5] T. Easton, S.B. Horton, and R.G. Parker, *A solvable case of the optimal linear arrangement problem on Halin graphs*, Congr. Numer. **119** (1996), 3–17.

[6] G. Even, J. Naor, S. Rao, and B. Schieber, *Divide-and-conquer approximation algorithms via spreading metrics*, J. of the ACM **47** (2000), 585–616.

[7] U. Feige and J.R. Lee, *An improved approximation ratio for the minimum linear arrangement problem*, Inf. Proc. Lett. **101** (2007), 26–29.
[8] G.N. Frederickson and S.E. Hambrush, *Planar linear arrangements of outerplanar graphs*, IEEE Trans. on Circuits and Systems **35** (1988), 323–332.
[9] M.R. Garey and D.S. Johnson, *Computers and Intractability: An Introduction to the Theory of NP-Completeness*, Freeman (1979).
[10] M.R. Garey, D.S. Johnson, and L.J. Stockmeyer, *Some simplified NP-complete graph problems*, Theor. Comp. Sci. **1** (1976), 237–267.
[11] L.H. Harper, *Optimal assignment of numbers to vertices*, SIAM Journal **12** (1964), 131–135.
[12] Y. Koren and D. Harel, *A multi-scale algorithm for the linear arrangement problem*, in L. Kucera (ed.) *Proc. of 28th Inter. Workshop on Graph-Theoretic Concepts in Computer Science*, LNCS **2573**, Springer (2002), 293–306.
[13] J. Petit, *Layout Problems*. Doctoral thesis, Department of Languages and Information Systems, University Polytechnic of Cataluña (2001).
[14] J. Petit, *Combining spectral sequencing and parallel simulated annealing for the MinLA problem*, Parallel Processing Letters **13** (2003), 77–91.
[15] J. Petit, *Experiments on the linear arrangement problem*, Journal of Experimental Algorithmics **8**, article 2.3 (2003).
[16] S. Rao and A.W. Richa, *New approximation techniques for some linear ordering problems*, SIAM J. on Computing **34** (2005), 388–404.
[17] E. Rodriguez-Tello, J.-K. Hao, and J. Torres-Jimenez, *An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem*, Comp. & Oper. Res. (2007), to appear.
[18] I. Safro, D. Ron, and A. Brandt, *Graph minimum linear arrangement by multilevel weighted edge contractions*, J. of Alg. **60** (2006), 24–41.
[19] Y. Shiloach, *A minimum linear arrangement algorithm for undirected trees*, SIAM J. on Computing **8** (1979), 15–32.

Crossing Families and Pairings

Tamás Király

Let V be a ground set of even cardinality. Two sets $X \subseteq V$ and $Y \subseteq V$ are said to be *crossing* if the sets $X-Y, Y-X, X \cap Y, V-(X \cup Y)$ are all non-empty. A family $\mathcal{F}$ of subsets of V is called *crossing* if the intersection and union of any two crossing members of $\mathcal{F}$ are also in $\mathcal{F}$.

A *pairing* M of V is a set of unordered pairs of elements of V so that every element appears in exactly one pair. For a pairing M of V and $X \subseteq V$ let $d_M(X)$ denote the number of pairs in M with exactly 1 element in X. The problem addressed in this talk is to characterize families $\mathcal{F}$ of subsets of V for which there exists a pairing M such that $d_M(X) = 1$ for every $X \in \mathcal{F}$ (such a pairing is called a *feasible pairing*).

A set is called even (odd) if its cardinality is even (odd). If there is a feasible pairing for $\mathcal{F}$ then obviously every set in $\mathcal{F}$ is odd. We say that a family $\mathcal{F}$ of odd sets contains a *bad configuration* if it contains four sets with the following properties:

- The intersection of any 3 sets is empty,
- The union of any 3 sets is V,
- The intersection of any 2 sets is odd.

It is easy to see that if $\mathcal{F}$ contains a bad configuration, then there is no feasible pairing. Our main result is that if $\mathcal{F}$ is a crossing family then this is the only possible obstacle for the existence of a feasible pairing.

Theorem 1. *Let $\mathcal{F}$ be a crossing family of odd subsets of V that contains no bad configuration. Then V has a pairing M such that $d_M(X) = 1$ for every $X \in \mathcal{F}$.*

We use this theorem to derive a result that is related to Woodall's conjecture on directed cut covers. Let $D = (V, E)$ be a directed graph. An edge set $F \subseteq E$ is a *directed cut cover* if it contains at least one edge from every directed cut of D; it is a *directed cut k-cover* if it contains at least k edges from every directed cut.

Woodall [2, 3] conjectured that the maximum number of edge-disjoint directed cut covers in a digraph equals the minimum size of a directed cut. This conjecture is wide open, there are only a few classes of graphs for which it is known to hold (source-sink connected graphs, series-parallel graphs, transitive closure of directed trees).

A possible generalization would be that a directed cut k-cover always contains k edge-disjoint directed cut covers. However, Schrijver [1] showed that this is not true in general. His counterexample also shows that there is a directed graph $D = (V, E)$ and a directed cut 2-cover $F \subseteq E$ such that for any $F' \subseteq F$, there is a directed cut that contains at most 3 edges from F and is disjoint from either F' or $F - F'$.

We show using the pairing theorem that a counterexample with the latter property cannot exist for higher values of k.

Theorem 2. *Let $D = (V, E)$ a directed graph and let $F \subseteq E$ be a directed cut $2k$-cover for some $k \geq 2$. Then there is an edge set $F' \subseteq F$ such that both F' and $F - F'$ contain at least k edges from any directed cut of size at most $2k + 1$.*

References

[1] A. Schrijver, *A counterexample to a conjecture of Edmonds and Giles*, Discrete Mathematics **32** (1980), 213–214.

[2] D.R. Woodall, *Menger and Kőnig systems*, in: Y. Alavi and D.R. Lick, eds, Theory and Applications of Graphs, LNCS **642**, Springer (1978), 620–635.

[3] D.R. Woodall, *Minimax theorems in graph theory*, in: L.W. Beineke and R.J. Wilson, eds, Selected Topics in Graph Theory, Academic Press (1978), 237–269.

On Lattice Reformulations of Mixed-Integer Programs

Karen Aardal

(joint work with Laurence A. Wolsey)

We revisit the lattice reformulation suggested by Aardal, Hurkens and Lenstra [1]. In particular we explain the distinction between the resulting reformulation when applied to equality and inequality constrained integer programs. We also show that this approach extends naturally to MIPs. We illustrate the two steps involved in the reformulation on some small examples.

References

[1] K. Aardal, C.A.J. Hurkens, and A.K. Lenstra, *Solving a system of diophantine equations with lower and upper bounds on the variables*, Mathematics of Operations Research **25** (2000), 427–442.

[2] K. Aardal and A.K. Lenstra, *Hard equality constrained integer knapsacks*, Mathematics of Operations Research **29(3)** (2004), 724–738. Erratum: Mathematics of Operations Research **31(4)** (2006), 846.

[3] B. Krishnamoorthy and G. Pataki, *Column Basis Reduction and Decomposable Knapsack Problems*, `http://www.optimization-online.org/DB_HTML/2007/06/1701.html`

[4] A.K. Lenstra, H.W. Lenstra Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen **261** (1982), 515–534.

[5] H.W. Lenstra Jr., *Integer programming with a fixed number of variables*, Mathematics of Operations Research **8** (1983), 538–548.

[6] L. Lovász, and H.E. Scarf, *The generalized basis reduction algorithm*, Mathematics of Operations Research **17** (1992), 751–764.

Path Partitions, Cycle Covers, and Integer Decomposition of Coflows

András Sebő

A polyhedron P has the *integer decomposition property*, if every integer vector in kP is the sum of k integer vectors in P. We explain that the projections of polyhedra defined by totally unimodular constraint matrices have the integer decomposition property, in order to deduce the same property for coflow polyhedra defined by Cameron and Edmonds. We then apply this result to the convex hull of particular stable sets in graphs. Therebye we prove a generalization of Greene and Kleitman's well-known theorem on posets to arbitrary digraphs which implies recent and classical purely graph theoretical results on cycle covers, is closely related to conjectures of Berge and Linial on path partitions, and implies these for some extremal values of the parameters.

Partitioning the vertex-set of a graph by a minimum number of paths is one of the most natural problems concerning graphs. Minimizing the number of paths in such a partition contains the Hamiltonian Path problems both in the directed and undirected case.

For digraphs, a classical theorem of Gallai and Milgram *relates the problem to the stability number of a graph.* Path partitions in digraphs have been treated both with elegant graph theory, network flows, and polyhedral combinatorics, but have not yet revealed all of their secrets:

Conjectures of Berge [1] and Linial [16] about the relation of maximum sets of vertices inducing a k-chromatic subgraph and particular path partitions resist through the decades. Hartman's excellent survey [12] witnesses of the variety of the methods that have been tried out with a lot of partial results but no breakthrough as far as the general conjectures are concerned. Some other conjectures are less well-known or have not yet been stated.

Led by analogies, we ask and answer in this talk more questions, and point at some connections.

We first state analogous pairs of theorems on path partitions and cycle covers.

Then we present the results concerning cycle covers deducing all from a general theorem proved with the help of the property of a corresponding polyhedron: the integer decomposition property.

Finally we present some results on path partitions, and some connections of these to cycle covers.

References

[1] C. Berge, *k-optimal partitions of a directded graph*, European J. of Combinatorics **3** (1982), 97–101.

[2] S. Bessy and S. Thomassé, *Spanning a strong digraph by α circuits: a proof of Gallai's conjecture*, Combinatorica **27** (2007), 659–667.

[3] K. Cameron, *Polyhedral and algorithmic ramifications of antichains*, Ph. D. Thesis, University of Waterloo, Waterloo, Canada (1982).

[4] K. Cameron, *On k-optimum dipath partitions and partial k-colorings of acyclic digraphs*, Europ. J. Combinatorics **7**, 115–118.

[5] K. Cameron and J. Edmonds, *Coflow Polyhedra*, Discrete Mathematics **101** (1992), 1–21.

[6] P. Charbit and A. Sebő, *Cyclic Orders: Equivalence, and Duality*, Combinatorica **28(2)** (2008), 131–143.

[7] A. Frank, *On chain and antichain families of a partially ordered set*, J. of Comb. Theory, Series B **29** (1980), 251–261.

[8] T. Gallai, Problem 15 in M. Fiedler ed, *Theory of Graphs and its Applications*, (Czech Acad. Sci. Prague, 1964), 161.

[9] T. Gallai and A. N. Milgram, *Verallgemeinerung eines graphentheoretischen Satzes von Rédei*, Acta Sc. Math. **21** (1960), 181–186.

[10] T. Gallai, *On directed paths and circuits*, in: *Theory of Graphs*, (P. Erdős and G. Katona eds.), Academic Press (1968), 115–118.

[11] C. Greene and D.J. Kleitman, *The structure of Sperner k-families*, Journal of Combinatorial Theory, Series A **20** (1976), 41–68.

[12] I. B.-A. Hartman, *Berge's Conjecture on Directed Path Partitions - A Survey*, volume in honor of Claude Berge, Discrete Mathematics **306** (2006), 2582–2592.

[13] S. Iwata and T. Matsuda, *Finding coherent cyclic orders in strong digraphs*, Combinatorica **28(1)** (2008), 83–88.

[14] D. E. Knuth, *Wheels within Wheels*, Journal of Combinatorial Theory B **16** (1974), 42–46.

[15] J.M. Laborde, C. Payan, and N. H. Xuong, *Independent sets and longest directed paths in digraphs*, in *Graphs and other combinatorial topics (Prague 1982)*, 173–177, Teubner (1983).

[16] N. Linial, *Extending the Greene-Kleitman theorem to directed graphs*, J. of Combinatorial Theory, Ser. A **30** (1981), 331–334.

[17] L. Rédei, *Ein Kombinatorischer Satz*, Acta Litt. Sci. Szeged **7** (1934), 39–43.

[18] B. Roy, *Nombre chromatique et plus longs chemins*, Rev. F1, Automat. Informat. **1** (1976), 127–132.

[19] M. Saks, *A short proof of the k-saturated partitions*, Adv. In Math. **33** (1979), 207–211.

[20] A. Sebő, *Minmax Theorems in Cyclically Ordered graphs*, Journal of Combinatorial Theory B **97(4)** (2007), 518–552.

[21] A. Sebő, *Path Partitions, Cycle Covers and Integer Decomposition*, to appear in LNCS, at the occasion of the 60th birthday of Martin Charles Golumbic.

[22] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley (1986).

Relaxations for Nonlinear 0–1 Optimization

Christoph Buchheim
(joint work with Giovanni Rinaldi, Angelika Wiegele, and Lanbo Zheng)

Nonlinear binary optimization problems are considerably harder to solve than the corresponding linear problems, both theoretically and in practice. In particular, the problem of maximizing a quadratic objective function over binary variables is NP-hard even in the unconstrained case. In fact, it is equivalent to the maximum cut problem, as shown by De Simone [5]. On the other hand, the latter problem has been well-studied and many results have been obtained for the corresponding *cut polytopes*, see e.g. [4]. These polyhedral results have led to practically fast algorithms for max-cut and quadratic 0–1 optimization based on ILP or SDP techniques [6, 7].

Our aim is to extend the connection between max-cut and binary quadratic optimization to more general non-linear problems: firstly, we are interested in unconstrained problems where the quadratic function is replaced by an arbitrary pseudo-boolean objective function [1, 2]. Secondly, we are interested in quadratic problems subject to linear constraints. In particular, we derive polyhedral results for a typical example of such a problem, the quadratic linear ordering problem [3].

Pseudo-boolean Optimization. We first address the problem of maximizing a non-linear objective function over binary variables. Here we assume that all non-linear expressions in the objective function are given by a recursive construction using arbitrary boolean operators $\{0,1\}^2 \to \{0,1\}$, i.e., we allow expressions such as, e.g., $((a \Rightarrow ((b \wedge c) \oplus d)) \Leftrightarrow (b \vee e))$. Our aim is to maximize an arbitrary linear combination of such expressions over binary variables (without constraints).

We obtain an IP formulation of this problem by a straightforward linearization: we introduce new variables for all non-linear terms in the objective function and recursively for all partial non-linear terms. In our example, the latter variables correspond to the terms $(a \Rightarrow ((b \wedge c) \oplus d))$, $(b \vee e)$, $((b \wedge c) \oplus d)$, and $(b \wedge c)$. Now let P be the polytope spanned by all feasible solutions in this model.

By the result of De Simone mentioned above, the problem of maximizing a quadratic polynomial over binary variables can be reduced to solving an instance of the max-cut problem; moreover, each boolean quadric polytope is isomorphic to a cut polytope. We generalize this result as follows:

Lemma 1. *If all non-linear terms in the objective function contain at most one operator, then the polytope P is isomorphic to a cut polytope defined on the same number of variables.*

In general, if the nesting depth of the objective function is not bounded by one, Lemma 1 does not hold. However, we can show the following:

Theorem 2. *Assume that no operator in the objective function is an exclusive disjunction or an equivalence. Then the polytope P is isomorphic to a face of a cut polytope, where the latter is defined on at most four times as many variables.*

In particular, this result holds in the case of a polynomial objective function, where all operators are multiplications [1]. Theorem 2 allows to reduce the general separation problem for P to the separation problem for a cut polytope in a very efficient way. Experimental results for polynomial 0–1 optimization and for the maximum satisfiability problem show that a branch-and-cut algorithm based on this approach is very powerful in practice. In many cases, it even outperforms problem-specific methods. This is due to a combination of a sparse construction and strong polyhedral results.

Quadratic Linear Ordering. A different type of generalization arises when sticking to a quadratic objective function but adding linear constraints. As a typical example, we consider the quadratic linear ordering problem, which has applications in, e.g., computational biology and bipartite crossing minimization. As in most such cases, the standard linearization of the objective function leads to a very weak relaxation. We show that a quadratic reformulation of the linear constraints leads to a strong improvement of the relaxation.

Let $\mathrm{QLO}(n)$ be the polytope spanned by all feasible solutions of the linearized problem, these solutions correspond to vectors of the form (x, y), where x is a feasible solution for the classical linear ordering model and y_{ijkl} linearizes the product $x_{ij}x_{kl}$. In the following, let P_n denote the polytope for the corresponding linearized unconstrained quadratic optimization problem over the linear ordering variables x, which is isomorphic to a cut polytope [5].

Lemma 3. *Let $(x, y) \in P_n$ be integer. Then $(x, y) \in \mathrm{QLO}(n)$ if and only if*

$$x_{ik} - y_{ijik} - y_{ikjk} + y_{ijjk} = 0$$

for all $i < j < k$. These equations form a minimal equation system for $\mathrm{QLO}(n)$.

The constraints in Lemma 3 are quadratic reformulations of 3-dicycle inequalities; their strength follows from the following statement.

Theorem 4. *For all $i < j < k$, the constraint $x_{ik} - y_{ijik} - y_{ikjk} + y_{ijjk} = 0$ induces a face of the polytope P_n. In particular, $\mathrm{QLO}(n)$ is a face of P_n.*

Again, this result can be exploited practically in a cutting plane approach, as it allows to reduce the separation problem for $\mathrm{QLO}(n)$ to the separation problem for the cut polytope P_n. Moreover, Theorem 4 shows that $\mathrm{QLO}(n)$ is closely related to the unconstrained problem, while any polyhedral knowledge of the linear version of the linear ordering problem is useless for understanding $\mathrm{QLO}(n)$.

References

[1] C. Buchheim and G. Rinaldi, *Efficient reduction of polynomial zero-one optimization to the quadratic case*, SIAM Journal on Optimization **18(4)** (2007), 1398–1413.

[2] C. Buchheim and G. Rinaldi, *Terse integer linear programs for boolean optimization*, Technical Report zaik2007-558, University of Cologne.

[3] C. Buchheim, A. Wiegele, and L. Zheng, *Exact algorithms for the quadratic linear ordering problem*, Technical Report zaik2007-560, University of Cologne.

[4] M. Deza and M. Laurent, *Geometry of Cuts and Metrics*, Springer (1997).

[5] C. De Simone, *The cut polytope and the boolean quadric polytope*, Discrete Mathematics **79** (1990), 71–75.
[6] F. Liers, M. Jünger, G. Reinelt, and G. Rinaldi, *Computing exact ground states of hard Ising spin glass problems by branch-and-cut*, in *New Optimization Algorithms in Physics*, Wiley-VCH (2004), 47–69.
[7] F. Rendl, G. Rinaldi, and A. Wiegele, *A branch and bound algorithm for Max-Cut based on combining semidefinite and polyhedral relaxations*, IPCO 2007, LNCS **4513** (2007), 295–309.

Extended Formulations for Packing- and Partitioning Orbitopes

Volker Kaibel
(joint work with Yuri Faenza)

The *packing orbitope* $O^{\leq}(p,q)$ and the *partitioning orbitope* $O^{=}(p,q)$ are the convex hulls of all 0/1-matrices of size $p \times q$ whose columns are in lexicographically decreasing order having at most or exactly, respectively, one 1-entry per row. In [1], complete descriptions with linear inequalities have been derived for these polytopes. Knowledge on orbitopes turns out to be useful in practical symmetry reduction for certain integer programming models. For instance, in a well-known formulation of the graph partitioning problem (for graphs having p nodes to be partitioned into q parts) the symmetry on the 0/1-variables x_{ij} indicating whether node i is put into part j of the partitioning the symmetry arising from permuting the parts can be removed by requiring $x \in O^{=}(p,q)$.

The topic of the talk are extended formulations for these orbitopes, i.e., (simple) linear descriptions of higher dimensional polytopes which can be projected to $O^{\leq}(p,q)$ and $O^{=}(p,q)$. In fact, such extended formulations play important roles in polyhedral combinatorics and integer programming in general, because rather than solving a linear optimization problem over a polyhedron in the original space, one may solve it over a (hopefully simpler described) polyhedron of which the first one is a linear projection.

The main result of [1] is a complete description of $O^{\leq}(p,q)$ and $O^{=}(p,q)$ by means of linear equations and inequalities. This system of constraints consists, next to nonnegativity constraints and row-sum inequalities (or row-sum equations for $O^{=}(p,q)$), of the exponentially large class of *shifted column inequalities* (*SCI*). In [1] it is also proved that, up to a few exceptions, these exponentially many SCIs define facets of the orbitopes.

In this talk, we describe a quite simple extended formulation for $O^{\leq}(p,q)$ (along with a rather short proof establishing this) and, moreover, we show by some simple and natural (not technical) arguments that the SCI-system describes the projection of the feasible region of that extended formulation to the original space, thus providing a new proof showing that the SCI-system describes $O^{\leq}(p,q)$. The basic idea of our extended formulation is to assign to each vertex of $O^{\leq}(p,q)$ a directed path in a certain acyclic digraph. The additional variables in our extended formulations are used to suitably express these paths. The proof showing that SCI-inequalities basically suffice in order to describe $O^{\leq}(p,q)$ proceeds as follows.

We first construct, for each nonnegative x in the original space (with entries in every row summing up to at most one), a *canonical lifting* by assigning to x a certain flow in the digraph. Then we identify linear inequalities that need to be satisfied by x in order to guarantee that the canonical lifting satisfies the extended formulation. Finally showing that these inequalities (the SCI-inequalities) are valid for $\mathrm{O}^{\leq}(p,q)$, we complete the analysis of the projection. This latter proof is much shorter than the original one, and it seems to provide much better insight into the reasons for the SCI-system to describe the orbitopes. Clearly, as $\mathrm{O}^{=}(p,q)$ is a face of $\mathrm{O}^{\leq}(p,q)$, the results for the latter polytope immediately yield corresponding results for the first one.

Besides leading to that simpler proof our extended formulation for $\mathrm{O}^{\leq}(p,q)$ provides a description of the orbitope by a system of constraints in a space whose dimension is roughly twice the original dimension with only linearly (in that dimension) many nonzero coefficients, while every linear description of the orbitope in the original space requires exponentially many inequalities.

References

[1] V. Kaibel and M. E. Pfetsch, *Packing and Partitioning Orbitopes*, Mathematical Programming A **114** (1) (2008), 1–36.

Reporter: Christoph Buchheim

Participants

Prof. Dr. Karen I. Aardal
Delft Institute of Applied Mathematics
TU Delft
Postbus 5031
NL-2600 GA Delft

Prof. Dr. Robert E. Bixby
Rice University
Jesse H. Jones Graduate School of Mathematics
6100 Main Street
Houston , TX 77005
USA

Dr. Christoph Buchheim
Forschungsinstitut für
Diskrete Mathematik
Universität Bonn
Lennestr. 2
53113 Bonn

Prof. Dr. Rainer E. Burkard
Institut für Mathematik
Technische Universität Graz
Steyrergasse 30
A-8010 Graz

Prof. Dr. Alberto Caprara
Dipartimento di Elettronica, Informatica e Sistemistica (DEIS)
Universita degli Studi di Bologna
Viale Risorgimento, 2
I-40136 Bologna

Prof. Dr. Michelangelo Conforti
Dipartimento di Matematica Pura ed Applicata
Universita di Padova
Via Trieste
I-35131 Padova

Prof. Dr. William J. Cook
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta , GA 30332-0205
USA

Prof. Dr. William H. Cunningham
Department of Combinatorics and Optimization
University of Waterloo
200 University Ave.
Waterloo Ont., N2L 3G1
CANADA

Dr. Sanjeeb Dash
IBM Corporation
Thomas J. Watson Research Center
1101 Kitchawan Rd.
P.O.Box 218
Yorktown Heights , NY 10598
USA

Prof. Dr. Jack Edmonds
Dept. of Combinatorics and Optimization
University of Waterloo
200 University Avenue West
Waterloo , Ont. N2L 3G1
CANADA

Prof. Dr. Friedrich Eisenbrand
EPFL SB IMA
MA-C1-573
Station 8
CH-1015 Lausanne

Prof. Dr. Matteo Fischetti
Dipartimento di Ingegneria dell' Informazione
Universita di Padova
Via Gradenigo 6/A
I-35131 Padova

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

Prof. Dr. Michel Goemans
Department of Mathematics
Massachusetts Institute of
Technology
77 Massachusetts Avenue
Cambridge , MA 02139-4307
USA

Prof. Dr. Bertrand Guenin
Department of Combinatorics
and Optimization
University of Waterloo
200 University Ave.
Waterloo Ont., N2L 3G1
CANADA

Prof. Dr. Satoru Iwata
Research Institute for
Mathematical Sciences
Kyoto University
Kitashirakawa, Sakyo-ku
Kyoto 606-8502
JAPAN

Prof. Dr. Tibor Jordan
Department of Operations Research
Eötvös Lorand University
ELTE TTK
Pazmany Peter setany 1/C
H-1117 Budapest

Prof. Dr. Michael Jünger
Institut für Informatik
Universität zu Köln
Pohligstr. 1
50969 Köln

Dr. Volker Kaibel
Fakultät für Mathematik
Otto-von-Guericke-Universität
Magdeburg
Universitätsplatz 2
39106 Magdeburg

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

Prof. Dr. Dr.h.c. Bernhard Korte
Forschungsinstitut für
Diskrete Mathematik
Universität Bonn
Lennestr. 2
53113 Bonn

Dr. Monique Laurent
CWI - Centrum voor Wiskunde en
Informatica
Postbus 94079
NL-1090 GB Amsterdam

Prof. Dr. Andrea Lodi
Dipartimento di Elettronica, Informatica
e Sistemistica (DEIS)
Universita degli Studi di Bologna
Viale Risorgimento, 2
I-40136 Bologna

Prof. Dr. Alexander Martin
Fachbereich Mathematik
FG Diskrete Optimierung
Technische Universität 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
Fakultät II, Sekr. MA 5-1
Technische Universität Berlin
Straße des 17. Juni 136
10623 Berlin

Prof. Dr. Kazuo Murota
Dept. of Mathematical Informatics
Grad. School of Inform. Sci. & Techn.
University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo 113-8656
JAPAN

Prof. Dr. Petra Mutzel
Fakultät für Informatik
LS 11
Technische Universität Dortmund
Otto-Hahn-Straße 14
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 Computer Science
Cornell University
Upson Hall
Ithaca , NY 14853
USA

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. Andreas S. Schulz
Massachusetts Institute of Technology
Sloan School of Management
E 53 - BS 7
77 Massachusetts Avenue
Cambridge MA 02139-4307
USA

Prof. Dr. Andras Sebö
Institut National Polytechnique de Grenoble
Laboratoire G-SCOP
46, Avenue Felix Viallet
F-38031 Grenoble Cedex 1

Prof. Dr. Frederick Bruce Shepherd
Dept. of Mathematics and Statistics
McGill University
805, Sherbrooke Street West
Montreal , P.Q. H3A 2K6
CANADA

Prof. Dr. Martin Skutella
Institut für Mathematik
Fakultät II, Sekr. MA 5-1
Technische Universität Berlin
Straße des 17. Juni 136
10623 Berlin

Dr. Dirk Oliver Theis
Dept. de Mathematiques
Universite Libre de Bruxelles
CP 216 Campus Plaine
Bd. du Triomphe
B-1050 Bruxelles

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
Cornell University
School of Operations Research and
Information Engineering
Rhodes Hall 236
Ithaca NY 14853
USA

Prof. Dr. Gerhard Woeginger
Department of Mathematics
Eindhoven University of Technology
P.O.Box 513
NL-5600 MB Eindhoven

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 f. Mathematische Optimierung
T.U. Braunschweig
Pockelsstr. 14
38106 Braunschweig