Mathematisches Forschungsinstitut Oberwolfach

Report No. 28/2004

Approximation Algorithms for NP-Hard Problems

Organised by
Ravi Kannan (Yale)
Marek Karpinski (Bonn)
Hans Jürgen Prömel (Berlin)

June 6th–June 12th, 2004

Introduction by the Organisers

It is an interesting artifact that most computational tasks today that arise in realistic scenarios are intractable, at least if one insists on delivering exact solutions with certainty within a strict deadline. An important mean for surmounting this intractability barrier is that of *approximate computation*, where the answer is guaranteed to be within some small fraction of optimality. One of the great recent successes in that area has been the discovery of a new paradigm connecting probabilistic proof verification theory to the theory of approximate computation as well as some new probabilistic combinatorial and algebraic paradigms in designing efficient approximation algorithms.

The workshop was concerned with the most important recent developments in the area of *efficient approximation algorithms* for NP-hard optimization problems as well as with new techniques for proving *intrinsic lower bounds* for efficient approximation.

In addition to 25 lectures delivered at general sessions, there were several additional lectures given at the special sessions and the evening problem session. The Program of the meeting and Abstracts of all talks are listed in the subsequent sections of this report. The special sessions were on the following topics:

- Steiner Tree and Related Optimization Problems.
- Query Efficient PCPs.
- Routing Problems in Distributed Networks.
- New PCP Results.
- Approximating Combinatorial Auctions Without Randomized Rounding.

The meeting was held in very informal and stimulating atmosphere. The *ad hoc* organized special sessions were extraordinarily interesting and intensive venues for communicating most recent results. Thanks to everybody who contributed to the success of this meeting and made it such an enjoyable event!

The organizers and participants thank the *Mathematisches Forschungsinstitut Oberwolfach* for its help in organizing this conference.

Special thanks go to Mathias Hauptmann for his help in editing this proceedings. We are also indebted to Christiane Andrade, Cornelia Kaufmann, Martin Löhnertz, Ignatios Souvatzis and Claus Viehmann for their continuous support in organizing this conference.

August 2004

Ravi Kannan
Marek Karpinski
Hans Jürgen Prömel

Workshop: Approximation Algorithms for NP-Hard Problems

Table of Contents

Special Sessions

Abstracts

Fast and Crude Combinatorial Counting

Alexander Barvinok
(joint work with Alex Samorodnitsky)

We consider the problem of obtaining fast estimates for the cardinality $|X|$ of a family $X \subset 2^{\{1,\dots,n\}}$ of subsets of the set $\{1,\dots,n\}$. Geometrically, we think of X as a subset of the Boolean cube $\{0,1\}^n$. We assume that X is given to us by the *Optimization Oracle*. The oracle accepts a set $\gamma_1,\dots,\gamma_n$ of n real weights as an input and outputs the maximum weight $\max_{x\in X}\sum_{i\in x}\gamma_i$ of a subset $x \in X$. Examples of families X for which such an oracle is readily available through existing optimization algorithms include the family of bases of a matroid or bases in the intersection of two matroids on the common ground set, the family of all perfect matchings in a given graph, etc. One can argue that the existence of the Optimization Oracle is a reasonable assumption: the weaker *Membership Oracle* which just reports whether a given set $x \subset \{1,\dots,n\}$ belongs the the family X is obviously too weak for counting purposes: with the Membership Oracle alone, unless P=NP, we will have hard time deciding whether X is non-empty, for example, in the case when X is the set of all Hamiltonian cycles in a given graph. On the other hand, if we consider a slightly more general problem of counting with multiplicities (each element i of the ground set has a positive integer multiplicity m_i and our goal is to compute $\sum_{x\in X}\prod_{i\in x} m_i$), then efficient counting necessarily implies efficient optimization.

Let us fix a symmetric probability measure μ on $\mathbf{R}$ and let us define $\Gamma(X,\mu) = \mathbf{E}\max_{x\in X}\sum_{i\in x}\gamma_i$, the expected maximum weight of a subset $x \in X$ provided the weights $\gamma_1,\dots,\gamma_n$ are sampled independently at random from the distribution μ. One can easily see that $\Gamma(X,\mu) = 0$ if $|X| = 1$, that $\Gamma(X,\mu) \geq \Gamma(Y,\mu)$ if $Y \subset X$, and that $\Gamma(X\times Y,\mu) = \Gamma(X,\mu) + \Gamma(Y,\mu)$, where $X \subset \{0,1\}^n$, $Y \subset \{0,1\}^m$, and $X \times Y \subset \{0,1\}^{n+m}$. Thus $\Gamma(X,\mu)$ measures how large X is and is somewhat akin to $\ln|X|$. It turns out that in some rigorously defined sense (see below), the best measure μ for which $\Gamma(X,\mu)$ approximates $\ln|X|$ the closest is the *logistic measure* μ_0 with the density $(2+e^{\gamma}+e^{-\gamma})^{-1}$. Denoting $\Gamma(X) = \Gamma(X,\mu_0)$, we have $\ln|X| \leq \Gamma(X)$ with the equality attained if X is a face of the Boolean cube $\{0,1\}^n$. Furthermore, suppose that X is *homogeneous*, that is, all subsets $x \in X$ have the same cardinality k. Then, for any $\alpha > 1$ there is a $\beta = \beta(\alpha) > 0$ such that if $|X| \geq \alpha^k$ then $\beta\Gamma(X) \leq \ln|X|$, and, moreover $\beta \longrightarrow 1$ as $\alpha \longrightarrow +\infty$. In other words, as long as $|X|$ grows exponentially in k, the value of $\Gamma(X)$ approximates $\ln|X|$ within a constant factor and the factor approaches 1 as $k^{-1}\ln|X|$ grows. More precisely, letting $t = k^{-1}\Gamma(X)$, we have $t - \ln t - 1 \leq k^{-1}\ln|X| \leq t$ for all sufficiently large t. Even more precisely, let us define $h(t) = \sup\limits_{0\leq\delta<1}\left(\delta t + \ln\dfrac{\sin\pi\delta}{\pi\delta}\right)$ for $t \geq 0$. Then $h(t)$ is an increasing convex function and $h(t) \leq k^{-1}\ln|X| \leq t$ for all $t \geq 0$. The lower bound is attained asymptotically on random subsets

$X \subset \{0,1\}^n$ while the upper bound, as we noted above, is attained on faces of $\{0,1\}^n$. Note that the bounds are independent on the size n of the ambient set and depend only on the size k of the subsets x from X. This phenomenon is measure-sensitive: roughly, it holds if and only if μ has exponential tail (thus it does not hold, for example, for the Gaussian measure μ). We also note that $h(t) \approx (3/2\pi^2)t^2$ for $t \approx 0$ and that $h(t) \approx t - \ln t - 1$ for large t.

Given the Optimization Oracle for X, it is straightforward to compute $\Gamma(X,\mu)$: we just sample several n-tuples of weights $\gamma_1, \dots, \gamma_n$ and average the outputs. Suppose that $|x| = k$ for all $x \in X$ and that $\mu = \mu_0$ is the logistic measure (the case most interesting for us). Then, to approximate $\Gamma(X) = \Gamma(X,\mu_0)$ within an error $\epsilon > 0$, with an overwhelming probability it is enough to average $O(k\epsilon^{-2})$ outputs. Our numerical experiments indicate that often just one random sample is enough to compute $\Gamma(X)$ with a sufficient accuracy. Again, we note that the number of samples is independent on the size n of the ground set: the same effect is observed for measures μ with at most exponential tails.

Let μ be the Bernoulli measure: $\mu\{1\} = \mu\{-1\} = 1/2$. Then the value of $\Gamma(X,\mu)$ has a very simple geometric meaning: for a subset $X \subset \{0,1\}^n$, the value $\Gamma(X,\mu)$ is equal to $n/2$ minus the average Hamming distance from a point $x \in \{0,1\}^n$ to X. This observation allows us to relate our method to the classical Monte Carlo method: sample a number of random points $x \in \{0,1\}^n$ and estimate $|X|$ by counting how often x lands in X. Indeed, if we choose the trivial distance d in the Boolean cube: $d(x,y) = 0$ if $x = y$ and $d(x,y) = 1$ if $x \neq y$ then the classical Monte Carlo method reduces to estimating the cardinality $|X|$ of a set through the average distance to the set. It works fine if X is almost the whole cube $\{0,1\}^n$, but it works poorly if X constitutes an exponentially small fraction of the cube. Changing d to the Hamming metric allows us to recognize exponentially small subsets $X \subset \{0,1\}^n$. Finally, fine-tuning the Hamming metric we obtain the quantity $\Gamma(X)$ corresponding to the logistic measure μ_0. As we noted before, $\Gamma(X)$ provides an asymptotically sharp approximation of $\ln|X|$ provided X is a family of k-subsets of $\{1,\dots,n\}$ of a sufficiently large size: $|X| \geq \alpha^k$ for a large α. In particular, we should have $n \gg k$. The underlying geometric intuition is as follows: if X is that large, then typical elements $x, y \in X$ are roughly $2k$ Hamming distance apart, so the difference between "sparse" and "dense" sets X disappears: all such sets look roughly the same. Thus the upper bound for $\ln|X|$ in terms of $\Gamma(X)$, achieved for dense sets, and the lower bound, achieved for sparse sets, converge.

Finally, we state if what sense the logistic measure is optimal. Let $\mathcal{M}$ be the class of symmetric measures μ such that $\ln|X| \leq \Gamma(X,\mu)$ for all subsets $X \subset \{0,1\}^n$. Thus, for every $\mu \in \mathcal{M}$, an upper bound on $\Gamma(X,\mu)$ is automatically an upper bound for $\ln|X|$. For a $\mu \in \mathcal{M}$ and $t \geq 0$, let $c(t,\mu) = \inf k^{-1} \ln|X|$, where the infimum is taken over all families X of k-subsets with $k^{-1}\Gamma(X,\mu) \geq t$. Thus $c(t,\mu)$ measures how bad the lower bound for $\ln|X|$ can get, given a lower bound for $\Gamma(X,\mu)$. Then, for all $t \geq 0$ and all $\mu \in \mathcal{M}$, we have $c(t,\mu) \leq c(t,\mu_0)$, where μ_0 is the logistic measure. The definition of $\mathcal{M}$ is quite natural: unless μ is

concentrated in 0, we have $\Gamma(X_0, \mu) = \alpha > 0$, where X_0 is any family of cardinality 2. Then, for direct products $X = X_0 \times \ldots \times X_0$, we have $\Gamma(X, \mu) \geq \alpha \log_2 |X|$, which, after scaling, becomes $\Gamma(x, \mu) \geq \ln |X|$.

A 3/4-Approximation Algorithm for Maximum ATSP with Weights Zero and One

Markus Bläser

Introduction

Traveling salesperson problems with weights one and two have been studied for many years. They are an important special case of traveling salesperson problems with triangle inequality. Papadimitriou and Yannakakis [8] showed that the undirected minimization problem is MaxSNP-complete. On the other hand, they presented a 7/6-approximation algorithm with polynomial running time. Vishwanathan [9] considered the corresponding asymmetric problem ATSP$(1, 2)$ and gave a 17/12-approximation algorithm.

Let MaxATSP$(0, 1)$ be the following problem: Given a directed complete loopless graph with edge weights zero and one, compute a TSP tour of *maximum* weight. MaxATSP$(0, 1)$ is a generalization of ATSP$(1, 2)$ in the following sense: Vishwanathan [9] showed that any $(1-\alpha)$-approximation algorithm for the former problem transforms into an $(1+\alpha)$-algorithm for the latter when replacing weight two with weight zero. (The other direction is not known to be true.)

By computing a matching of maximum weight and patching the edges together arbitrarily, one easily obtains a polynomial time 1/2-approximation algorithm for MaxATSP$(0, 1)$. (Note that each edge has weight at least zero, thus we cannot loose any weight during the patching process.) Vishwanathan [9] was the first to improve on this by designing a 7/12-approximation algorithm with polynomial running time. In 1994, Kosaraju, Park, and Stein [6] gave a 48/63-approximation algorithm with polynomial time that also worked for maximum ATSP with arbitrary nonnegative weights. In their work, they also formulated the so-called path coloring lemma, which will be crucial for our algorithm. Bläser and Siebert [3] obtained a 4/3-approximation algorithm with running time $O(n^{5/2})$ for ATSP$(1, 2)$. This algorithm can also be modified to give a 2/3-approximation algorithm for MaxATSP$(0, 1)$ with the same running time [4]. Finally, Kaplan et al. [5] generalize this result by designing a 2/3-approximation algorithm that works for maximum ATSP with arbitrary nonnegative weights but has a worse running time.

Closely related to MaxATSP$(0, 1)$ is the Directed Path Packing Problem DPP. Here we are given a directed graph $G = (V, E)$. The aim is to find a subset P of node-disjoint paths of G such that the number of edges in P is maximized. By giving edges in G weight one and "non-edges" weight zero, any path packing transforms into a TSP tour by patching the paths arbitrarily together. On the other hand, any TSP tour yields a path packing by discarding all edges of weight

zero. The only exception is the case where an optimum TSP tour has weight n. Here one weight one edge has to be discarded.

Our main result is a 3/4-approximation algorithm for $\texttt{MaxATSP}(0,1)$ with polynomial running time. As corollaries, we get a 5/4-approximation algorithm for $\texttt{ATSP}(1,2)$ and a 3/4-approximation algorithm for DPP.

Outline of our Algorithm

Kosaraju, Park, and Stein [6] formulate the so-called path coloring lemma. It states that if each node of a multigraph H has indegree and outdegree at most two and total degree at most three and H does not contain any 2-cycles (that is, a cycle with exactly two edges) or triple edges, then H is 2-path colorable. Kosaraju, Park, and Stein proceed with computing a cycle cover and a matching. (A cycle cover of a graph is a collection of node-disjoint directed cycles such that each node belongs to exactly one cycle.) If the matching is carefully chosen, then the union of the cycle cover and the matching fulfills the premises of the path coloring lemma and henceforth, is 2-path-colorable. (One also has to deal with the 2-cycles in the cycle cover separately, the interested reader is referred to the original work.) If one now takes the color class with the larger weight and patches the paths arbitrarily together, one gets a TSP tour that has at least half the weight of the combined weight of the cycle cover and the matching. The weight of an optimum cycle cover is at least the weight of an optimum TSP tour and the weight of an optimum matching is at least half the weight of an optimum TSP tour. Thus in the ideal case, this would yield an 3/4-approximation. However, Kosaraju, Park, and Stein have to deal with 2-cycles and have to avoid triple edges. Therefore, they only get a 48/63-approximation. This approach is refined in subsequent works [2, 7].

In this work, we directly compute a maximum weight multigraph that fulfills the premises of the path coloring lemma. This is done via an LP approach. The fractional solution H^* is then rounded to an integer one via an iterated decomposition scheme, inspired by the one of Alon [1]. Finally the integer solution is transformed into a TSP tour (or Path Packing) via the path coloring lemma.

References

[1] N. Alon. A simple algorithm for edge-coloring bipartite multigraphs. *Inform. Processing Letters* 85:301–302, 2003.

[2] M. Bläser. An 8/13-approximation algorithm for the maximum asymmetric TSP. *J. Algorithms*, 50(1): 23–48, 2004.

[3] M. Bläser and B. Siebert. Computing cycle covers without short cycles. In *Proc. 9th Ann. European Symp. on Algorithms (ESA)*, Lecture Notes in Comput. Sci. 2161, 369–379, Springer, 2001.

[4] M. Bläser and B. Manthey. Approximating maximum weight cycle covers in directed graphs with edge weights zero and one. Manuscript, 2003.

[5] H. Kaplan, M. Lewenstein, N. Shafrir, and M. Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. In *Proc. 44th Ann. IEEE Symp. on Foundations of Comput. Sci. (FOCS)*, pages 56–65, 2003.

[6] S. R. Kosaraju, J. K. Park, and C. Stein. Long tours and short superstrings. In *Proc. 35th Ann. IEEE Symp. on Foundations of Comput. Sci. (FOCS)*, pages 166-177, 1994.
[7] M. Lewenstein and M. Sviridenko. A 5/8 approximation algorithm for the maximum asymmetric TSP. *SIAM J. Discrete Math.*, 17:237–248, 2003.
[8] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Math. Oper. Res.*, 18:1–11, 1993.
[9] S. Vishwanathan. An approximation algorithm for the asymmetric travelling salesman problem with distances one and two. *Inform. Proc. Letters*, 44:297–302, 1992.

Approximation Hardness of Optimization Problems

Janka Chlebíková

In the last decade tight bounds on the efficient approximability for several problems have been achieved using the Probabilistic Checkable Proof (PCP) theory ([9], [11], [12]). However, the current state of the PCP technique hardly allows to obtain tight results for some basic problems (e.g., Steiner Tree, Metric Travelling Salesman) and for restricted instances of several optimization problems (e.g., in bounded degree graphs or in intersection graphs of some geometric objects). The research on restricted instances is particularly motivated as an intermediate step for proving approximation hardness of some important problems, but it is also of independent interest.

In restricted cases, gap preserving reductions are usually used to prove approximation hardness results for them from those already known. Due to problem dependence and lack of universal methods of designing gap preserving reductions, there are more open questions than known tight results in this area. We contribute to the systematic research of approximation hardness of various optimization problems and their restricted variants [2, 3, 4, 5, 6, 7].

In this contribution we present some gap preserving techniques which were used to obtain inapproximability results for various kinds of dominating set problems (see [7] for more details) and for many optimization problems restricted to some geometric graphs (e.g., intersection graphs of d-dimensional boxes).

Dominating Set problems. A dominating set D in a graph G is an *independent dominating set* if the subgraph G_D of G induced by D has no edges; D is a *total dominating set* if G_D has no isolated vertices; and D is a *connected dominating set* if G_D is a connected graph. The corresponding problems Minimum Independent Dominating Set (Min-IDS), Minimum Total Dominating Set (Min-TDS), and Minimum Connected Dominating Set (Min-CDS) ask for an independent, total, and connected dominating set of minimum size, respectively.

Due to a close relation of the Minimum Dominating Set problem to the Minimum Set Cover problem almost tight approximability results are known for the Min-DS problem in general graphs. The best upper bound, which is logarithmic in maximum degree of the graph, almost matches the lower bound.

Similar approximation results as in general graphs hold for MIN-DS, MIN-TDS, and MIN-CDS even in split and bipartite graphs, unless NP has slightly superpolynomial time algorithms.

For B-bounded graphs (B large) we prove asymptotically tight lower bounds of $\ln B$ (up to lower order terms) for MIN-DS, MIN-TDS, and MIN-CDS even in bipartite graphs. These results follow from Trevisan's approximation hardness result for size-bounded instances of Minimum Set Cover [13] using suitable original reductions.

The B-MIN-IDS problem completely differs from other studied variants of MIN-DS. We present the lower bound for B-MIN-IDS that increases linearly with B, similarly as known upper bound. The result can be obtained using a gap preserving reduction from a bounded occurrence version of the MAX-3-SAT problem.

The following table summarizes the current state of the research for dominating set problems in asymptotical case (upper bounds in all cases are due to [1], [8], [10]).

	B-MIN-DS	B-MIN-CDS	B-MIN-TDS	B-MIN-IDS
L	$\ln B - C \ln\ln B$	$\ln B - C \ln\ln B$	$\ln B - C \ln\ln B$	δB
U	$\mathcal{H}_{B+1} - \frac{1}{2}$	$\mathcal{H}_B + 2$	$\mathcal{H}_B - \frac{1}{2}$	$B - \frac{B-1}{B^2+1}$

We introduce various kinds of reductions to achieve lower bounds for B-MIN-DS and B-MIN-IDS problems for small values of B. The presented reductions start mainly from small degree instances of vertex cover, for which inapproximability results are known also in highly restricted cases (regular graphs with perfect matching) [2, 6]. All these lower bounds are summarized in the table (* means that lower bound is achieved even in bipartite graphs).

	3-MIN-DS	4-MIN-DS	5-MIN-DS	3-MIN-IDS	4-MIN-IDS	5-MIN-IDS
L	$\frac{391}{390}^*$	$\frac{100}{99}$	$\frac{53}{52}$	$\frac{681}{680}$	$\frac{294}{293}^*$	$\frac{152}{151}^*$
U	$\frac{19}{12}$	$\frac{107}{60}$	$\frac{117}{60}$	2	$\frac{65}{17}$	$\frac{63}{13}$

We show that in directed graphs with indegree bounded by a constant $B \geq 2$ the directed version of MIN-DS has simple $(B+1)$-approximation algorithm, but it is NP-hard to approximate within any constant smaller than $B-1$ for $B \geq 3$ (1.36 for $B = 2$). In directed graphs with outdegree bounded by a constant $B \geq 2$ we prove almost tight approximation lower bound of $\ln B$ for directed version of MIN-DS. We also point out that the problem in directed graphs is NP-complete (even in case when both in- and outdegree are bounded).

Optimization problems in d-boxes. The challenge problem in d-box intersection graphs ($d \geq 2$) is the MAXIMUM INDEPENDENT SET (MAXIS) problem: for a given set $\mathcal{R}$ of n axis-parallel d-dimensional boxes, find a maximum cardinality subset $\mathcal{R}^* \subseteq \mathcal{R}$ of pairwise disjoint boxes. As the problem is NP-hard for any fixed $d \geq 2$, the attention is focused on efficient approximation algorithms. However, in spite

of a great deal of efforts, it remained open possibilities on efficient approximability of MaxIS in d-box intersection graphs ($d \geq 2$) from the existence of a PTAS, to non-nonexistence of $o(\log^{d-1} n)$-approximation algorithm.

We answer in the negative the problem of existence of a PTAS for MaxIS in axis-parallel d-dimensional boxes for $d \geq 3$ and present some explicit NP-hard gap type results, even in highly restricted cases. Furthermore, we provide a generic method for the proof of APX-hardness (and hence non-existence of a PTAS) for many combinatorial optimization problems in intersection graphs of axis-parallel d-dimensional boxes for any fixed $d \geq 3$, e.g. Vertex Cover, Dominating Set, Edge Dominating Set, Independent Dominating Set, and Induced Matching.

The idea of our APX-hardness results is based on the following two results: (i) Many optimization problems are APX-hard even when restricted to suitable subdivisions of graphs of degree 3, e.g., $2k$ or $3k$ subdivision of each edge. (ii) Each graph obtained from an arbitrary graph by at least 3-subdivision of each edge is an intersection graph of d-boxes (for any $d \geq 3$). Moreover, its realization can be done in time polynomial in size of the subdivision graph.

This is joint work with Miroslav Chlebík, MPI Leipzig, Germany.

References

[1] P. Alimonti and T. Calamoneri. Improved approximations of independent dominating set in bounded degree graphs. In *Proceedings of the 22nd International Workshop on Graph-Theoretic Concepts in Computer Science,* WG 1996, LNCS, pages 2–16, 1996.

[2] M. Chlebík and J. Chlebíková. Approximation hardness for small occurrence instances of NP-hard problems. In *Proceedings of the 5th Conference on Algorithms and Complexity,* CIAC 2003, LNCS 2653, pages 152–164.

[3] M. Chlebík and J. Chlebíková. Approximation hardness of the Steiner tree problem on graphs. In *Proceedings of the 8th Scandinavian Workshop on Algorithm Theory,* SWAT 2002, LNCS 2368, pages 170–179.

[4] M. Chlebík and J. Chlebíková. On approximation hardness of the Min-2-SAT-deletion problem. In *Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science,* MFCS 2004.

[5] M. Chlebík and J. Chlebíková. Approximation hardness of minimum edge dominating set and minimum maximal matching. In *Proceedings of the 14th International Symposium on Algorithms and Computation,* ISAAC 2003, LNCS 2906, pages 415–424, 2003.

[6] M. Chlebík and J. Chlebíková. Inapproximability results for bounded variants of optimization problems. In *Proceedings of the 14th International Symposium on Fundamentals of Computation Theory,* FCT 2003, LNCS 2751, pages 27–38, 2003.

[7] M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems. In *Proceedings of the 12th Annual European Symposium on Algorithms,* ESA 2004, 2004.

[8] R. Duh and M. Fürer. Approximation of k-set cover by semi-local optimization. In *Proceedings of the 29th ACM Symposium on Theory of Computing,* STOC 1997, pages 256–264.

[9] U. Feige. A threshold of $\ln n$ for approximation set cover. *Journal of ACM*, 45(4):634–652, 1998.

[10] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. In *Proceedings of the 4th Annual European Symposium on Algorithms,* ESA 1996, LNCS 1136, pages 179–193, 1996.

[11] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182(1):627–636, 1999.
[12] J. Håstad. Some optimal inapproximability results. *Journal of ACM*, 48(4):798–859, 2001.
[13] L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proceedings of the 33rd Symposium on Theory of Computing,* STOC 2001, pages 453–461.

Coloring Semirandom Graphs Optimally

Amin Coja-Oghlan

The *graph coloring problem* – given a graph G, compute the chromatic number $\chi(G)$ – is of fundamental interest in theoretical computer science. At the same time, graph coloring is notoriously hard. Indeed, no polynomial time algorithm can approximate the chromatic number of graphs of order n within a factor of $n^{1-o(1)}$ (under a certain complexity theoretic assumption) [3]. This hardness results motivates the quest for *coloring heuristics* that run in polynomial time and succeed on "most" instances; Krivelevich [7] provides a recent survey.

To evaluate heuristics rigorously, we need to specify a stochastic model of the input instances. Kučera [8] has suggested the following model $G_{n,p,k}$ of random k-colorable graphs: first, partition the vertex set $V = \{1, \ldots, n\}$ into k classes $V_1, \ldots, V_k$ of size n/k randomly (we assume that k divides n). Then, include every possible V_i-V_j-edge with probability $p = p(n)$ independently ($i \neq j$). However, a drawback of this model is that the instances are purely random. As the theory of random graphs shows, such instances have a very particular combinatorial structure, so that designing heuristics for $G_{n,p,k}$ yields heuristics for a very special class of instances.

To figure out more "robust" heuristics, *semirandom models* have been studied, where problem instances are made up of a random share and an adversarial part. For instance, Blum and Spencer [1] have suggested a semirandom model $G^*_{n,p,k}$ of k-colorable graphs: first, choosing a random k-colorable graph $G_0 = G_{n,p,k}$. Let $V_1, \ldots, V_k$ be its planted k-coloring. Then, an adversary may add further V_i-V_j-edges ($i \neq j$) to complete the instance $G = G^*_{n,p,k}$. We say that $G^*_{n,p,k}$ has some property *with high probability* ("whp.") if this property holds with probability $1 - o(1)$ as $n \to \infty$ regardless of the adversary's decisions.

We have the following two results, which complement each other.

Theorem 1. *Suppose that $k = k(n)$ and $p = p(n)$ are such that $np \geq \max\{(1+\varepsilon)k\ln(n), C_0k^2\}$ for a certain constant C_0. There is a polynomial time algorithm* `Color` *that colors $G^*_{n,p,k}$ optimally whp.*

Theorem 2. *Let $3 \leq k \leq n^{99/100}$. There is no polynomial time algorithm that for $np \leq (1-\varepsilon)\frac{k}{2}\ln(n/k)$ k-colors $G^*_{n,p,k}$ whp., unless NP⊂RP.*

The best previous result on coloring $G^*_{n,p,k}$ is due to Feige and Kilian [4], who have suggested a semidefinite programming ("SDP") based heuristic that finds a k-coloring whp. if k is constant and $np \geq (1+\varepsilon)k\ln(n)$. Theorem 1 improves on this result in the following respects.

- It is not clear whether the algorithm in [4] can handle the case that k grows as a function of n (the analysis of the SDP rounding techniques does not seem to work in this case). In contrast, choosing $p = 1/2$ we can make k as large as $\Omega(\sqrt{n})$ in Theorem 1.
- The algorithm `Color` is somewhat simpler. For instance, it needs to solve an SDP only once, whereas [4] requires several SDP computations. However, the techniques of [4] apply to further problems that we do not address (e.g. "maximum independent set").
- Instead of just producing a k-coloring of $G = G^*_{n,p,k}$ whp., `Color` also provides a certificate that the output is indeed optimal.

Furthermore, Theorem 2 improves by a factor of $\frac{k}{2}$ on a hardness result from [4], where it is shown that it is NP-hard to k-color $G^*_{n,p,k}$ if $np \leq (1-\varepsilon)\ln(n)$. For $k = o(\ln n)$, Theorem 2 implies that the positive result Theorem 1 is essentially best possible (up to a factor of 2).

The algorithm `Color` for Theorem 1 employs a SDP relaxation $\bar{\vartheta}_2$ of the chromatic number, which has been studied by Szegedy [9]. The SDP $\bar{\vartheta}_2$ is defined as follows. A *rigid vector k-coloring* of a graph $G = (V, E)$ is a family $(x_v)_{v \in V}$ of unit vectors in $\mathbf{R}^n$ such that $\langle x_v, x_w \rangle \geq -\frac{1}{k-1}$ for all $v, w \in V$, and $\langle x_v, x_w \rangle = -\frac{1}{k-1}$ for all $\{v, w\} \in E$. Let

$$\bar{\vartheta}_2(G) = \inf\{k > 1 |\ G \text{ has a rigid vector } k\text{-coloring}\}.$$

Then, $\bar{\vartheta}_2(G) \leq \chi(G)$.

The algorithm `Color` for Theorem 1 is as follows. The input is a graph $G = (V, E)$, and the output is either an optimal coloring of G or "fail".

1. Compute $\bar{\vartheta}_2(G)$ along with a rigid vector $\bar{\vartheta}_2(G)$-coloring $(x_v)_{v \in V}$.
2. Let $H = (V, F)$ be the graph with edge set $F = \{\{v, w\} |\ \langle x_v, x_w \rangle \leq 0.995\}$. Apply the greedy algorithm for graph coloring to H. Let $\mathcal{C}$ be the resulting coloring.
3. If $\mathcal{C}$ uses at most $\lceil \bar{\vartheta}_2(G) \rceil$ colors, then output $\mathcal{C}$ as a coloring of G. Otherwise, output "fail".

Thus, `Color`(G) computes the rigid vector coloring $(x_v)_{v \in V}$ (this can be done in polynomial time via SDP [6]) to construct an auxiliary graph in which two vertices v, w are adjacent if and only if $\|x_v - x_w\| \geq 0.1$. To this graph H, `Color` applies the simple greedy algorithm that goes through the vertices V in a fixed order and colors each vertex v with the least color among $\{1, \ldots, n\}$ not yet used by the neighbors of v.

The proof of Theorem 1 is based on the fact that in $G = G^*_{n,p,k}$ whp. all optimal solutions to the SDP relaxation are "integral", i.e. encode colorings of G. The phenomenon that optimal fractional solutions are integral whp. has also been observed in the context of the minmum bisection and the maximum independent set problem [2, 4, 5].

References

[1] Blum, A., Spencer, J.: Coloring random and semirandom k-colorable graphs. J. of Algorithms **19** (1995) 203–234

[2] Boppana, R.: Eigenvalues and graph bisection: An average-case analysis. Proc. 28th FOCS (1987) 280–285
[3] Engebretsen, L., Holmerin, J.: Towards optimal lower bounds for clique and chromatic number. TCS **299** (2003) 537–584
[4] Feige, U., Kilian, J.: Heuristics for semirandom graph problems. JCSS **63** (2001) 639–671
[5] Feige, U., Krauthgamer, J.: Finding and certifying a large hidden clique in a semirandom graph. Random Structures & Algorithms **16** (2000) 195–208
[6] Grötschel, M., Lovász, L., Schrijver, A.: Geometric algorithms and combinatorial optimization. Springer (1988)
[7] Krivelevich, M.: Coloring random graphs – an algorithmic perspective, Proc. 2nd MathInfo (2002) 175-195.
[8] Kučera, L.: Expected behavior of graph coloring algorithms. Proc. 1st FCT (1977) 447–451
[9] Szegedy, M.: A note on the θ number of Lovász and the generalized Delsarte bound. Proc. 35th FOCS (1994) 36–39

The Natural Random Walk on the Transportation Polytope when the Number of Sources is Constant

Mary Cryan

(joint work with Martin Dyer, Haiko Müller and Leed Stougie)

In my Oberwolfach talk I presented results on the mixing time behaviour of a natural random walk on the edge-vertex graph of a *transportation polytope*. We are able to show that this walk converges to the uniform distribution on the vertex set in time $n^{O(m^2)}$ whenever the number of sources m is a constant. As far as we are aware, this is the first result proving rapid mixing of a random walk on the graph of any non-trivial class of polytopes. Very little is known about the mixing times of random walks on polytope graphs in general. In fact, it is not even known whether the diameter of the graph is polynomially bounded in the dimension and number of facets of the polytope. (See Kalai [6] and Ziegler [11].) In consequence, Markov chain Monte Carlo (MCMC) has not been well explored as a means of sampling, or approximately counting, vertices of general polytopes. Even for special classes of polytopes, such as arbitrary transportation polytopes, approximate counting algorithms are not known to exist, either by MCMC or by other means (see, for example, Pak [9]). In fact, the only previous mixing results known are for very special, and highly symmetric polytopes, such as the n-cube [3] and the Birkhoff polytope [10].

The *transportation problem* (TP) is the combinatorial optimization problem of assigning shipments of some commodity from sources to destinations so that the transportation cost is minimized. We are given a list of m sources and a list $r = (r_1, \ldots, r_m)$ of the quantities at each source (r_i is the quantity at source i). We are given a list of n destinations and a list $c = (c_1, \ldots, c_n)$ of the quantities required at each destination (c_j units are required at destination j). Without loss of generality, we will assume that $\sum_{i=1}^{m} r_i = \sum_{j=1}^{n} c_j$, so that demand exactly matches supply. We will let the total number of units be denoted by $N = \sum_{i=1}^{m} r_i$. In the general setting of the transportation problem, a list of *costs* is also given as input, but these are not relevant for the problem we consider.

For a given list of supplies $r = (r_1, \ldots, r_m)$ (available from the sources) and a given list of demands $c = (c_1, \ldots, c_n)$, the elements of the em transportation polytope $\mathbb{P}(r, c)$ can be represented (see, for example, Dyer, Kannan and Mount [5]) as the set of points $\{X_j^i : i \in [m-1], j \in [n-1]\}$ satisfying the system of inequalities (0.1)-(0.4):

$$X_j^i \geq 0,\ i \in [m-1], j \in [n-1] \tag{0.1}$$

$$\sum_{j \in [n-1]} X_j^i \leq r_i,\ i \in [m-1] \tag{0.2}$$

$$\sum_{i \in [m-1]} X_j^i \leq c_j,\ j \in [n-1] \tag{0.3}$$

$$\sum_{\substack{i \in [m-1], \\ j \in [n-1]}} X_j^i \geq N - r_m - c_n \tag{0.4}$$

The minimum cost for a TP is always attained at a vertex. Therefore counting and enumerating the vertices of transportation polytopes is of interest. Some results on the complexity of enumerating the vertices of a polytope appeared in Dyer [4], where it was shown to be #P-complete to count exactly the number of vertices of a $2 \times n$ transportation polytope,[1] and that it is NP-complete to decide if a $2 \times n$ transportation polytope is degenerate.

In this paper we consider the problem of sampling the vertices of $\mathbb{P}(r, c)$ almost uniformly at random, when the number of sources m is a constant. We define a Markov chain $\mathcal{W}$ on the set Ω of all vertices of $\mathbb{P}(r, c)$ and prove it is rapidly mixing when m is constant.

Our chain $\mathcal{W}$ is a random walk on the *edge-vertex graph* of the polytope $\mathbb{P}(r, c)$. This graph, also called the *skeleton* of the transportation polytope, contains a vertex Z for every vertex of $\mathbb{P}(r, c)$, and an edge (Z, W) for every pair of vertices Z, W that form an edge of $\mathbb{P}(r, c)$. We denote the edge-vertex graph by $G(\mathcal{W})$. In the full version of this work [2], we show that any vertex Z of $\mathbb{P}(r, c)$ has at most d_m incident edges, where $d_m = \lfloor me^{m-1}n^m \rfloor$ is polynomially bounded in n. A single step of our Markov chain $\mathcal{W}$ is performed as follows: if Z is the current vertex, we walk along any incident edge of Z with probability $1/2d_m$. It is not difficult to show that this Markov chain is ergodic, and that it converges to the uniform distribution on the vertices of the transportation polytope. However, our proof of rapid mixing involves more complex analytical techniques.

Our approach towards proving rapid mixing is inspired by that of Cryan, Dyer, Goldberg, Jerrum and Martin [1] for sampling contingency tables. This was itself based on the "balanced permutation" ideas of Morris and Sinclair [7, 8] for the knapsack problem. However, following the line of proof given in [1], and using the m-dimensional balanced permutations of [7], leads inevitably to a mixing time bound of $n^{2^{O(m)}}$. To obtain our improvement in the exponent, from exponential to polynomial, it is necessary to sharpen the tools of [7, 8] using the special structure

[1] In fact, [4] only claims NP-hardness, but the proof establishes #P-completeness.

of the problem at hand. Our improvement then results principally from the fact that we can prove that a *strongly* $O(m^2)$-balanced $n^{O(m^2)}$-uniform permutation exists for this problem. Note that it is unknown whether a strongly-balanced almost-uniform permutation exists for an arbitrary set of m-dimensional vectors. (See [7] for further information.)

More details and full proofs can be found in [2].

References

[1] M. Cryan, M. Dyer, L.A. Goldberg, M. Jerrum and R. Martin, Rapidly mixing Markov chains for sampling contingency tables with a constant number of rows. *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, pp. 711–720, 2002. Full version available at `http://www.dcs.warwick.ac.uk/~leslie /papers/contingency.ps`.

[2] M. Cryan, M. Dyer, H. Müller and L. Stougie, Random walks on the vertices of transportation polytopes with constant number of sources. *Proceedings of the 14th Annual Symposium on Discrete Algorithms*, pp. 330–339, 2003.

[3] P. Diaconis, R.L. Graham, and J.A. Morrison, Asymptotic analysis of a random walk on a hypercube with many dimensions. *Random Structures and Algorithms*, **1**, 1990, pp. 51–72.

[4] M.E. Dyer, The complexity of vertex enumeration methods. *Mathematics of Operations Research*, **8**(3), 1983, pp. 381–402.

[5] M. Dyer, R. Kannan and J. Mount, Sampling contingency tables. *Random Structures & Algorithms*, **10**(4), 1997, pp. 487–506.

[6] G. Kalai, Polytope skeletons and paths. In *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O'Rourke (Eds), CRC Press, Boca Raton, pp. 331–345.

[7] B. Morris, *Random Walks in Convex Sets*. PhD thesis, Department of Statistics, University of California, Berkeley, 2000.

[8] B. Morris and A.J. Sinclair, Random walks on truncated cubes and sampling 0-1 knapsack solutions. *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, 1999, pp. 230–240. Full version available from `http://www.cs.berkeley.edu/~sinclair /knapsack.ps`.

[9] I. Pak, On the Number of Faces of Certain Transportation Polytopes. *European Journal of Combinatorics*, **21**, 2002, pp. 689–694.

[10] I. Pak, Four questions on Birkhoff polytope. *Annals of Combinatorics*, **4**, 2000, pp. 83–90.

[11] G. M. Ziegler. Lectures on polytopes. *Graduate Texts in Mathematics* **152**, Springer-Verlag, 1995.

Sublinear Time Approximations for Metric MST and Facility Location Problems

Artur Czumaj

(joint work with Christian Sohler and in part with Mihai Bădoiu, Piotr Indyk, and Anastasios Sidiropoulos)

We present recent advances in the area of *sublinear-time approximation algorithms*. We first overview the existing results and briefly discuss the techniques used. Then, we present two recent results obtained for two classical combinatorial optimization problems: metric *Minimum Spanning Tree* [1] and metric uniform *Minimum Facility Location* problem [2].

The design of algorithms operating on massive data sets, has received a lot of attention in recent years. The practical motivation of this study is that polynomial

algorithms that are efficient in relatively small inputs, may become impractical for input sizes of several gigabytes. For example, when we consider approximation algorithms for clustering problems in metric spaces then they typically have $\Omega(n^2)$ running time where n is the number of input points. Clearly, such a running time is not feasible for massive data sets. But for many problems such a running time is provably unavoidable. Surprisingly, these lower bounds do not necessarily hold when one wants to estimate the *cost* of an optimal solution. In this talk, we will present two examples of problems for which one can estimate the cost of optimal solutions in sublinear-time: the metric Minimum Spanning Tree [1] and metric uniform Minimum Facility Location [2] problems.

Our approach is motivated by the fact that in many applications it suffices to know the *approximate cost* of the optimal solution to the problem rather than to find an approximate solution. Let us consider an example of a company that wants to invest money and it can relate the cost of the *facility location* problem to the possible return on investment. Then it would first solve an instance of the problem for every market to find out the most profitable one. In such a situation it is sufficient to know the return on investment before one decides which market to enter. It is not (yet) necessary to know how to achieve it. Finally, when one knows which market to enter one only has to compute a solution to a single instance of the problem. Therefore, if one could approximate the *cost* of an optimal solution significantly faster than finding such a particular approximate solution this would significantly speed up the market analysis.

Similar arguments holds for applications of facility location algorithms, i.e., to clusterings, and also for other combinatorial optimization problems, including the minimum spanning tree, the traveling salesman problem, and the minimum Steiner tree problem.

Specific results: metric MST and its applications [1]. For the Minimum Spanning Tree problem, we present a randomized algorithm that in time $O(n \log^{O(1)} n/\epsilon^{O(1)})$ returns a $(1+\epsilon)$-approximation of the *cost* of the minimum spanning tree of an n-point metric space. Since the full description of an n-point metric space is of size $\Theta(n^2)$, the complexity of our algorithm is *sublinear* with respect to the input size. Our algorithm is almost optimal as it is not possible to approximate in $o(n)$ time the cost of the minimum spanning tree to within any factor. Furthermore, it has been previously shown that no $o(n^2)$ algorithm exists that *returns a spanning tree* whose cost is within a constant times the optimum.

We also mention two interesting applications of our result: randomized $O(n \log^{(O(1)} n/\epsilon^{O(1)})$-time algorithms that return a $(2+\epsilon)$-approximation of the *cost* of the minimum-cost traveling salesman and the minimum Steiner tree of an n-point metric space.

Specific results: metric Facility Location [2]. Next, we discuss the problem of computing the optimal *cost* of the Minimum Facility Location problem, in the case of uniform costs and uniform demands. We present a randomized algorithm that runs in $O(n \log^2 n)$ time and that approximates the optimal cost to within a constant factor, where n is the number of metric space points. Since the size of the

representation of an n-point metric space is $\Theta(n^2)$, the complexity of our algorithm is *sublinear* with respect to the input size.

Furthermore, we prove that if the set of facilities and the cites (points that are to be connected to the facilities) are allowed to be disjoint, then any, even randomized, approximation algorithm for the cost of the Minimum Facility Location that guarantees any bounded approximation ratio for the cost, requires time $\Omega(n^2)$. Moreover, our proof can be extended to the problems of estimating the cost of *minimum-cost matching*, the cost of *bi-chromatic matching*, and the cost of *k-median* for $k = n/2$; all these problems require $\Omega(n^2)$ to estimate the cost of their optimal solution to within any factor.

References

[1] A. Czumaj and C. Sohler. Estimating the weight of metric minimum spanning trees in sublinear-time. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC'2004)*, pages 175–183, Chicago, IL, June 13–15, 2004. ACM Press, New York, NY.

[2] M. Bădoiu, A. Czumaj, P. Indyk, A. Sidiropoulos, and C. Sohler. Facility location in sublinear time. Manuscript in preparation, 2004.

Pass-Efficient Algorithms for Approximating Large Matrices

Petros Drineas

Introduction

We are interested in developing and analyzing fast Monte Carlo algorithms for performing useful computations on large matrices. We consider new methods for common problems such as matrix multiplication, the Singular Value Decomposition (SVD), and the computation of a compressed approximate decomposition of a large matrix. Since such computations generally require time which is superlinear in the number of nonzero elements of the matrix, we expect our algorithms to be useful in many applications where data sets are modeled by matrices and are extremely large. In all these cases, we assume that the input matrices are prohibitively large to store in Random Access Memory (RAM) and thus that only external memory storage is possible. Our algorithms will be allowed to read the matrices a few, e.g., one or two or three, times and keep a small randomly-chosen and rapidly-computable "sketch" of the matrices in RAM; computations will then be performed on this "sketch". We will work within the framework of the Pass-Efficient computational model, in which the scarce computational resources are the number of passes over the data, the additional RAM space required, and the additional time required [7, 8].

Recent interest in computing with massive data sets has led to the development of computational models in which the usual notions of time-efficiency and space-efficiency have been modified [3, 7, 13, 14, 15]. In the applications that motivate these data-streaming models the data sets are much too large to fit into main

memory. Thus, they are either not stored or are stored in a secondary storage device which may be read sequentially as a data stream but for which random access is very expensive. Typically, algorithms that compute on a data stream examine the data stream, keep a small "sketch" of the data, and perform computations on the sketch. Thus, these algorithms are usually randomized and approximate, and their performance is evaluated by considering resources such as the time to process an item in the data stream, the number of passes over the data, the additional workspace and additional time required, and the quality of the approximations returned. The motivation for our particular "pass-efficient" approach is that in modern computers the amount of disk storage (external memory) has increased enormously, while RAM and computing speeds have increased, yet at a substantially slower pace. Thus, we have the ability to store large amounts of data, but not in RAM, and we do not have the computational ability to process these data with algorithms that require superlinear time.

The Matrix Multiplication Algorithm

We present a simple, novel algorithm for the Matrix Multiplication Problem. Suppose A and B (which are $m \times n$ and $n \times p$ respectively) are the two input matrices. In our main algorithm, we perform $c = O(1)$ independent trials, where in each trial we randomly sample an element of $\{1, 2, \ldots n\}$ with an appropriate probability distribution $\mathcal{P}$ on $\{1, 2, \ldots n\}$. We form a $m \times c$ matrix C consisting of the sampled columns of A, each scaled appropriately, and we form a $c \times n$ matrix R using the same rows of B, again scaled appropriately. The choice of $\mathcal{P}$ and the column and row scaling are crucial features of the algorithm. When these are chosen judiciously, we prove that CR is a good approximation to AB; more precisely, we show that, with high probability,

$$\|AB - CR\|_F \in O(\|A\|_F \|B\|_F / \sqrt{c}),$$

where $\|\cdot\|_F$ denotes the Frobenius norm, i.e., $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$. This algorithm can be implemented without storing the matrices A and B in RAM, provided it can make two passes over the matrices stored in external memory and use $O(m+p)$ additional RAM memory to construct C and R.

The CUR approximation algorithm

We subsequently present an algorithm which, when given an $m \times n$ matrix A, computes approximations to A which are the product of three smaller matrices, C, U, and R, each of which may be computed rapidly. Let $A' = CUR$ be the computed approximate decomposition; our algorithm has provable bounds for the error matrix $A - A'$. The CUR algorithm chooses $c = O(1)$ columns of A and $r = O(1)$ rows of A randomly; if the $m \times c$ matrix C consists of those c columns of A (after appropriate rescaling) and the $r \times n$ matrix R consists of those r rows of A (also after appropriate rescaling) then the $c \times r$ matrix U may be calculated

from C and R. For any matrix X, let $\|X\|_F$ and $\|X\|_2$ denote its Frobenius norm and its spectral norm, respectively. It is proven that

$$\|A - A'\|_\xi \, x \leq \min_{D:\mathrm{rank}(D)\leq k} \|A - D\|_\xi + poly(k, 1/c) \, \|A\|_F$$

holds in expectation and with high probability for both $\xi = 2, F$ and for all $k = 1, \ldots, rank(A)$; thus by appropriate choice of k

$$\|A - A'\|_2 \leq \epsilon \, \|A\|_F$$

also holds in expectation and with high probability. This algorithm may be implemented without storing the matrix A in Random Access Memory (RAM), provided it can make two passes over the matrix stored in external memory and use $O(m+n)$ additional RAM memory. To achieve an additional error (beyond the best rank k approximation) that is at most $\epsilon \, \|A\|_F$, the CUR algorithm takes time which is a low-degree polynomial in $\max(m, n)$, k, $1/\epsilon$, and $1/\delta$. The proofs for the error bounds make important use of matrix perturbation theory and previous work on approximating matrix multiplication and computing low-rank approximations to a matrix. The probability distribution over columns and rows and the rescaling are crucial features of the algorithms and must be chosen judiciously.

A PTAS for the weighted Max-Cut Problem on dense graphs

Recent work in the development and analysis of randomized approximation algorithms for NP-hard problems has involved approximating the solution to a problem by the solution to an induced subproblem of constant size, where the subproblem is constructed by sampling elements of the original problem uniformly at random. In light of interest in problems with a heterogeneous structure, for which uniform sampling might be expected to yield suboptimal results, we investigate the use of nonuniform sampling probabilities. We show that by judicious choice of sampling probabilities and a variant of the CUR approximation algorithm, one can obtain error bounds that are superior to the ones obtained by uniform sampling for weighted versions of the Max-Cut problem, for certain regimes of the error parameter ϵ. Of particular interest is one of our techniques: we develop a method to approximate the feasibility of a large linear program by a nonuniformly randomly chosen subprogram; for more details see [11].

Related Work

In other related work, Achlioptas and McSherry have also computed succinctly-described matrix approximations using somewhat different sampling techniques [2, 1]. Also included in [2, 1] is a comparison of their methods with those of [7].

Recent work has focused on developing new techniques for proving lower bounds on the number of queries a sampling algorithm is required to perform in order to approximate a given function accurately with a low probability or error [4].

Acknowledgements: The aforementioned results emerged from joint work with Ravi Kannan (Computer Science Department, Yale University) and Michael W.

Mahoney (Math Department, Yale University); more details can be found in [12, 5, 6, 7, 11, 8, 9, 10].

References

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 611–618, 2001.

[2] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. *submitted*.

[3] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 20–29, 1996.

[4] Z. Bar-Yossef. Sampling lower bounds via information theory. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 335–344, 2003.

[5] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 291–299, 1999.

[6] P. Drineas and R. Kannan. Fast Monte-Carlo algorithms for approximate matrix multiplication. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 452–459, 2001.

[7] P. Drineas and R. Kannan. Pass efficient algorithms for approximating large matrices. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 223–232, 2003.

[8] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. Technical Report YALEU/DCS/TR-1269, Yale University Department of Computer Science, New Haven, CT, February 2004.

[9] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. Technical Report YALEU/DCS/TR-1270, Yale University Department of Computer Science, New Haven, CT, February 2004.

[10] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. Technical Report YALEU/DCS/TR-1271, Yale University Department of Computer Science, New Haven, CT, February 2004.

[11] P. Drineas, R. Kannan, and M.W. Mahoney. Sampling sub-problems of Max-Cut problems and approximation algorithms. *manuscript*.

[12] P. Drineas. *Randomized Algorithms for Matrix Operations*. PhD thesis, Yale University, 2003.

[13] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate L^1-difference algorithm for massive data sets. In *Proceedings of the 40th Annual IEEE Symposium on the Foundations of Computer Science*, pages 501–511, 1999.

[14] M.R. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. Technical Report 1998-011, Digital Systems Research Center, Palo Alto, CA, May 1998.

[15] J.I. Munro and M.S. Paterson. Selection and sorting with limited storage. In *Proceedings of the 19th Annual IEEE Symposium on Foundations of Computer Science*, pages 253–258, 1978.

Counting knapsack solutions

Martin Dyer

We give efficient algorithms to sample uniformly, and count approximately, the solutions to a zero-one knapsack problem. The algorithm is based on using dynamic programming to provide a *deterministic* relative approximation. Then "dart throwing" is used to give arbitrary approximation ratios.

Introduction

We describe an efficient algorithm to sample uniformly, and count approximately, solutions to the zero-one knapsack problem. The algorithm is based on a dynamic programming computation which provides a *deterministic* approximation ratio of polynomial size. Then simple "dart throwing" techniques give arbitrary approximation ratios. Previous approaches to this problem were based on the Markov chain Monte Carlo (MCMC) approach. See, for example, the survey of Jerrum and Sinclair [2]. The best result known, due to Morris and Sinclair [3], gave sampling in time $O(n^{9/2+\epsilon})$, for any $\epsilon > 0$, for a problem with n variables. We give an $O(n^3)$ time sampling algorithm and a fully polynomial randomized approximation scheme (*fpras*), with relative error ε, running in time $O(n^3 + \varepsilon^{-2}n^2)$, i.e. essentially the same time bound. The algorithm can be improved using randomized rounding, and also extended to several related problems [1].

The zero-one knapsack problem

Throughout, $\mathbb{N}$ will denote the set of all *non-negative* integers. For positive integers $i \le j$, we denote by $[i, j]$ the set of integers $\{i, \ldots, j\}$, and by $[j]$ the set $[1, j]$ for $1 \le j$.

Let S denote the solution set of

$$\sum_{j=1}^{n} a_j x_j \le b, \quad \text{with } x \in B_n = \{0,1\}^n,$$

where $0 \le a_1 \le a_2 \le \cdots \le a_n \le b$ are integers.[1]

Let k be such that $a_j \le b/n$ for $j \le k$ and either $k = n$ or $a_{k+1} > b/n$. Let $C = \{0,1\}^k \times \{0\}^{n-k}$. If $x \in C$ then $\sum_{j=1}^{n} a_j x_j \le \sum_{j=1}^{k} a_j \le kb/n \le b$, so $x \in S$. Thus $C \subseteq S$.

Let $\alpha_j = \lfloor n^2 a_j / b \rfloor$ and $\delta_j = n^2 a_j / b - \alpha_j$, so $0 \le \delta_j < 1$. Let S' be the solution set of

$$\sum_{j=1}^{n} \alpha_j x_j \le n^2, \quad \text{with } x \in B_n,$$

Now $|S'|$ can be determined in $O(n^3)$ time, using dynamic programming. Write $F(r, s) = |\{x \in B_r : \sum_{j=1}^{r} \alpha_j x_j \le s\}|$. In $O(n^3)$ time, the dynamic programming

[1] A single (rational) linear inequality in zero-one variables can always be put in this form.

tabulates $F(r,s)$ $(1 \le r \le n,\ 0 \le s \le n^2)$, using the recursion

$$\begin{aligned} F(1,s) &= \begin{cases} 1 & \text{if } s < \alpha_1 \\ 2 & \text{otherwise} \end{cases} \\ F(r,s) &= F(r-1,s) + F(r-1, s-\alpha_r) \quad (r \ge 2). \end{aligned}$$

Then we have $|S'| = F(n, n^2)$.

If $x \in S$, $\sum_{j=1}^n \alpha_j x_j \le (n^2/b) \sum_{j=1}^n a_j x_j \le (n^2/b)b = n^2$, so $x \in S'$. Thus $S \subseteq S'$ and $|S| \le |S'|$. If $S' \ne S$, suppose $x \in S' \setminus S$. Then clearly there exists an integer $p(x)$ such that $x_p = 1$ and $p \notin [k]$. Otherwise $x \in C \subseteq S \subseteq S'$, a contradiction. If there is more than one such integer, take $p(x)$ to be the smallest. Note that we have $\alpha_p \ge n$.

Define a map $f : S' \to B_n$, as follows. If $x \in S$ then $f(x) = x$. Otherwise $x \in S' \setminus S$, and $p(x)$ is well defined. Define $f(x) = y$, where $y_j = x_j$ for $j \ne p(x)$, and $y_p = 0$. If $x \in S' \setminus S$ then, with $y = f(x)$,

$$\begin{aligned} \sum_{j=1}^n a_j y_j &= \frac{b}{n^2} \sum_{j=1}^n (\alpha_j + \delta_j) y_j \\ &= \frac{b}{n^2} \Big(\sum_{j=1}^n \alpha_j y_j + \sum_{j=1}^n \delta_j y_j \Big) \\ &= \frac{b}{n^2} \Big(\sum_{j=1}^n \alpha_j x_j - \alpha_p + \sum_{j=1}^n \delta_j y_j \Big) \\ &\le \frac{b}{n^2} (n^2 - n + n) \\ &= b, \end{aligned}$$

so $f(x) \in S$. Hence $f(S') = S$. But, for $y \in S$, we have $|f^{-1}(y)| \le (n+1)$, since any element of $f^{-1}(y)$ may change a single coordinate of y or none. Thus

$$|S'| = |f^{-1}(S)| \le (n+1)|S|.$$

Hence $1 \le |S'|/|S| \le (n+1)$ so $|S'|/\sqrt{n+1}$ approximates $|S|$ deterministically within a factor $\sqrt{n+1}$ and can be computed in $O(n^3)$ time. Since knapsack is obviously self-reducible, existence of an for the problem now follows indirectly from a general result of Sinclair and Jerrum [2]. However, we will now describe a simpler and more efficient "dart-throwing" method to construct an *fpras* directly.

The $F(r,s)$ table can be used to determine a uniform point in S' in $O(n)$ time, by tracing back probabilistically from $F(n,n^2)$, as follows. With probability $F(n-1,n^2)/F(n,n^2)$ set $x_n = 0$, else set $x_n = 1$ with the remaining probability $F(n-1,n^2-\alpha_n)/F(n,n^2)$. If $x_n = 0$, recursively determine $x_{n-1}, x_{n-2}, \ldots, x_2, x_1$ by tracing back from $F(n-1,n^2)$ and, if $x_n = 1$, trace back similarly from $F(n-1,n^2-\alpha_n)$. The resulting point of S' has probability at least $1/(n+1)$ of lying in S. If so, it is uniformly distributed in S, and we accept it. Otherwise we repeat the whole process independently. After $n+1$ repetitions we have a sample with

probability at least $1 - e^{-1}$. Hence a sample of ν uniform points in S can be determined in $O(n^3 + n^2\nu)$ time[2] with probability at least $1 - e^{-\Omega(n)}$.

To have an *fpras* for $|S|$, we need only estimate the probability $\rho = |S|/|S'| \geq 1/(n+1)$, since $|S'| = F(n, n^2)$. With ν points in S', the sampling error is $O(1/\sqrt{\nu n})$. We require this to be smaller than $\varepsilon\rho = \Omega(\varepsilon/n)$. Hence we need $\nu = O(\varepsilon^{-2}n)$. The complexity of the *fpras* is then $O(n^3 + \varepsilon^{-2}n^2)$.

References

[1] Martin Dyer, Approximate counting by dynamic programming, in *Proc. 35th Annual ACM Symposium on Theory of Computing*, pp. 693–699, 2003.

[2] M. Jerrum and A. Sinclair, Markov chain Monte Carlo method: an approach to approximate counting and integration, in *Approximation Algorithms for NP-Hard Problems* (D.S. Hochbaum, ed.), PWS, Boston, pp. 482–520, 1997.

[3] B. Morris and A.J. Sinclair, Random walks on truncated cubes and sampling 0-1 knapsack solutions, July 2002, submitted. (A preliminary version appeared in *Proc. 40th IEEE Symposium on Foundations of Computer Science*, pp. 230–240, 1999.)

More Efficient Queries in PCPs for NP and Improved Approximation Hardness of Maximum CSP

Lars Engebretsen

Background. For more than a decade, one of the most powerful techniques for proving approximation hardness results for various types of discrete optimization problems, in particular constraint satisfaction problems, has been the use of Probabilistically Checkable Proofs (PCPs) for **NP**. In the PCP model, the verifier is given an input and oracle access to an alleged proof of the fact that the input belongs to some specified language. The verifier also has access to a specified amount of random bits. Based on the random bits and the input, the verifier decides which positions in the proof it should look at. Once it has examined the positions of its choice, it uses all available information to decide if the input should be accepted or rejected. The PCP theorem [1] asserts the startling fact that any language in **NP** can be probabilistically checked by a verifier that uses logarithmic randomness, always accepts a correct proof of an input in the language, accepts proofs of inputs not in the language with probability at most 1/2, and examines a *constant* number of bits of the proof. The probability that the PCP verifier accepts a correct proof of an input in the language is called the *completeness* c, while the probability that the verifier accepts any proof of an input not in the language is called the *soundness* s. It is generally desirable to have $c \approx 1$ and s as small as possible.

PCPs using a logarithmic number of random bits can be used to prove approximation hardness results for many combinatorial optimization problems. In particular, PCPs querying a small number of bits, say q bits, are intimately connected with Boolean q-ary constraint satisfaction problems: Strong approximation

[2]Here and elsewhere we count arithmetic operations, rather than operations on bits.

hardness results follow immediately from such PCPs with high completeness and low soundness.

Håstad's approximation hardness result for linear equations mod 2 gives such a characterization [4]: The verifier in his PCP for **NP** queries three bits, has completeness $1-\epsilon$ and soundness $1/2+\delta$ for arbitrary ϵ and δ. Allowing the verifier to make more queries to the proof is a natural way to lower the soundness even further; independent repetition of Håstad's protocol k times gives a PCP that queries $3k$ bits, has completeness at least $1-k\epsilon$ and soundness at most $(1/2+\delta)^k$. Hence the soundness goes down *exponentially fast* with the number of bits read from the proof. The purpose of our work [3] is to study exactly how fast the soundness can go down. There are several possible measures of "fast" in this context. One is the so called *amortized query complexity*: For a PCP with q queries, the amortized query complexity is defined as $\bar{q} = q/\log(c/s)$. The task of constructing PCPs for **NP** with low amortized query complexity—as well as the related question of testing if a function is linear—has been explored previously, most notably in a sequence of papers by Trevisan with different coauthors [8, 6, 5]. The key idea in those papers is to use *dependent repetitions* of Håstad's basic protocol. The technical part of the argument then boils down to showing that this dependence does not destroy the soundness of the verifier. We adapt and extend these previous ideas. In particular, we show that the idea of using dependent repetitions can be combined with the recently introduced *layered label cover problem* [2].

Our results. Another important efficiency measure for PCPs is the *free bit complexity*: A PCP has free bit complexity f if there are, for every outcome of the random bits used by the verifier, at most 2^f possible answers to the verifier's queries that make the verifier accept. Using the free bit complexity, our first main result can be written as follows:

Theorem 3. *For any integer $f \geq 2$, any positive integer $t \leq f(f-1)/2$, and any constant $\epsilon > 0$, there is a PCP for* **NP** *with free bit complexity f, query complexity $f+t$, completeness $1-\epsilon$, and soundness $2^{-t}+\epsilon$.*

To compare this with the previously best known result, due to Samorodnitsky and Trevisan [5], it is instructive to cast this result in terms of the amortized query complexity as a function of the number of queries:

Corollary 1. *For any integer $q \geq 3$ and any constant $\epsilon > 0$, there is a PCP for* **NP** *with query complexity q and amortized query complexity $1+\sqrt{2/q}+\epsilon$.*

Writing the soundness of our PCP as a function of the number of queries, we also get as an immediate corollary of our main result an improved approximation hardness result for the q-CSP problem:

Corollary 2. *For any integer $q \geq 3$, it is* **NP***-hard to approximate the q-CSP problem within $2^{q-\sqrt{2q-2}-1/2}$.*

The previously best known construction, due to Samorodnitsky and Trevisan [5] gives amortized query complexity $1+2/\sqrt{q}+\epsilon$ and hardness of approximation

within $2^{q-2\sqrt{q+1}+1}$. While our improvements might at first seem moderate, we remark that it is possible to approximate the q-CSP problem within 2^{q-1} in polynomial time and that a PCP for **NP** cannot have amortized query complexity $1 + 1/(q-1)$ unless $\mathbf{P} = \mathbf{NP}$; this follows from Trevisan's approximation algorithm for q-CSP [7]. Hence the only possible further improvements, unless $\mathbf{P} = \mathbf{NP}$, along this line of research concern the lower order term in $\bar{q}$ and the lower term in the exponent of the approximation factor—where we get an improvement by a factor of $\sqrt{2}$.

In the full version of our paper [3], we also show that there seems to be an underlying reason for why our improvement compared to previous work is moderate. Our construction follows a paradigm for query efficient PCPs for **NP** outlined by many previous researchers: On a high level, it combines a state-of-the-art "outer verifier" corresponding to a so called "layered label cover problem" with a corresponding "inner verifier" that is more query efficient than previously known verifiers. There are natural ways to extend this inner verifier in certain ways to produce what, at first, looks like even more query efficient verifiers. We prove, however, that all such extensions give verifiers that are *less query efficient* than our proposed verifier in the sense that the new verifiers have the same soundness as our verifier but pose more queries. This implies that significantly new ideas regarding proof composition and encoding of PCP proofs are required to construct PCPs for **NP** that are more query efficient than the one we propose here.

References

[1] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Márió Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

[2] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 595–601. San Diego, California, 9–11 June 2003.

[3] Lars Engebretsen and Jonas Holmerin. More efficient queries in PCPs for NP and improved approximation hardness of maximum CSP. Technical Report TRITA-NA-0409, Royal Institute of Technology, April 2004.

[4] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, July 2001.

[5] Alex Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 191–199. Portland, Oregon, 21–23 May 2000.

[6] Madhu Sudan and Luca Trevisan. Probabilistically checkable proofs with low amortized query complexity. In *39th Annual Symposium on Foundations of Computer Science*, pages 18–27. IEEE, Palo Alto, California, 8–11 November 1998.

[7] Luca Trevisan. Parallel approximation algorithms by positive linear programming. *Algorithmica*, 21(1):72–88, May 1998.

[8] Luca Trevisan. Recycling queries in PCPs and in linearity tests. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 299–308. Dallas, Texas, 23–26 May 1998.

The Hardness of Approximating Hereditary Properties

Uriel Feige
(joint work with Shimon Kogan)

A graph property π is a collection of graphs. A property π is called nontrivial if there are infinitely many graphs for which π holds and infinitely many graphs for which π does not hold. A nontrivial graph property is said to be *hereditary* if whenever a graph G satisfies property π then also every vertex induced subgraph of G satisfies π. The maximum subgraph with property π problem is defined in the following manner: Given a graph G find the maximum vertex induced subgraph of G which satisfies property π. The maximum subgraph problem is NP-hard for any nontrivial hereditary property [LY80].

It is shown in [LY93] that for any nontrivial hereditary property π which is false for some complete multipartite graph, the maximum subgraph with property π problem cannot be approximated within a factor of n^ϵ for some $\epsilon > 0$ unless $P = NP$. In particular this theorem applies to the following graph properties: complete graph, independent set, planar, outerplanar, bipartite, complete bipartite, acyclic, max degree, interval, chordal.

Furthermore it was proven in [LY93] that for every nontrivial hereditary property π, the maximum subgraph with property π problem cannot be approximated within a factor of $2^{(\log n)^c}$ for some $c > 0$, unless $NP \subseteq QP$. Here QP is the class of languages which can be recognized in quasipolynomial time, i.e. time $2^{(\log n)^d}$ for some constant d. The conclusion of this theorem applies to the graph properties stated above and the following graph properties: comparability, permutation, perfect, circular-arc, circle, line graph.

In [Has99] it was shown that max-clique cannot be approximated within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $NP = ZPP$. We prove the following result:

Theorem 4. *For every nontrivial hereditary property π and for every $\epsilon > 0$, the maximum subgraph with property π problem cannot be approximated within a factor of $n^{1-\epsilon}$, unless $NP = ZPP$.*

For nontrivial hereditary properties which are false for some clique or independent set, this result follows in a straightforward manner by combining Håstad's result with the proof described in [LY93]. Thus the main contribution of this paper is in showing that this hardness result holds even for nontrivial hereditary properties which hold for all cliques and all independent sets.

A hereditary property π for which feasibility can be decided in time at most exponential in the size of the input is called a feasible hereditary property. In [Hal00] it was shown that for each feasible hereditary property π, the maximum subgraph with property π problem can be approximated within a factor of $n/\log n$. The maximum hereditary subgraph problem can be approximated within a factor of $O(n(\log\log n/\log n)^2)$, for feasible properties that fail for some clique or independent set (Theorem 2.6 of [Hal00]).

In certain situations we may wish to find a subgraph which does not only satisfy a property π but is also connected. A property π is called nontrivial on connected graphs if it holds only for connected graphs, and there are infinitely many connected graphs for which π holds and infinitely many connected graphs for which π does not hold. A nontrivial graph property on connected graphs is said to be *hereditary* if whenever a connected graph G satisfies property π then also every vertex induced connected subgraph of G satisfies π. The maximum connected subgraph with property π problem is defined in the following manner: Given a graph G find the maximum vertex induced connected subgraph of G which satisfies property π. The maximum connected subgraph problem is NP-hard for any nontrivial hereditary problem [Yan79]. Examples of properties that are hereditary and nontrivial on connected graphs include: clique, star, complete bipartite, path, tree, planar, outerplanar, bipartite, chordal, interval, max degree and others.

It is shown in [LY93] that for every property that is nontrivial and hereditary on connected graphs, the maximum connected subgraph problem cannot be approximated with ratio $2^{(\log n)^c}$ for some $c > 0$, unless $NP \subseteq QP$. Furthermore it is stated in [LY93] that if π is a nontrivial hereditary property on connected graphs which is satisfied by all paths and does not hold for some complete bipartite graph, then the maximum connected subgraph with property π problem cannot be approximated within a factor of $n^{1-\epsilon}$ for every $\epsilon > 0$, unless $P = NP$. We prove the following results:

Theorem 5. *For every property that is nontrivial and hereditary on connected graphs and for every $\epsilon > 0$, the maximum connected subgraph problem cannot be approximated with ratio $n^{1-\epsilon}$, unless $NP = ZPP$.*

Theorem 6. *Let π be a nontrivial hereditary property on connected graphs which is satisfied by all paths and does not hold for some star. Then for every $\epsilon > 0$, the maximum connected subgraph with property π problem cannot be approximated within a factor of $n/(\log n)^{1+\epsilon}$, unless 3-SAT can be solved in time $2^{n^{1-\delta}}$ for some $\delta > 0$.*

The reduction used in the proof of theorem 6 is similar to the one used in in theorem 1 of [Yan79] to show the NP-hardness of the maximum connected subgraph problem. It is interesting to notice that it follows from theorem 6 that for certain hereditary properties it is harder to approximate the maximum connected subgraph problem then the maximum subgraph problem. For example by theorem 6 the maximum connected subgraph of degree smaller then k for every $k \geq 3$, cannot be approximate within a factor of $n/(\log n)^{1+\epsilon}$ (under the assumption that there is no subexponential time algorithm for 3-SAT). On the other hand by Theorem 2.6 of [Hal00] the maximum subgraph of degree smaller then k for every $k \geq 3$, can be approximated within a factor of $O(n(\log\log n/\log n)^2)$ and thus it is easier to approximate then its connected counterpart.

One can also consider hereditary properties in directed graphs as well as in undirected graphs. Examples of such properties are: acyclic, transitive, symmetric,

antisymmetric , tournament, max degree, line digraph. It was proved in [LY93] that for every nontrivial hereditary property on directed graphs, the maximum subgraph problem cannot be approximated with ratio $2^{(\log n)^c}$ for some $c > 0$, unless $NP \subseteq QP$. we prove the following:

Theorem 7. *For every nontrivial hereditary property on directed graphs and for every $\epsilon > 0$, the maximum subgraph problem cannot be approximated with ratio $n^{1-\epsilon}$, unless $NP = ZPP$.*

Acknowledgements This research was supported by The Israel Science Foundation (grant No. 263/02).

References

[Hal00] Magnús M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *Journal of Graph Algorithms and Applications*, 4(1):1–16, 2000.

[Has99] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(105-142), 1999.

[LY80] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980.

[LY93] Carsten Lund and Mihalis Yannakakis. The approximation of maximum subgraph problems. *In proceeding of Automata, Languages and Programming, 20th International Colloquium, volume 700 of Lecture Notes in Computer Science*, pages 40–51, July 1993.

[Yan79] Mihalis Yannakakis. The effect of a connectivity requirement on the complexity of maximum subgraph problems. *J. ACM*, 26(4):618–630, 1979.

An $\Omega(\log^\star n)$ Hardness of Approximation for the Asymmetric k-Center Problem

Sudipto Guha

Abstract

In this short note we discuss the approximability of the Asymmetric k-Center problem. In the Asymmetric k-Center problem, the input is an integer k and a complete digraph over n points together with a distance function obeying the directed triangle inequality. The goal is to choose a set of k points to serve as centers and to assign all the points to the centers, so that the maximum distance of any point to its center

This is the first natural problem whose approximability threshold does not polynomially relate to the known approximation classes.

Introduction

The input to the Asymmetric k-Center problem consists of a complete digraph G with vertex set V, a non-negative weight (or distance) function $c_{uv} \geq 0$ for every $u, v \in V$, and an integer k. The weight function c satisfies the directed triangle inequality, that is, $c_{uv} + c_{vw} \geq c_{uw}$ for all $u, v, w \in V$. Note that c_{uv} might differ from c_{vu}. The goal is to find a set S of k vertices, called *centers*, and

to assign each vertex of V to a center, such that the maximal distance of a vertex from its center is minimized. More formally, we want to find a subset $S \subseteq V$ of size k, that minimizes $\max_{v \in V} \min_{u \in S} c_{uv}$. This quantity is called the *covering radius* of the centers S.

The problem is well-known to be NP-hard [5] and although factor 2 approximation algorithms were known for the symmetric case (when $c_{uv} = c_{vu}$ for all $u, v \in V$), good approximation algorithms for Asymmetric k-Center were elusive. In a significant step, Panigrahy and Vishwanathan [6] designed an elegant $O(\log^* n)$ approximation algorithm for the Asymmetric k-Center problem, which was subsequently improved by Archer [2] to $O(\log^* k)$. Recently Chuzhoy *et al.* [3] showed that the approximation algorithms of [6, 2] are asymptotically the best possible, unless NP $\subseteq$ DTIME$(n^{\log \log n})$. This is a lower bound for a natural problem that does not conform to any of the known classes of approximation (see [1]). A hardness of $\log^* n$ is not even polynomially related to any of the known approximation classes.

In what follows we review the upper and lower bounds for the Asymmetric k-Center problem. We will focus on the bicriteria bounds, namely what is the best approximation that can be achieved using $2k$ centers. Thus we would prove a stronger lower bound and a weaker version of the upper bound, but the overall discussion will be significantly simpler due to this choice.

The Panigrahy-Vishwanathan upper bound

Panigrahy and Vishwanathan observed that we can guess the optimum answer. They set up a natural set covering instance where there is a set and an element each for every vertex. Element v belongs to set u if c_{uv} is less than the optimum radius. If there exists a solution using k centers for the Asymmetric k-Center problem, then the greedy algorithm yields a set cover of size $k(\log \frac{n}{k} + 1)$ for this set cover instance. Let this cover be S. We can now solve a set cover instance where the vertices corresponding to the sets in S are the elements. Observe that there exists a cover of size k using the solution of the Asymmetric k-Center problem. Thus the greedy algorithm yields a cover S' of size $k(\log(\log \frac{n}{k} + 1) + 1)$.

Observe that if the vertices corresponding to the sets in the cover S' were chosen as centers we would be covering every vertex within radius 2 times the optimum radius. This process after $\log^* n$ steps gives us a cover $\hat{S}$ of size at most $2k$, such that the vertices corresponding to $\hat{S}$ cover every other vertex within radius $\log^* n$ times the optimum in the original graph.

Panigrahy and Vishwanathan show that the $2k$ vertices can be reduced to k by blowing up the coverage radius by $O(1)$, the reader is referred to [6] for further details.

The $\log^* n$ lower bound

To motivate the construction, we start from a set cover instance. Given a collection of subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$ of a ground set U, we construct a 3 level graph. There is one vertex at level 0, m vertices in level 1 (corresponding to the

sets) and $|U|$ vertices at level 2 (corresponding to the elements). The root is at level 0, which has a directed edge to the m vertices at level 1 . The vertex s_i at level 1 (corresponding to the set S_i) has a directed edge to vertex j if and only if element j belongs to S_i. The length of each directed edge is 1.

It is easy to see that every vertex can be covered at radius 1 using $k+1$ centers iff there is a set cover of size k. If the triangle inequality does not hold then we can set all remaining lengths as ∞ and show that there exists a set cover of size k if and only if we have a bounded covering radius with $k+1$ centers. In presence of (directed) triangle inequality, choosing the root covers every vertex within radius 2.

A natural question arises: can we continue this trend, i.e., create a new set cover instance with m' sets and $|U'|$ elements with $m' = |U|$ and introduce a level 3 of $|U'|$ vertices where the directed edges from vertex i of level 2 to vertex j in level 3 is present if and only if j belongs to set i in the new set cover instance. We would like to argue that a small number of vertices can be used to cover all vertices at radius 1 if and only if both instances have small set covers.

But there is a difficulty to be overcome in the above - first the two set cover instances have to be correlated. This is not difficult. The hard part is to argue that if we had allocated all the centers to level 1 vertices even then we do not cover all the level 2 vertices within radius 1 and further, there is a vertex in level 3 which can only be reached (from the root) through the vertices in level 2 which are covered within radius 2. Thus we require a set cover instance where it is NP-hard to decide between the existence of a small set cover or any cover that excludes a small fraction of the sets leaves a fraction of elements uncovered.

Chuzhoy *et al.* in [3] show that using the recent hardness results for the d-Hypergraph Cover problem, given a 3SAT(5) formula and a parameter d we can construct a set cover instance such that if the input formula is satisfiable then an $2/d$-fraction of the sets are sufficient to cover all the elements. On the other hand, if at most $1-\epsilon$ fraction of the clauses of the given formula are satisfiable, then any collection of $(1-2/d)$-fraction of the sets covers at most a $(1-1/f(d))$-fraction of the elements where $f(d) = 2^{\text{poly}(d)}$. Further, the ratio of the number of elements to number of sets is $f(d)$.

Given the above, for the second set cover instance we choose $d' = 2df(d)$ (and assume that the number of sets is the same as level 3 vertices) then in the good case we would choose $2/d$ fraction of the level 1 vertices and $2/(2df(d))$ fraction of the level 2 vertices plus the root. Observes that this adds up to $2/d + 1/d$ fraction of the vertices of level 1 and the root (using the ratio of sets and elements for the first instance). If we were to keep adding levels using the recurrence $d_i = 2d_{i-1}f(d_{i-1})$ (with $d_1 = d$), it is easy to see that the total number of sets used across the level can be at most $4/d$ due to the geometrically decreasing terms.

In the good case (the formula is satisfiable) we would cover all vertices within radius 1. In the bad case, suppose we allocated all the centers to level 1 vertices then if $8/d_1 \ll 1 - 2/d_1$ we will cover at most $1 - 1/f(d)$ fraction of the vertices in level 1, and since $1 - 1/f(d_1) < 1 - 2/d_2$ at most $1 - f(d_2)$ level 2 vertices are

covered and so on. Thus if we have a $h+1$-level construction we can easily argue that there will be a vertex in level h none of whose ancestors are covered. This would imply that this element must be covered at distance h from the root.

Now, it is well known that it is NP-hard to decide if a 3SAT(5) formula lies in the good case or the bad case. Thus it is NP-hard to decide if in the graph we constructed, if there is a solution of radius 1 using k centers or radius h using $2k$ centers. How large can we make h ? Observe the recurrence sets up a tower of 2's and thus h can be made $\log^* n - O(1)$ while ensuring that the overall size of the graph remains subexponential.

To keep the discussion simple, we have been a bit inaccurate in the above description, e.g., we cannot control the parameter d as well as the number of sets, the parameters have to be slightly different due to technical reasons, etc., – but the above captures the main intuition of the lower bound proof. The reader is encouraged to look up the detailed reduction in [3].

References

[1] S. Arora and C. Lund. Hardness of approximation. In Dorit Hochbaum, editor, *Approximation Algorithms for NP Hard Problems*. PWS publishing Co, 1996.

[2] A. F. Archer. Two $O(\log^* k)$-approximation algorithms for the asymmetric k-center problem. *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization*, pp. 1–14, 2001.

[3] J. Chuzhoy, S. Guha, E. Halperin, S. Khanna, G. Kortsarz, R. Krauthgamer, and J. Naor. Asymmetric K-center is $\Omega(\log^* n)$ hard to approximate Submitted for publication.

[4] I. Dinur, V. Guruswami, and S. Khot. Vertex cover on k-uniform hypergraphs is hard to approximate within factor $(k-3-\epsilon)$. *Electronic Colloquium on Computational Complexity (ECCC)*, (027), 2002.

[5] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-completeness. W.H. Freeman and Company, 1979.

[6] R. Panigrahy and S. Vishwanathan. An $O(\log^* n)$ approximation algorithm for the asymmetric p-center problem. *J. of Algorithms*, 27(2):259–268, 1998.

On the advantage over a random assignment

Johan Håstad

(joint work with S. Venkatesh)

Given an NP-hard optimization problem we are interested in efficiently finding a reasonably good solution. Usually, an algorithm is said to be a c-approximation algorithm for a maximization problem if it, for each instance, produces a solution whose objective value is at least OPT/c where OPT is the global optimum. A more general criterion to evaluate its performance is:

$$\frac{OPT - X}{ALG - X}$$

where OPT is the optimum, ALG is the objective value of the solution output by the algorithm and X is a parameter to be chosen. In the usual definition, X is chosen to be zero since in most optimization problems, all feasible solutions have

non-negative values. Another possibility previously used by Bellare and Rogaway [2] is to let X be the minimum possible value of a feasible solution.

In this paper, we focus on constraint satisfaction problems with an underlying Boolean predicate P. Håstad [4] has shown that for many constraint satisfaction problems like Max-Lin-p, in which we are required to maximize the number of satisfied equations in a system of linear equations modulo a prime p, and Max-E3-Sat, in which we are required to maximize the number of satisfied clauses in a 3-CNF formula, the random assignment algorithm essentially yields the best possible approximation ratio. This makes the study of a new measure of approximation that compares the performance of an algorithm for a constraint satisfaction problem with the random assignment algorithm interesting. In a related work, Alon, Gutin and Krivelevich [1] recently studied a new measure for approximation algorithms called the domination ratio.

Preliminaries. We are given a system of m linear equations modulo 2 in n variables, together with positive weights w_i, $1 \le i \le m$.

We consider two cases: Max-k-Lin-2, in which each equation only contains at most k variables and Max-Lin-2, the general case. If W is the total weight of all equations our performance measure is given by

$$\max_L \frac{SAT[OPT(L)] - W/2}{SAT[ALG(L)] - W/2} \tag{0.1}$$

where L is an instance, SAT[OPT(L)] denotes the total weight of equations satisfied by the optimal solution and SAT[ALG(L)] denotes the total weight of equations satisfied by the solution output by the algorithm ALG.

Previous results give some bounds for our current measure. In particular, using Håstad's results [4], it can be shown, for $k \ge 3$, that it is hard to approximate Max-k-Lin-2 (and hence Max-Lin-2) within c for every $c > 1$ unless NP=P and within $(\log m)^c$ for some constant $c > 0$ unless NP $\subseteq$ DTIME$\left[m^{O(\log\log m)}\right]$.

Our results. We start with a randomized approximation scheme for our new measure for the case of few variables in each equation.

Theorem 8. *Consider Max-k-Lin-2. There exists a fixed constant $c > 1$ such that the following holds: for any $k \in O(\log n)$, there is a randomized polynomial time algorithm that, with probability at least 3/4, outputs an assignment that gives an approximation ratio at most $c^k\sqrt{m}$.*

We improve on the inapproximability results mentioned above by using a slightly stronger assumption.

Theorem 9. *Unless NP $\subseteq$ DTIME$\left[2^{(\log m)^{O(1)}}\right]$, for all $k \ge 3$ and $\epsilon > 0$, there is no algorithm that approximates Max-k-Lin-2 within $2^{(\log m)^{1-\epsilon}}$ and runs in time $2^{(\log m)^{O(1)}}$.*

The proof of this result is similar to the proof of the above cited result by Håstad but replaces the long code by the split code defined by Khot [3]. This bound can be strengthened if we allow more variables in each equation.

Theorem 10. *There exists a constant $\gamma > 0$ such that it is NP-hard to approximate Max-Lin-2 within m^γ.*

This proof uses an idea from derandomization and in particular it is based on the "walk on expanders" construction. If we allow randomization, we can get a stronger inapproximability result.

Theorem 11. *For any $\epsilon > 0$, unless NP $\subseteq$ RP, there is no randomized polynomial time algorithm that, with probability at least $\frac{1}{2}$, outputs an assignment for Max-Lin-2 with an approximation ratio at most $m^{\frac{1}{2}-\epsilon}$.*

The best upper bound we can show for the general case is rather poor.

Theorem 12. *For any $c > 0$, there is a randomized polynomial time algorithm that, with probability $3/4$, outputs an assignment for Max-Lin-2 with approximation ratio at most $\frac{m}{c \log m}$. There is also, for any $c > 0$, a deterministic approximation algorithm that approximates Max-Lin-2 within $\frac{m}{c}$.*

Note: This paper was presented at FOCS 2001 and is accepted for publication in Random Structures and Algorithms.

References

[1] N. Alon, G. Gutin and M. Krivelevich, Algorithms with large domination ratio, *Journal of Algorithms*, Vol 50, 2004, pages 118–131.

[2] M. Bellare and P. Rogaway, The complexity of approximating a nonlinear program, *Journal of Mathematical Programming B*, 69:3, 1995, pages 429–441.

[3] S. Khot, Hardness Results for Approximate Hypergraph Coloring, *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, 2002, pages 351–359.

[4] J. Håstad, Some optimal inapproximability results, *Journal of ACM*, Vol 48, 2001, pages 798-859.

PTAS for Dense Steiner Tree Problems

Mathias Hauptmann

We study the approximability of the Steiner Tree Problem and various related problems restricted to *dense instances*. According to Karpinski and Zelikovsky [KZ97a], an instance of the Steiner Tree Problem is called ϵ-dense if it consists of graph $G = (V, E)$ (all edge weights are 1) and terminal set $S \subseteq V$ such that each terminal has at least an ϵ-fraction of $V \setminus S$ in its direct neighborhood. They proved that for every fixed $\epsilon > 0$, the ϵ-Dense Steiner Tree Problem admits a polynomial time approximation scheme. We obtain polynomial time approximation schemes for the ϵ-dense Group Steiner Tree Problem and the ϵ-dense variants of the Prize Collecting Steiner Tree Problem and k-Steiner Tree Problem. For the ϵ-average dense case of these problems we obtain APX-hardness results. For the Steiner Tree Problem in ϵ-everywhere graphs we obtain a PTAS by a reduction to the ϵ-Dense Steiner Tree Problem.

Introduction

The Steiner Tree Problem asks for a minimum cost tree connecting a given set of points $S \subseteq V$ (called the set of terminals) in a graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}_+$. The Steiner Tree Problem is known to be NP-hard even to approximate. The currently best known lower bound for approximability is ≈ 1.01 due to Chlebík and Chlebíková [CC02], and Robins and Zelikovsky [RZ00a] gave a series of polynomial time algorithms with ratio $1 + \ln(3)/2 + \epsilon \approx 1.55 + \epsilon$ ($\epsilon > 0$ arbitrary small).

Here we consider the *ϵ-Dense Steiner Tree Problem*, where the instance consists of a graph $G = (V, E)$ (all edge weights are 1, i.e. we are just counting edges) and a terminal set $S \subseteq V$ such that the following condition holds:

$$\epsilon\text{-Density: } \forall\, s \in S\, |N_{V\setminus S}(s)| \geq \epsilon \cdot |V \setminus S| \qquad (\star)$$

where for some vertex v and some set of vertices $U \subseteq V$, $N_U(v) = \{u \in U | \{u, v\} \in E\}$ is the neighborhooh of v in U. Karpinski and Zelikovsky obtained the following result.

Theorem 13. *(Karpinski, Zelikovsky 1997* [KZ97a]*)*
For every fixed $\epsilon > 0$, there is a polynomial time approximation scheme for the ϵ-Dense Steiner Tree Problem.

Our Results

We consider ϵ-dense versions of various generalizations of the Steiner Tree Problem and study relaxations of the density condition as well as average and everywhere density. In particular we obtain the following results:

Dense Group Steiner, Prize Collecting and k-Steiner Tree Problems. The Group Steiner Tree Problem is the following: Given a graph $G = (V, E)$ with edge weights $c(e) \geq 0, e \in E$ and pairwise disjoint groups $S_i \subseteq V, 1 \leq i \leq m$, construct a minimum cost tree T containing at least one representative from each group S_i. Known algorithms for the general case of this problem achieve a polylogarithmic approximation ratio [GKR98]. An instance $G = (V, E), S_i, 1 \leq i \leq m$ of the Group Steiner Tree Problem in graphs (all edge weights are 1) is called *ϵ-dense* if for each $1 \leq i \leq m$ $|N_{V\setminus S}(S_i)| \geq \epsilon \cdot |V \setminus S|$, where $S := \bigcup_i S_i$. In the *k-Steiner Tree Problem* we are given an instance of the usual Steiner Tree Problem with terminal set S and a number $k \in \{1, \ldots, |S|\}$, and we have to construct a minimum cost tree T containing at least k vertices from S. Given a graph $G = (V, E)$, edge costs $c\colon \to \mathbb{R}$, terminal set S and a function $p\colon S \to \mathbb{R}$, the *Prize Collecting Steiner Tree Problem* asks for a tree T containing a possibly empty subset $S' \subseteq S$ such as to minimize $c(T) + p(S \setminus S')$. For both problems, an instance is called *ϵ-dense* if edge weights are all equal to 1 (graph case) and the density condition $(\star)$ holds. We obtain the following result:

Theorem 14. *For each $\epsilon > 0$, the ϵ-dense versions of the Group Steiner Tree Problem, the Prize Collecting Steiner Tree Problem and the k-Steiner Tree Problem provide a polynomial time approximation scheme.*

The Algorithm for the ϵ-Dense Group Steiner Tree Problem. The basic idea was already used by Karpinski and Zelikovsky for obtaining a PTAS for the Dense Steiner Tree Problem: The density condition enables us to perform a cheap greedy pick. Namely, there exists at least one vertex $v \in V \setminus S$ with $|\{i|\, N_{S_i}(v) \neq \emptyset\}| \geq \epsilon \cdot m$. We pick such vertex v, arbitrarily choose neighbors s_i in the neighbored groups S_i and contract the resulting star into a single vertex. Hence at this point we decide which representatives of the involved groups will be part of the tree. In order to preserve density, we declare the vertex resulting from contraction as a *non-terminal* and iterate. In this way we are able to perform $O(\log(m))$ greedy picks to reduce number of terminals to constant. Then we redeclare the contracted vertices as one-element groups. Now for each choice of representatives of the remaining groups not involved in contractions so far, we solve exactly an instance of the graph Steiner Tree Problem with a logarithmic number of terminals. Using Dreyfus-Wagner algorithm, this can be done in polynomial time. Note that the constants as well as the degree of the polynomial depend on ϵ as well as on the precision parameter δ.

Obtaining Efficient Approximation Schemes

When performing greedy picks we remove the resulting vertices from the terminal set. This happens in order to preserve the density condition and results in a running time of the form $O(n^{1/\delta})$. In order to obtain a running time bound not depending exponentially on $1/\delta$, we have to reduce the number of terminals to constant instead of logarithmic. This can be done as follows: Although after one round of greedy picks, ϵ-density cannot be guaranteed anymore, we observe that each terminal has lost at most one of its non-terminal neighbors. Hence for sufficiently large terminal set S $\frac{\epsilon}{2}$-density will still hold. Now we can iterate the greedy process $\log^\star(|S|)$ times, guaranteing $\frac{\epsilon}{2}$-density in each iteration and resulting in a terminal set of constant size.

Summary

Our results for the dense versions of the Steiner Tree Problem and related problems are listed in the two tables below. Possible extensions include the relaxation of the density notions considered so far and defining some intermediate density condition between everywhere and average density.

Problem Definition	**General Case**		**ϵ-Dense**
	Upper	Lower	Upper
Steiner Tree Problem	≈ 1.55 [RZ00a]	≈ 1.01 [CC02]	PTAS [KZ97a]
Group Steiner Tree	polylog. [GKR98]	$(1-\delta)\cdot\log(n)$	**PTAS**
k-Steiner Problem	$2+\delta$ [AK00]	≈ 1.01 [CC02]	**PTAS**
Prize Collecting STP	$2\cdot\left(1+\frac{1}{\lvert S\rvert}\right)$ [GW92]	≈ 1.01 [CC02]	**PTAS**
Steiner Forest	$2\cdot\left(1+\frac{1}{\lvert S\rvert}\right)$ [GW92]	≈ 1.01 [CC02]	**improved ratio**

Problem	Density Condition	Result
in ϵ-Everywhere Dense Graphs	$\forall\, v \in V\ d_G(v) \geq \epsilon \cdot n$	**PTAS**
in ϵ-Average Dense Graphs	$\lvert E\rvert \geq \epsilon \cdot n^2$	**APX-hard**
ϵ-Average Dense Steiner Tree P.	$\lvert E(S, V \setminus S)\rvert \geq \epsilon \cdot \lvert S\rvert \cdot \lvert V \setminus S\rvert$	**APX-hard**

References

[AK00] S. Arora and G. Karakostas. A 2+epsilon approximation for the k-MST problem. In *Proc. SIAM Symp. Discrete Algorithms (SODA)*, 2000

[CC02] M. Chlebikov and J. Chlebikova. Approximation hardness of the steiner tree problem on graphs. In *Proceedings of 8th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 170–179. LNCS 2368, 2002.

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[GKR98] Garg, Konjevod, and Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[GW92] Goemans and Williamson. A general approximation technique for constrained forest problems. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 1992.

[KZ97a] Marek Karpinski and Alexander Zelikovsky. Approximating dense cases of covering problems. In *Proc. DIMACS Workshop on Network Design: Connectivity and Facilities Location, Princeton,* 1997, pp. 169-178.

[K01] Marek Karpinski. Polynomial Time Approximation Schemes for Some Dense Instances of NP-Hard Problems. Algorithmica 30 (2001), pp. 386-397.

[RZ00a] G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *Proc. 11th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 770–779. ACM-SIAM, 2000.

Coupling with Stationarity: Rapid Sampling for Graph Coloring

Tom Hayes

(joint work with Eric Vigoda)

Let G be a graph on n vertices with maximum degree Δ, and let $k \geq \delta + 2$. For these parameters, it is trivial to construct a k-coloring of G. However the problem of exactly counting the number of such colorings is $\sharp P$-complete. (By k-coloring we mean a function $f\colon V \to \{1, \ldots, l\}$ such that $\forall$ edges $\{u, v\}$ $f(u) \neq f(v)$.) We are interested in the related problem of almost-uniform random generation of k-colorings of G.

The "Glauber dynamics" is a very simple Markov chain on state space $\Omega = \{k\text{-colorings of } G\}$.Given $X_t \in \Omega$, the distribution of $X_{t+1} \in \Omega$ is defined by: Choose $v \in V$ uniformly at random, and $c \in \{1, \ldots, k\} \setminus X_t(N(v))$, uniformly at random, where $N(v) = \{w \in V | \{v, w\} \in E\}$ is the set of neighbors of v. Then

$$X_{t+1}(w) \quad - \quad \begin{cases} c & \text{if } w = v \\ X_t(w) & \text{otherwise} \end{cases}$$

For $k \geq \Delta+2$, this is an ergodic Markov chain with stationary distribution uniform over Ω. For $k > 2\Delta$, the chain is known to converge rapidly to stationarity: $\forall X_0 \forall T \geq Cn\log(n/\epsilon)$, $||\mu_T - \pi|| \leq \epsilon$, where π is the uniform distribution over Ω and μ_T is the distribution of X_T. This is conjectured to hold whenever $k \geq \Delta+2$. However, for $k < \frac{11\Delta}{6}$, it is not known whether $Cn\log(n)$ can be replaced by a subexponential dependency on n.

Under some restrictions on G, better results are known. Notably, $O(n\log n)$ convergence has been shown assuming:

- $k \geq 1.763\Delta, k = \Omega(\log(n))$ and $girth(G) \geq 5$[Dyer, Frieze 2001], [Hayes 2003]
- $k \geq 1.489...\Delta, k = \Omega(\log(n))$ and $girth(G) \geq 6$[[Hayes 2003], [Molloy 2002]
- $k \geq (1+\epsilon)\Delta, k = \Omega(\log(n))$ and $girth(G) \geq 9$[Hayes, Vigoda 2003]

All these results have rather complicated proofs, and rely on proving certain regularity properties hold with high probability for X_T when T is sufficiently large.

We present a simpler proof of $O(n\log n)$ convergence time, assuming

$$k \geq 1.763..\Delta, k = \Omega(\log(n)) \text{ and } girth(G) \geq 4 \quad \text{(i.e. triangle-free)}.$$

Our method only requires regularity properties to be established for the stationary distribution, which is typically much easier to analyze than the distributions μ_T.

More generally, we present an extension to the Coupling Inequality, a classical and general technique for proving rapid mixing of Markov chains [Doeblin 1938]:

Suppose $\rho\colon \Omega \times \Omega \to \mathbb{N} \cup \{0\}$ is an integer valued distance metric on Ω, and $S \subseteq \Omega$ satisfies $\forall (X_t, Y_t) \in S \times \Omega$: There is a coupling $(X_t, Y_t) \to (X_{t+1}, Y_{t+1})$ of the Markov chain with itself, such that

$$E(\rho(X_{t+1}, Y_{t+1})) \leq (1-\epsilon)\rho(X_t, Y_t).$$

Then, $\forall T \geq 0, \forall X_0$,

$$||\mu_T - \pi||_1 \leq diam(\Omega) \cdot ((1-\epsilon)^T + \frac{\pi(\Omega \setminus S)}{\epsilon}),$$

where $diam(\Omega) = \max_{X,Y} \rho(X,Y)$.

Transformations to totally unimodular optimizations, Half Integrality and 2-Approximations

Dorit Hochbaum

We describe a set of techniques for dealing with intractable problems. The techniques consist of a transformation of the the problem constraint matrix into a matrix that is totally unimodular. The transformed problem, since defined on a totally unimodular constraint matrix, is easy to solve in polynomial time. The transformations are effective when the *loss* of integrality is bounded. That is, the inverse transformation of the integer solution maps it to non-integers that are integer multiple of some $\frac{1}{\alpha}$. In most cases we show that the value of α is 2 thus providing $\frac{1}{2}$ integer solution that are superoptimal. Superoptimality means that the associated objective value of such a solution is only better than the optimal

integer solution. In that sense the superoptimal solution provides a bound on the optimum solution. In many cases, which we characterize, it is possible to derive α-approximate solutions from this superoptimal solution.

We now add more details on each one of the problem classes listed.

1. Two variables per inequality

This is the class of integer programs with up to two variables per inequality. So a generic constraint would be of the form $ax + by \leq c$. All these problems have a 2-approximation algorithm resulting from the transformation to a minimum cut problem. The monotonization transformation and the existence of rounding procedure leading to a 2-approximation algorithms are discussed in [HMNT93]. Major applications include the complement of maximum clique problem, minimum weight of true variables in a 2SAT truth assignment and the set cover problem.

2. Three variables per inequality

This class of problems is integer programs with up to three variables per inequality, where the third variable appears in one constraint only. $ax+by \leq c+dz$. If $d = 1$ then the transformation yields superoptimal half integral solutions. If there exists a feasible rounding then this rounded solution is a 2-approximation. For $|d| > 1$ the inverse transformation yields rationals with denominator $2d$ at most. In that case the approximation factor can be as large as $2d$. The algorithm solving the transformed problem is also a minimum cut algorithm.

$$\begin{array}{ll} \text{Min} & \sum_{j=1}^{n} w_j x_j + \sum e_i z_i \\ \text{subject to} & a_i x_{j_i} + b_i x_{k_i} \geq c_i + d_i z_i \quad \text{for } i = 1, \ldots, m \\ & \ell_j \leq x_j \leq u_j \quad j = 1, \ldots, n \\ & z_i \ \text{ integer} \qquad i = 1, \ldots, m \\ & x_j \ \text{ integer} \qquad j = 1, \ldots, n. \end{array}$$

The transformation process and numerous applications resulting in 2-approximation algorithms are described in [Hoc02]. Applications to problems involving cliques and bicliques are given in [Hoc98].

3. Integer and linear programs with two nonzeroes per column

This problem is investigated in [Hoc03]. For linear programs with up to two nonzeroes per column the transformation transforms the problem to a generalized flow problem. Although the latter is not known to be solvable in polynomial time in integers, it is solvable more efficiently than a general linear program.

When the two nonzero values are of absolute value 1, the problem is also known as the *bidirected network flow* problem. The transformation maps the problem into a flow problem which is solved in integers in polynomial time. The inverse map provides half integral super optimal solution for this problem. This solution serves as a tight bound for a relaxation of the maximum cut problem.

4. Integer problems with k blocks of 1s per column/row.

The multi-set cover problem with k blocks of 1s per *column* is the basic problem in mutli-shift scheduling. In [HL03] we describe the process of mapping the constraints into a matrix of consecutive 1 per column. The latter is known to be equivalent to a flow problem. The inverse mapping provides solutions that when rounded form a k-approximation for the problem.

The set cover problem with up to k blocks of 1s per row is the formulation of the rectangle stabbing problem in a k dimensional space. Gaur, Ibaraki, Krishnamurti showed (2002) that by solving the linear programming relaxation first, and finding for each row the block that covers at least $\frac{1}{d}$, one gets a new formulation with consecutive 1's per row on this single found block. The (integer) solution for this formulation is a k approximation for the stabbing problem (since it is of value at most k times the linear programming optimum).

5. Tree "paths" structure integer programming

This interesting case of transformation to totally unimodular matrices was pointed out to me by R. Ravi at the Oberwolfach approximation algorithms workshop. This provide a 2-approximation algorithm for minimum cost 2-edge connected subgraph. Take a minimum spanning tree in the graph, and suspend it from an arbitrarily selected root node. We now want to add minimum cost selection of out of tree arcs that will doubly connect every pair. We will restrict it to edges that go between a node i in the tree and its ancestor $j = a(i)$ only. Adding such edge assures double connectivity of all nodes on the path between i and j. So we present it as a covering problem where each edge covers the nodes along the path $[i, j]$. Each such edge is assigned its own weight in the set cover problem.

Now for edges that are cross edges (i, j) (do not go between a node and an ancestor) we replace those by two edges going from i and from j to their least common ancestor $a(i, j)$. each one of these edges get the same weight. This is where the factor of 2 appears.

A matrix with each column corresponding to a path in the tree – that is, have the value 1 for each node (or edge) that is on the path – is totally unimodular. To see that, we use the necessary and sufficient condition of Ghouila-Houri 1962 (p. 269 Schrijver), specifying that a matrix is totally unimodular iff for every subset of rows there is a partition of the subset to $S_1 \cup S_2$ so that the sum of the rows in S_1 minus the sum of the rows in S_2 is a row vector with no entry larger than 1 in absolute value. Now, given a subset of rows S we sort them according to the level of each node in the tree. So in S we might have level $L(S) = 2, 5, 6, 9, 11$ etc. Now we assign alternating levels in the sorted set of levels to S_1 and S_2 alternately. the sum for each column will be either 0 if it includes even number of levels in S, or 1 or -1 if the number of levels is odd.

This proves that the constraint matrix of this set cover is totally unimodular. Of course the same set cover solves the k-connectivity within a factor of 2. Just have each right hand side equal to k.

References

[HL03] D. S. Hochbaum and A. Levin. Cyclical scheduling and multi-shift scheduling: complexity and approximation algorithms. Manuscript, UC Berkeley, June 2003.

[HMNT93] D. S. Hochbaum, N. Megiddo, J. Naor and A. Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Mathematical Programming*, 62, 69–83, 1993.

[Hoc98] D. S. Hochbaum. Approximating clique and biclique problems. *Journal of Algorithms*, 29, 174–200, (1998).

[Hoc02] D. S. Hochbaum. Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. *European Journal of Operational Research* 140:2, 2002, 291-321.

[Hoc03] D. S. Hochbaum. Monotonizing linear programs with up to two nonzeroes per column. *Operations Research Letters* 32:1 2003, 49-58.

Approximation Algorithms for the Weighted Matching Problem

Stefan Hougardy

(joint work with Doratha E. Drake)

[1]We present a linear time approximation algorithm for the weighted matching problem with a performance ratio of $2/3 - \epsilon$. This improves the previously best performance ratio of $1/2$.

A *matching* M in a graph $G = (V, E)$ is a subset of the edges of G such that no two edges in M are incident to the same vertex. In a graph $G = (V, E)$ with edge weights given by a function $w : E \to \mathbb{R}^+$ the *weight of a matching* M is defined as $w(M) := \sum_{e \in M} w(e)$. The *weighted matching problem* is to find a matching in G that has maximum weight. The first polynomial time algorithm for the weighted matching problem was given by Edmonds [7] in 1965. A straightforward implementation of this algorithm requires a running time of $O(n^2m)$, where n and m denote the number of vertices and edges in the graph. Lawler [13] and Gabow [8] improved the running time to $O(n^3)$. Galil, Micali, and Gabow [11] presented an implementation of Edmond's algorithm with a running time of $O(nm \log n)$. This was improved by Gabow, Galil, and Spencer [10] to a running time of $O(nm \log\log\log n + n^2 \log n)$. The fastest known algorithm to date for solving the weighted matching problem in general graphs is due to Gabow [9] and has a running time of $O(nm + n^2 \log n)$.

Together with the research on improving the worst case running time of Edmond's algorithm for the weighted matching problem there has been a parallel line of research concerned with the implementations of these algorithms. Implementations of Edmond's algorithm that turn out to be efficient in practice usually not only require the use of sophisticated data structures but also need additional new ideas to lower the running time in practice. During the last 35 years many

[1]supported by DFG research grant 296/6-3, supported in part by DFG Research Center *Mathematics for key technologies*, this is an extended abstract of the results presented in [5]

different implementations of Edmond's weighted matching algorithm have been presented. See [2] for a good survey on these. Currently the fastest implementations of Edmond's algorithm are due to Cook and Rohe [2] and to Mehlhorn and Schäfer [14].

Many real world problems require graphs of such large size that the running time of the fastest available weighted matching algorithm is too costly. Therefore, there is considerable need for *approximation algorithms* for the weighted matching problem that are very fast, and that nevertheless produce very good results even if these results are not optimal. The quality of an approximation algorithm for the weighted matching problem is measured by its so-called *performance ratio*. An approximation algorithm has a performance ratio of c, if for all graphs it finds a matching with a weight of at least c times the weight of an optimal solution.

Approximation algorithms for the weighted matching problem have been used in practice already for a long time. Their good running times are one of the main motivations for using them. Another reason why these algorithms are used in practice is that they usually require only a few lines of code for their implementation contrary to several thousand lines of code that a good implementation of Edmond's algorithm may require [2].

The two approximation algorithms that are most often used in practice are variants of the maximal matching algorithm and the greedy algorithm. A *maximal matching* in a graph is a matching that is not properly contained in any other matching. Such a matching can easily be computed by starting with an empty matching and extending it in each step by an arbitrary edge in such a way that it remains a matching. Several variants of this simple algorithm are used in practice [12]. The advantage of maximal matching algorithms is that they have linear running time. The major disadvantage of these algorithms is that they have a performance ratio of 0, i.e., the solutions returned by these algorithms can be arbitrarily bad. The greedy algorithm works similarly to the maximal matching algorithm but chooses in each step not an arbitrary but the heaviest edge currently available. It is easy to see that the greedy algorithm has a performance ratio of $\frac{1}{2}$ [1]. The running time of this algorithm is $O(m \log n)$ as it requires sorting the edges of the graph by decreasing weight.

Preis [15] was the first who was able to combine the advantages of the greedy algorithm and the maximal matching algorithm in one algorithm. In 1999 he presented a linear time approximation algorithm for the weighted matching problem with a performance ratio of $\frac{1}{2}$. The idea of his algorithm is to replace the heaviest edge that is needed in the greedy algorithm by a so called *locally heaviest* edge. It is easy to see that the performance ratio of Preis' algorithm is $\frac{1}{2}$. But it is difficult to prove that finding a locally heaviest edge in each step can be done in such a way that the total running time remains linear.

By using a completely different approach Drake and Hougardy [3] obtained another linear time approximation algorithm for the weighted matching problem with a performance ratio of $\frac{1}{2}$. The main idea of their algorithm is to grow in a

greedy way *two* matchings simultaneously and return the heavier of both as the result. Their algorithm and its analysis are simpler than that of Preis.

In [4] the idea of local improvements is used as a postprocessing step to enhance the performance of approximation algorithms for the weighted matching problem in practice. This postprocessing step requires only linear running time and it is shown for a large set of test instances that it significantly improves the quality of the solutions. However, this postprocessing step does not improve the performance ratio of $\frac{1}{2}$.

As the main result we prove in [5, 6] that there exist linear time approximation algorithms for the weighted matching problem that have performance ratios arbitrarily close to $\frac{2}{3}$.

Main Theorem 1. *For each $\epsilon > 0$ there exists a linear time approximation algorithm for the weighted matching problem with a performance ratio of $\frac{2}{3} - \epsilon$.*

The dependence on ϵ of the running time of these algorithms is quite moderate. Moreover our new algorithm is easy to implement and therefore is of relevance in practice.

The main idea of our algorithm is to start with a maximal matching M and to increase its weight by local changes. These local changes which we call *short augmentations* add in each step at most two new edges to M while up to four edges of M will be removed. A graph can possess up to $\Omega(n^4)$ short augmentations. To achieve linear running time only some part of these can be looked at. For each edge of the maximal matching M our algorithm only looks at all short augmentations that involve the endpoints of this edge. The maximality of M ensures that the short augmentations considered by the algorithm are in some sense spread evenly over the graph.

As the short augmentations are partly overlapping it can happen that after performing one short augmentation several others are no longer available. For the performance ratio it is therefore important to be able to reject short augmentations that achieve only minor improvements in the weight of the matching. This is achieved by taking only short augmentations into consideration that gain at least some constant factor β. Such augmentations are called β-augmentations. In linear time it seems not to be possible to find the best β-augmentation. However we show that in linear time a constant factor approximation of the best β-augmentation can be found.

To prove the performance ratio of our algorithm we use an amortized analysis. The idea is that the gain that is achieved by an augmentation is not realized immediately but part of it is stored for later use. This way we are able to prove that the algorithm increases the weight of the given matching by some constant factor. By repeating the algorithm a constant number of times and choosing β sufficiently small the resulting matching will have a weight that comes arbitrarily close to $\frac{2}{3}$.

References

[1] D. Avis, A Survey of Heuristics for the Weighted Matching Problem, Networks, Vol. 13 (1983), 475–493.
[2] W. Cook, A. Rohe, Computing minimum-weight perfect matchings, INFORMS Journal on Computing 11 (1999), 138–148.
[3] D.E. Drake, S. Hougardy, A Simple Approximation Algorithm for the Weighted Matching Problem, Information Processing Letters 85 (2003), 211–213.
[4] D.E. Drake, S. Hougardy, Linear Time Local Improvements for Weighted Matchings in Graphs, In: Workshop on Efficient Algorithms (WEA) 2003, K.Jansen, M.Margraf, M.Mastrolli, J.D.P.Rolim (eds.), LNCS 2647, Springer 2003, 107–119.
[5] D.E. Drake, S. Hougardy, Improved linear time approximation algorithms for weighted matchings, In: Approximation, Randomization, and Combinatorial Optimization, (Approx/Random) 2003, S.Arora, K.Jansen, J.D.P.Rolim, A.Sahai (eds.), LNCS 2764, Springer 2003, 14–23.
[6] D.E. Drake, S. Hougardy, A linear time approximation algorithm for weighted matchings in graphs, Technical Report, Humboldt-Universität zu Berlin, December 2003.
[7] J. Edmonds, Maximum matching and a polyhedron with 0,1-vertices, J. Res. Nat. Bur. Standards 69B (1965), 125–130.
[8] H.N. Gabow, An efficient implementation of Edmond's algorithm for maximum matching on graphs, Journal of the ACM 23 (1976), 221–234.
[9] H.N. Gabow, Data Structures for Weighted Matching and Nearest Common Ancestors with Linking, SODA 1990, 434–443.
[10] H.N. Gabow, Z. Galil, T.H. Spencer, Efficient implementation of graph algorithms using contraction, J. ACM 36 (1989), 815–853.
[11] Z. Galil, S. Micali, H. Gabow, An O(EV log V) algorithm for finding a maximal weighted matching in general graphs, SIAM Journal on Computing 15 (1986), 120–130.
[12] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20:1 (1998), 359–392.
[13] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.
[14] K. Mehlhorn, G. Schäfer, Implementation of $O(nmlogn)$ Weighted Matchings in General Graphs: The Power of Data Structures, In: S. Näher, D. Wagner (eds.), 4th International Workshop on Algorithm Engineering (WAE) 2000, Saarbrücken, Germany, September 5–8, LNCS 1982, Springer 2001, 23–38.
[15] R. Preis, Linear Time 1/2-Approximation Algorithm for Maximum Weighted Matching in General Graphs, Symposium on Theoretical Aspects of Computer Science (STACS) 1999, C. Meinel, S. Tison (eds.), LNCS 1563, Springer 1999, 259–269.

Improved approximation algorithm for the mixed fractional packing and covering problem

Klaus Jansen

We study mixed fractional packing and covering problems (MPC_ϵ) of the following form: Given a vector $f : B \to \mathbb{R}_+^M$ of M nonnegative continuous convex functions and a vector $g : B \to \mathbb{R}_+^M$ of M nonnegative continuous concave functions, two M - dimensional nonnegative vectors a, b, a nonempty convex compact set B and a relative tolerance $\epsilon \in (0, 1)$, find an approximately feasible vector $x \in B$ such that $f(x) \leq (1 + \epsilon)a$ and $g(x) \geq (1 - \epsilon)b$ or find a proof that no vector

is feasible (that satisfies $x \in B$, $f(x) \leq a$ and $g(x) \geq b$). W.l.o.g. we may assume that a and b are equal to the vector e of all ones.

The fractional packing problem with convex constraints, i.e. to find $x \in B$ such that $f(x) \leq (1+\epsilon)a$, is solved in [3, 4, 7] by the Lagrangian decomposition method in $O(M(\epsilon^{-2} + \ln M))$ iterations where each iteration requires a call to an approximate block solver $ABS(p,t)$ of the form: find $\hat{x} \in B$ such that $p^T f(\hat{x}) \leq (1+t)\Lambda(p)$ where $\Lambda(p) = \min_{x \in B} p^T f(x)$. Furthermore, Grigoriadis et al. [5] proposed also an approximation algorithm for the fractional covering problem with concave constraints, i.e. to find $x \in B$ such that $g(x) \geq (1-\epsilon)b$, within $O(M(\epsilon^{-2}+\ln M))$ iterations where each iteration requires here a call to an approximate block solver $ABS(q,t)$ of the form: find $\hat{x} \in B$ such that $q^T g(\hat{x}) \geq (1-t)\Lambda(q)$ where $\Lambda(q) = \max_{x \in B} q^T g(x)$. Both algorithms solve also the corresponding min-max and max-min optimization variants within the same number of iterations. Furthermore, the algorithms can be generalized to the case where the block solver has arbitrary approximation ratio [6, 7, 8].

Further interesting algorithms for the fractional packing and fractional covering problem with linear constraints were developed by Plotkin et al. [11] and Young [13]. These algorithms have a running time that depends linearly on the width - an unbounded function of the input instance. Several relatively complicated techniques were proposed to reduce this dependence. Garg and Könemann [2] described a nice algorithm for the fractional packing problem with linear constraints that needs only $O(M\epsilon^{-2} \ln M)$ iterations. On the other hand, the algorithm by Grigoriadis et al. [5] is the only known algorithm that solves the fractional covering problem with a number of iterations independently on the width.

For the mixed packing and covering problem (with linear constraints), Plotkin et al. [11] proposed also approximation algorithms where the running time depends on the width. Young [14] described an approximation algorithm for a special mixed packing and covering problem with linear constraints and special convex set $B = \mathbb{R}_+^N$. The algorithm has a running time of $O(M^2\epsilon^{-2} \ln M)$. Recently, Fleischer [1] gave an approximation scheme for the optimization variant (minimizing $c^T x$ such that $Cx \geq b$, $Px \leq a$ and $x \geq 0$ where a, b, and c are nonnegative integer vectors and P and C are nonnegative integer matrices). Young [14] posed the following interesting open problem: find an efficient width-independent Lagrangian-relaxation algorithm for the abstract mixed packing and covering problem: find $x \in B$ such that $Px \leq (1+\epsilon)a$, $Cx \geq (1-\epsilon)b$, where P, C are nonnegative matrices, a, b are nonnegative vectors and B is a polytope that can be queried by an optimization oracle (given a vector c, return $x \in B$ minimizing $c^T x$) or some other suitable oracle.

New Result: Our contribution is an efficient width-independent Lagrangian-relaxation algorithm for the mixed packing and covering problem. Interestingly, our algorithm works also for a more general problem with a convex set B and nonnegative convex packing and concave covering constraints. The algorithm uses a variant of the Lagrangian or price directive decomposition method. This is an iterative strategy that solves (MPC_ϵ) by computing a sequence of triples (p, q, x)

as follows. A coordinator uses the current vector $x \in B$ to compute two price vectors $p = p(x) \in \mathbb{R}_+^M$ and $q = q(x) \in \mathbb{R}_+^M$ with $\sum_{m=1}^M p_m + q_m = 1$. Then the coordinator calls a feasibility oracle to compute a solution $\hat{x} \in B$ of the block problem $BP(p, q)$

$$\text{find } x \in B \text{ s.t. } p^T f(\hat{x}) \leq q^T g(y) + \alpha,$$

(where α is a value that depends on p and q) and makes a move from x to $(1 - \tau)x + \tau\hat{x}$ with an appropriate step length $\tau \in (0, 1)$. Such a iteration is called a coordination step. Our main result is the following:

Theorem 15. *There is an approximation algorithm that for any given accuracy $\epsilon \in (0, 1)$ solves the mixed fractional packing and covering problem (MPC_ϵ) within*

$$N = O(M(\epsilon^{-2} \ln \epsilon^{-1} + \ln M))$$

iterations or coordination steps, where each of which requires a call to the block problem $BP(p, q)$ and a coordination overhead of $O(M \ln(M\epsilon^{-1}))$ arithmetic operations.

Alternatively, we can use also approximate variants of the block problem. Independently, Khandekar and Garg [9] proposed an approximation algorithm for the mixed fractional packing and covering problem that uses $O(M\epsilon^{-2} \ln M)$ iterations or coordination steps.

References

[1] L. Fleischer, A fast approximation scheme for fractional covering problems with variable upper bounds, *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*, SODA 2004.

[2] N. Garg and J. Könemann, Fast and simpler algorithms for multicommodity flow and other fractional packing problems, *Proceedings of the 39th IEEE Annual Symposium on Foundations of Computer Science*, FOCS 1998, 300-309.

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

[4] M.D. Grigoriadis and L.G. Khachiyan, Coordination complexity of parallel price-directive decomposition, *Mathematics of Operations Research*, 2 (1996), 321-340.

[5] M.D. Grigoriadis, L.G. Khachiyan, L. Porkolab and J. Villavicencio, Approximate max-min resource sharing for structured concave optimization, *SIAM Journal on Optimization*, 41 (2001), 1081-1091.

[6] K. Jansen and L. Porkolab, On preemptive resource constrained scheduling: polynomial-time approximation schemes, *Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization*, IPCO 2002, LNCS 2337, 329-349.

[7] K. Jansen and H. Zhang, Approximation algorithms for general packing problems with modified logarithmic potential function, *Proceedings of the 2nd IFIP International Conference on Theoretical Computer Science*, TCS 2002, Foundations of information technology in the era of network and mobile computing, Kluwer Academic Publisher, 2002, 255-266.

[8] K. Jansen, Approximation algorithms for the general max-min resource sharing problem: faster and simpler, to appear in: Proceedings of the 9th Scandinavian Workshop on Algorithm Theory, SWAT 2004.

[9] R. Khandekar and N. Garg, personal communication, 2004.

[10] J. Könemann, Fast combinatorial algorithms for packing and covering problems, Diploma Thesis, Max-Planck-Institute for Computer Science Saarbrücken, 2000.

[11] S.A. Plotkin, D.B. Shmoys, and E. Tardos, Fast approximation algorithms for fractional packing and covering problems, *Mathematics of Operations Research*, 20 (1995), 257-301.
[12] J. Villavicencio and M. D. Grigoriadis, Approximate Lagrangian decomposition with a modified Karmarkar logarithmic potential, *Network Optimization*, Lecture Notes in Economics and Mathematical Systems, 450 (1997), 471-485.
[13] N.E. Young, Randomized rounding without solving the linear program, *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms* SODA 1995, 170-178.
[14] N.E. Young, Sequential and parallel algorithms for mixed packing and covering, *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2001, 538-546.

Random Sampling and Approximation of MAX-CSP Problems

Ravi Kannan
(joint work with Noga Alon, W. Fernandez de la Vega, and Marek Karpinski)

Suppose r is a fixed integer. In the MAX-rSAT problem, we are given a Conjunctive Normal Form Boolean formula on n variables, with each clause being the OR of precisely r literals. The objective is to maximize the number of clauses satisfied by an assignment to the n variables. The exact problem is NP-hard for each fixed $r \geq 2$. A special case of our result is that for any $\epsilon > 0$, there is a positive integer $q \in O(\log(1/\epsilon)/\epsilon^4)$ such that if we pick at random a subset of q variables (among the n) and solve the "induced" problem on the q variables (maximize the number of clauses satisfied among those containing only those variables and their negations), then the answer multiplied by n^r/q^r is, with high probability, within an additive factor ϵn^r of the optimal answer for the n variable problem. The q needed here will be called the "sample complexity" of the problem for obvious reasons.

In fact, we show the same result for all MAX-rCSP problems. (MAX-rCSP problems, also called MAX-rFUNCTION-SAT, are equivalent to MAX-SNP [2]). Recall that the input to a MAX-rCSP problem (for r fixed) consists of a set F of m distinct Boolean functions $f_1, f_2, \ldots f_m$ of n Boolean variables $x_1, x_2, \ldots x_n$, where each f_i is a function of only r of the n variables. The answer $\mathrm{Max}(F)$ is the maximum number of functions which can be simultaneously set to 1 by a truth assignment to the variables. For a subset Q of the variables, we let F^Q denote the subset of F which are functions of only the variables in Q (and their negations).

Theorem 16 (Main Theorem)**.** *Let r, n be positive integers, with r fixed. Suppose ϵ is a positive real. There exists a positive integer $q \in O(\log(1/\epsilon)/\epsilon^4)$ such that for any F (as above), if Q is a random subset of $\{x_1, x_2, \ldots x_n\}$ of cardinality q, then with probability at least 9/10, we have*

$$|\frac{n^r}{q^r} Max(F^Q) - Max(F)| \leq \epsilon n^r.$$

It is worth noting that one half of the Theorem - namely the assertion that

$$\frac{n^r}{q^r}\mathrm{Max}(F^Q) - \mathrm{Max}(F) \geq -\epsilon n^r$$

is relatively easy to prove. Indeed, if, the assignment of truth values to $x_1, x_2, \ldots x_n$ achieving $\mathrm{Max}(F)$ sets to 1 a set S of functions among $f_1, f_2, \ldots f_m$, one can show that a sufficient number of functions in S are in F^Q just from the fact that Q is random. This then says that the same assignment restricted to Q sets to 1 a sufficient number of functions. So, a good solution to the whole problem provides also good solutions to random induced sub-problems. The other half -

$$\frac{n^r}{q^r}\mathrm{Max}(F^Q) - \mathrm{Max}(F) \leq \epsilon n^r$$

is much harder. Intuitively, for proving this part, we have to show that if there is no good solution to the whole problem, then also, there are no good solutions to random induced sub-problems.

The MAX-rSAT and other MAX-rCSP problems all admit fixed factor relative approximation algorithms which run in polynomial time. For some MAX-SNP problems, there have been major breakthroughs in achieving better factors using semi-definite programming and other techniques. More relevant to our paper is the line of work started with the paper of Arora, Karger and Karpinski [2] which introduced the technique of smooth programs, and designed the first polynomial time algorithms for solving MAX-SNP problems (of arity r) to within additive error guarantee ϵn^r, for each fixed $\epsilon > 0$. Frieze and Kannan [4] proved an efficient version of Szémeredi's Regularity Lemma and used it to get a uniform framework to solve all MAX-SNP and some other problems in polynomial time with the same additive error. In [5], they introduced a new way of approximating matrices and more generally r-dimensional arrays, called the "cut-decomposition" and using those, proved a result somewhat similar to the main result here (for each fixed r), but with two important differences - (i) the sample complexity was exponential in $1/\epsilon$ and (ii) their result did not relate the optimal solution value of the whole problem to the optimal solution of the random sub-problems in their original setting; instead it related it to a complicated computational quantity associated with the random sub-problem. We will make central use of cut-decompositions in this paper.

For the special case of $r = 2$, Goldreich, Goldwasser and Ron [6] designed algorithms, where the sample complexity was polynomial in $1/\epsilon$; indeed, by exploiting the special structure of individual problems like the MAX-CUT problem they improved the polynomial dependence. See also [4], for higher dimensional cases, or for cases in which our only objective is to decide if we can satisfy almost all constraints. Our new method here is more uniform and general.

Our result is derived by general Linear Algebra based arguments about approximating multi- (and 2-) dimensional arrays by some simple arrays and then using Linear Programming arguments. Unlike previous papers, we do not use problem-specific arguments which dwell into the special structure of individual problems. The MAX-CUT problem (a special MAX-2CSP problem) has received much attention in this context. Indeed, independently of the papers so far cited, Fernandez de la Vega [3] developed a different algorithm for this problem which within polynomial time, produced a solution with additive error ϵn^2. [6] used the

special structure of the problem to derive an algorithm with sample complexity $O(1/\epsilon^5)$ (best known upto now).

Independently of our work, Anderson and Engebretsen [1] have obtained a constant time approximation algorithm for MAX-rCSP. They state their results within the query model of [6] and their algorithm makes $O(\log^2(1/\epsilon)/\epsilon^7)$ queries for accuracy ϵn^r.

References

[1] G. Andersson and L. Engebretsen, *Property testers for dense constraint satisfaction programs on finite domains*, Random Structures and Algorithms 21 (2002), 14-32.

[2] S.Arora, D.Karger and M.Karpinski, *Polynomial time approximation schemes for dense instances of NP-hard problems*, Proceedings of the 27th Annual ACM Symposium on Theory of Computing (1995) 284-293; the full paper appeared in J. Comp. System. Sci. **58** (1999), 193-210.

[3] W.Fernandez-de-la-Vega, *MAX-CUT has a Randomized Approximation Scheme in Dense Graphs*, Random Structures and Algorithms 8 (1996) 187-199.

[4] A.M.Frieze and R.Kannan, *The Regularity Lemma and approximation schemes for dense problems*, Proceedings of the 37th Annual IEEE Symposium on Foundations of Computing, (1996) 12-20.

[5] A. M. Frieze and R. Kannan *Quick Approximation to matrices and applications*, Combinatorica **19** (2) (1999) pp 175-200.

[6] O.Goldreich, S.Goldwasser and D.Ron, *Property testing and its connection to learning and approximation*, Proc. 37th IEEE FOCS 96, 339-348; the full paper appeared in J. ACM **45** (1998), 653-750.

Approximation Hardness of Short Symmetric Instances of MAX-3SAT

Marek Karpinski

(joint work with Piotr Berman and Alex D. Scott)

Abstract. We prove approximation hardness of *short symmetric* instances of MAX-3SAT in which each literal occurs *exactly twice*, and each clause is exactly of size 3. We display also an explicit approximation lower bound for that problem. The bound *two* on the number of occurrences of literals in symmetric MAX-3SAT is thus the *smallest* possible bound for the MAX-3SAT *hardness gap property* to exist and making the instances hard to approximate.

We define a *symmetric (balanced) MAX-(3,Bk)-SAT* instance of the *maximizing* MAX-3SAT problem as a set of clauses of size exactly 3, in which every literal occurs exactly k times. MAX-(3,k)-SAT stands for the set of *relaxed* (possibly unbalanced) instances of MAX-3SAT in which every variable occurs exactly k times and each clause is of size exactly 3. We will also denote by (3,Bk)-SAT and (3,k)-SAT the corresponding sets of formulas.

It was proven in [BKS03] that MAX-(3,4)-SAT is hard to approximate to within a certain constant. It was also shown that the *balanced* MAX-(3,B3)-SAT is hard

to approximate [F98], [FLT02]. It remained an open question on whether, in fact, the balanced class MAX-(3,B2)-SAT remains hard to approximate. Because MAX-(3,4)-SAT is the smallest, with respect to the occurrence number, class of instances which are still inapproximable, the *balanced bound* 2 (B2), would be then the best possible.

In this paper we answer this question, and prove somewhat surprisingly that MAX-(3,B2)-SAT is, in fact, hard to approximate to within a certain constant. We display also an explicit factor for the approximation hardness of that problem. The bound 2 for the number of occurrences of literals is thus the smallest bound for symmetric MAX-3SAT for which the approximation gap property is still NP-hard (see for the applications of regular and symmetric 3SAT gap properties towards other lower approximation bounds in [ALMSS98], [F98], and [FLT02]).

We note also, that, interestingly, a dual version of this balanced satisfiability problem leads to a certain natural problem studied in graph theory. Let C be the set of clauses and $V = \{v_1, \ldots, v_n\}$ the set of boolean variables. For each $v_i \in V$, let e_i be the pair of clauses in which v_i occurs without negation, and let f_i be the pair of clauses in which v occurs negated. Thus if we set v_i true then we satisfy both clauses in e_i and if we set v_i false we satisfy both clauses in f_i. Now consider the graph $G_{V,C}$ with vertex set C and edges $e_1, f_1, \ldots, e_n, f_n$. Finding a satisfying assignment for (V, C) is equivalent to choosing one edge from each pair $\{e_i, f_i\}$ such that the resulting subgraph of $G_{V,C}$ has no isolated vertices (or, equivalently, finding a spanning forest of $G_{V,C}$ with no isolated vertices and at most one edge from each pair). We remark that a problem of similar type, where the edges come in pairs but we instead attempt to choose one edge from each pair without creating a giant component, has been considered by Bohman, Frieze and Wormald [BFW03] (see also [BF01]).

We follow in this paper a line of [BKS03] of constructing efficient enforcers for the boolean variables; however, in our present setting we have to produce resulting *balanced* unsatisfiable (3,B2)-SAT formulas. In fact, at that time the existence of such balanced and unsatisfiable formulas was an open question in the area.

We give here the first construction of balanced *enforces*, and a resulting balanced unsatifiable (3,B2)-SAT formula. Later we show how to transform the existence of balanced enforcers and unsatisfiable (3,B2)-SAT formulas into the NP-hardness result in exact setting. Finally, we prove our main result on approximation hardness of MAX-(3,2B)-SAT and give an explicit approximation lower bound and a gap property.

References

[ALMSS98] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, *Proof Verification and the Hardness of Approximation Problems*, Journal of the ACM **45(3)** (1998) pp. 501 - 555.

[BF01] T. Bohman and A. M. Frieze, *Avoiding a Giant Component*, Random Structures and Algorithms **19** (2001), pp. 75–85.

[BFW03] T. Bohman, A. M. Frieze, and N. C. Wormald, *Avoiding a Giant Component II*, to appear.

[B88] B. Bollobás, *The Isoperimetric Number of Random Graphs*, Europ. J. Combinatorics **9** (1988), pp. 241 - 244.

[BK03] P. Berman and M. Karpinski, *Improved Approximation Lower Bounds on Small Occurrence Optimization*, Elec. Coll. on Comp. Compl., ECCC TR03-008 (2003).

[BK01] P. Berman and M. Karpinski, *Efficient Amplifiers and Bounded Degree Optimization*, Elec. Coll. on Comp. Compl., ECCC TR01-053 (2001).

[BKS03] P. Berman, M. Karpinski and A. D. Scott, *Approximation Hardness and Satisfiability of Bounded Occurrence Instances of SAT*, Elec. Coll. on Comp. Compl., ECCC TR03-022 (2003).

[F98] U. Feige, *A Threshold of* $\ln n$ *for Approximating Set Cover*, *Journal of ACM*, **45(4)** (1998), pp.634–652.

[FLT02] U. Feige, L. Lovász and P. Tetali, *Approximating MIN-SUM Set Cover*, Proc. 5th APPROX (2002), LNCS Vol. 2462 (2002), pp. 94-107.

[H97] J. Håstad, *Some Optimal Inapproximability Results*, Proc. 29th ACM STOC, 1-10, 1997.

[T84] C. A. Tovey, *A Simplified Satisfiability Problem*, Discr. Appl. Math. **8** (1984), pp. 85 - 89.

[S78] T. J. Schaefer, *The Complexity of Satisfiability Problems*, Proc. 10th ACM STOC (1978), pp. 216 - 226.

On Multicuts and Related Problems

Michael Langberg
(joint work with Adi Avidor)

Abstract

In this work, we define and study a natural generalization of the multicut and multiway cut problems: the *minimum multi-multiway cut* problem. The input to the problem is a weighted undirected graph $G = (V, E)$ and k sets $S_1, S_2, \ldots, S_k$ of vertices. The goal is to find a subset of edges of minimum total weight whose removal completely disconnects each one of the sets $S_1, S_2, \ldots, S_k$, i.e., disconnects every pair of vertices u and v such that $u, v \in S_i$, for some i. This problem generalizes both the multicut problem, when $|S_i| = 2$, for $1 \leq i \leq k$, and the multiway cut problem, when $k = 1$.

We present an approximation algorithm for the multi-multiway cut problem with an approximation ratio which matches that obtained by Garg, Vazirani, and Yannakakis [GVY96] on the standard multicut problem. Namely, our algorithm has an $O(\log 2k)$ approximation ratio. Moreover, we consider instances of the minimum multi-multiway cut problem which are known to have an optimal solution of *light* weight. We show that our algorithm has an approximation ratio substantially better than $O(\log 2k)$ when restricted to such "light" instances. Specifically, we obtain an $O(\log LP)$-approximation algorithm for the problem, when all edge weights are at least 1, where LP is the value of a natural LP-relaxation of the problem. The latter improves the $O(\log LP \log\log LP)$ approximation ratio for the minimum multicut problem (implied by the work of Seymour [Sey95] and Even *et al.* [ENSS98]).

Introduction

The input to the *minimum multicut* problem is an undirected graph $G = (V, E)$ with a weight (or cost) function $w : E \to R^+$ defined on its edges, and a collection $(s_1, t_1), \ldots, (s_k, t_k)$ of vertex pairs. The objective is to find a subset of edges of minimum total weight whose removal disconnects s_i from t_i, for every $1 \leq i \leq k$. The problem is known to be APX-hard ([DJP$^+$94]). An $O(\log k)$-approximation algorithm for the problem was obtained by Garg, Vazirani and Yannakakis [GVY96].

The *minimum multiway cut* problem is a subproblem of the minimum multicut problem. The input consists of a weighted undirected graph $G = (V, E)$, as in the multicut problem, and a set $\{t_1, t_2, \ldots, t_k\}$ of vertices. The goal is to find a subset of edges of minimum total weight whose removal disconnects t_i from t_j, for every $1 \leq i < j \leq k$. The problem is also known to be APX-hard ([DJP$^+$94]). A $(\frac{3}{2} - \frac{1}{k})$-approximation algorithm for the problem was obtained by Calinescu, Karloff and Rabani [CKR98]. An improved $(1.3438 - \varepsilon_k)$-approximation algorithm for the problem was obtained by Karger *et al.* [KKS$^+$99]. In particular, for $k = 3$ the algorithm of Karger *et al.* [KKS$^+$99] achieves an approximation ratio of 12/11, which matches the integrality gap of the linear programming relaxation. This result was also obtained independently by Cunningham and Tang [CT99].

In this work, we define and study a natural generalization of both the multicut and multiway cut problems: the *minimum multi-multiway cut* problem. The input of the minimum multi-multiway cut problem consists of an undirected graph $G = (V, E)$ with a weight function $w : E \to R^+$ defined on its edges, and k *sets* of vertices $S_1, S_2, \ldots, S_k$ (also referred to as groups). The goal is to find a subset of edges of minimum total weight whose removal disconnects, for every $1 \leq i \leq k$, every two vertices $u, v \in S_i$. When $|S_i| = 2$, for all $1 \leq i \leq k$, the minimum multi-multiway cut problem is exactly the minimum multicut problem, and when $k = 1$, the minimum multi-multiway cut problem is the minimum multiway cut problem.

The minimum multicut problem

The minimum multicut problem (and its relation to multicommodity flow) have been extensively studied during the last few decades. The problem in which $k = 1$ is the standard $s - t$ cut problem, and is known to be solved exactly in polynomial time [FF56]. The case in which $k = 2$ was also shown to be polynomially solvable by Yannakakis *et al.* [YKCP83] using multiple applications of the max-flow algorithm. For any $k \geq 3$ the problem was proven to be APX-hard by Dahlhaus *et al.* [DJP$^+$94] and thus cannot permit a PTAS (unless P=NP).

The currently best known approximation ratio for the minimum multicut problem is obtained in the work of Garg, Vazirani, and Yannakakis [GVY96]. They present a polynomial algorithm that, given a graph G and a set of k pairs of vertices, finds a multicut of weight at most $O(\log k)$ times the optimal multicut in G. Their algorithm is based on a natural linear programming relaxation of the minimum multicut problem and has the following outline. By solving the relaxation, a *fractional* multicut of the given graph G is obtained. It can be seen that this

fractional solution implies a semi-metric on the vertices of G. This semi-metric is now used to round the fractional multicut into an integral one. Namely, the so called *region growing* scheme (introduced by Leighton and Rao [LR99] and used also by Klein *et al.* [KRAR95]) is applied to define for each pair (s_i, t_i) a region, *i.e.* a subset of vertices, which are in this case a *ball* of a specific radius centered at s_i. The multicut obtained by the algorithm is now defined as all edges in E with are cut by one of the defined regions.

Several results in the field of approximation algorithms have been inspired by the region growing technique for rounding the solution of linear programs. These include applications of the divide and conquer paradigm (see for example a survey by Shmoys [Shm96]), the design of approximation algorithms for the minimum multicut problem on directed graphs [KPRT93, ENSS98, CKR01, Gup03], and the results recently obtained for the minimum correlation clustering problem [DI03, CGW03].

In this work we study the region growing rounding technique when applied to the multi-multiway cut problem.

Our results

In this work we present two main results. First, we present an approximation algorithm for the multi-multiway cut problem with an approximation ratio which matches that obtained by [GVY96] on the standard multicut problem. Namely, our algorithm has an $O(\log k)$ approximation ratio. Our algorithm solves a natural linear programming relaxation of the multi-multiway cut problem, and rounds the fractional solution obtained using an *enhanced* region growing technique. Roughly speaking, the region growing technique used in this work differs from that used in previous works as in our case multiple regions are grown in a simultaneous manner rather than one by one.

Secondly, we consider instances to the minimum multi-multiway cut problem which are known to have an optimal solution of *light* weight. Denote such instances as light instances. We show that our algorithm has an approximation ratio substantially better than $O(\log k)$ when restricted to such light instances. Considering the connection between the multi-multiway cut problem and the closely related minimum uncut problem, we show that our result on light instances of minimum multi-multiway cut imply a result of independent interest on the minimum uncut problem. Our results can be summarized as follows.

Theorem 17 (General multi-multiway cuts)**.** *There exists a polynomial time algorithm which approximates the minimum multi-multiway cut problem within an approximation ratio of* $4\ln(k+1)$.

Theorem 18 (Light multi-multiway cuts)**.** *Let I be an instance to the minimum multi-multiway cut problem. Let Opt_I be the weight of the optimal multi-multiway cut of instance I. If $w(e) \geq 1$ for all $e \in E$, then one can approximate the minimum multi-multiway cut problem on I within an approximation ratio of* $4\ln(2Opt_I)$.

Corollary 3 (Light minimum uncut)**.** *If an undirected graph $G = (V, E)$ can be made bipartite by the deletion of k edges, then a set of $O(k \log k)$ edges whose deletion makes the graph bipartite can be found in polynomial time.*

References

[CGW03] M. Chrikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. *In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003.

[CKR98] G. Calinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *In Proceedings of the 30th Ann. ACM Symp. on Theory of Comp., ACM-SIAM*, 1998.

[CKR01] J. Cheriyan, H. J. Karloff, and Y. Rabani. Approximating directed multicuts. *In Proceedings of IEEE Symposium on Foundations of Computer Science*, 2001.

[CT99] W. H. Cunningham and L. Tang. Optimal 3-terminal cuts and linear programming. In *Proceedings of IPCO (Lecture Notes in Computer Science 1610)*, pages 114–125, 1999.

[DI03] E. D. Demaine and N. Immorlica. Correlation clustering with partial information. *In Proceedings of APPROX (Lecture Notes in Computer Science 2764)*, 2003.

[DJP+94] R. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[ENSS98] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[Gup03] A. Gupta. Improved results for directed multicut. *In Proceedings 14th Ann. ACM-SIAM Symp. on Discrete Algorithms, ACM-SIAM*, pages 454–455, 2003.

[GVY96] N. Garg, V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.

[KKS+99] D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *In Proceedings of ACM Symposium on the Theory of Computing*, pages 668–678, 1999.

[KPRT93] P. N. Klein, S. A. Plotkin, S. Rao, and S. E. Tardos. New network decompositions theorems with applications. *Brown University Technical Report CS-93-30*, 1993.

[KRAR95] P. Klein, S. Rao, A. Agrawal, and R. Ravi. An approximate max-flow min-cut relation for undirected multicommodity flow, with applications. *Combinatorica*, 15:187–202, 1995.

[LR99] F. T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[Sey95] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.

[Shm96] D. Shmoys. Cut problems and their applications to divide-and-conquer. *In D. Hochbaum, editor, Approximation Algorithms for NP-Hard Problems, PWS Publishing*, pages 192–235, 1996.

[YKCP83] M. Yannakakis, P. C. Kanellakis, S. C. Cosmadakis, and C. H. Papadimitriou. Cutting and partitioning a graph after a fixed pattern. *In Proceedings of ICALP (Lecture Notes in Computer Science 154)*, pages 712–722, 1983.

A PTAS for the minimization of polynomials of fixed degree over the simplex

Monique Laurent
(joint work with Etienne de Klerk and Pablo Parrilo)

Polynomial optimization over the simplex

We consider the problem of minimizing a polynomial $p(x)$ of degree d over the standard simplex $\Delta := \{x \in \mathbb{R}^n_+ \mid \sum_{i=1}^n x_i = 1\}$; that is, the problem of computing

$$p_{\min} := \min_{x \in \Delta} p(x). \tag{0.1}$$

One may assume w.l.o.g. that $p(x)$ is a homogeneous polynomial (form). Indeed, as observed in [2], if $p(x) = \sum_{\ell=0}^d p_\ell(x)$, where $p_\ell(x)$ is homogeneous of degree ℓ, then minimizing $p(x)$ over Δ is equivalent to minimizing the degree d form $p'(x) := \sum_{\ell=0}^d p_\ell(x)(\sum_{i=1}^n x_i)^{d-\ell}$. Problem (0.1) is an NP-hard problem, already for forms of degree $d = 2$, as it contains the maximum stable set problem. Indeed, for a graph G with adjacency matrix A, the maximum size $\alpha(G)$ of a stable set in G can be expressed as

$$\frac{1}{\alpha(G)} = \min_{x \in \Delta} x^T(I + A)x$$

by the theorem of Motzkin and Straus [3]. In this paper we show that problem (0.1) has a polynomial approximation scheme (PTAS) when restricted to the class of polynomials having a fixed degree $d \geq 2$.

Given an integer $k \geq 1$, let $\Delta(k) := \{x \in \Delta \mid kx \in \mathbb{Z}^n\}$ denote the set of rational points in Δ with denominator k and define

$$p_{\Delta(k)} := \min p(x) \text{ s.t. } x \in \Delta(k).$$

Thus, $p_{\min} \leq p_{\Delta(k)}$ for any $k \geq 1$. As $|\Delta(k)| = \binom{n+k-1}{k}$, one can compute the bound $p_{\Delta(k)}$ in polynomial time for any *fixed* k. We can prove some estimates on the quality of the approximation $p_{\Delta(k)}$. We need some notation. For a polynomial $p(x) = \sum_\alpha p_\alpha x^\alpha$, set

$$p_{\max} := \max_{x \in \Delta} p(x), \quad p^{(0)}_{\max} := \max_\alpha p_\alpha \frac{\alpha_1! \cdots \alpha_n!}{|\alpha|!}.$$

One can verify that $p^{(0)}_{\max}$ is equal to the smallest scalar λ for which the polynomial $\lambda(\sum_{i=1}^n x_i)^d - p(x)$ has nonnegative coefficients. Therefore, $p_{\max} \leq p^{(0)}_{\max}$. We show the inequalities:

$$p_{\Delta(k)} - p_{\min} \leq \frac{d(d-1)}{2k}(p^{(0)}_{\max} - p_{\min}), \tag{0.2}$$

$$p^{(0)}_{\max} - p_{\min} \leq \binom{2d-1}{d} d^d (p_{\max} - p_{\min}), \tag{0.3}$$

which together imply:

$$p_{\Delta(k)} - p_{\min} \leq \frac{d^d}{k}\binom{d}{2}\binom{2d-1}{d}(p_{\max} - p_{\min}). \tag{0.4}$$

The last inequality (0.4) shows that the bounds $p_{\Delta(k)}$ provide a PTAS for the optimization problem (0.1) on the class of polynomials with *fixed* degree d.

A more detailed analysis permits to show sharper estimates in some cases. For instance, when $d = 2$,

$$p_{\Delta(k)} - p_{\min} \leq \frac{1}{k}(max_{i=1,\ldots,n} p(e_i) - p_{\min})$$

shown earlier by Bomze and de Klerk [1] and, when $d = 3$, we can show that

$$p_{\Delta(k)} - p_{\min} \leq \frac{4}{k}(p_{\max} - p_{\min}).$$

Sketch of proof

Our argument for (0.2) follows closely the proof given by Powers and Reznick [6] for the following theorem of Pólya. (The bound on r comes essentially from [6].)

Theorem 19. *Let p be a form of degree d which is positive on the simplex Δ, i.e., $p_{\min} > 0$. Then, the polynomial $(\sum_{i=1}^n x_i)^r p(x)$ has nonnegative coefficients for all $r \geq \binom{d}{2}\frac{p^{(0)}_{\max}}{p_{\min}} - d$.*

Pólya's result can be used for constructing an asymptotically converging hierarchy of lower bounds for $p_{\min}$. As $p_{\min}$ is equal to the largest scalar λ for which the polynomial $p(x) - \lambda\,(\sum_{i=1}^n x_i)^d$ is nonnegative over $\mathbb{R}^n_+$, the parameter:

$$p^{(r)}_{\min} := \max\ \lambda \text{ s.t.} \quad \text{the polynomial } (\textstyle\sum_{i=1}^n x_i)^r \left(p(x) - \lambda\,(\sum_{i=1}^n x_i)^d\right) \text{ has nonnegative coefficients}$$

is a lower bound for $p_{\min}$ for any integer $r \geq 0$. Moreover, $\lim_{r\to\infty} p^{(r)}_{\min} = p_{\min}$ by Pólya's theorem. It turns out that the quality of the two hierarchies of bounds $p_{\Delta(r+d)}$ and $p^{(r)}_{\min}$ can be simultaneously analyzed via the proof of Pólya's theorem.

Our proof for (0.3) uses some tools of Reznick [7] about powers of linear forms. It can be sketched as follows.

As $p^{(0)}_{\max} - p_{\min} \leq p^{(0)}_{\max} - p^{(0)}_{\min}$, it suffices to bound $p^{(0)}_{\max} - p^{(0)}_{\min}$ in terms of $p_{\max} - p_{\min}$. Define the vectors $y := (p(\alpha))_{\alpha\in\mathbb{Z}^n_+,|\alpha|=d}$ and $z := (p_\alpha \frac{\alpha!}{d!})_{\alpha\in\mathbb{Z}^n_+,|\alpha|=d}$. Thus, $p^{(0)}_{\max} - p^{(0)}_{\min} = z_{\max} - z_{\min}$, where $z_{\max}$ (resp., $z_{\min}$) denotes the maximum (resp., minimum) coordinate of z, and $y_{\max} - y_{\min} \leq p_{\max} - p_{\min}$. The key is now to observe that y, z are related by a linear equation: $z = Ay$, for some matrix A satisfying $Ae = d^{-d}e$ (e is the all ones vector). (Such matrix A can be found by considering an alternative basis for the space of degree d forms in n variables; A being then the matrix permitting to express the new basis in the terms of the

original basis $\{x^\alpha \frac{d!}{\alpha!} \mid \alpha \in \mathbb{Z}^n_+, |\alpha| = d\}$.) Define $\|A\|_\infty := \max_i \sum_j |A(i,j)|$. As $z = Ay$ and A has constant row sums, it follows that

$$z_{\max} - z_{\min} \leq \|A\|_\infty (y_{\max} - y_{\min}).$$

The final step consists in bounding $\|A\|_\infty$ in terms of the constant $\binom{2d-1}{d}$.

Conclusions

As observed by Nesterov [4], results for the simplex can be extended to optimization over an arbitrary polytope $P := \mathrm{conv}(u_1, \ldots, u_N)$, where $u_1, \ldots, u_N \in \mathbb{R}^n$. For an integer $k \geq 1$, one can define the grid approximation:

$$p_{P(k)} := \min_{x \in \Delta(k)} p\left(\sum_{i=1}^N x_i u_i\right)$$

where the simplex Δ now lies in the N-space (N is the number of vertices of P). The bounds obtained earlier translate into bounds for $p_{P(k)}$. For instance, when $p(x)$ has degree 2,

$$p_{P(k)} - p_{\min} \leq \frac{1}{k}\left(\max_{i=1,\ldots,N} p(u_i) - p_{\min}\right).$$

However, the complexity of computing the parameter $p_{P(k)}$ depends on the number N of vertices, which can be exponentially large in terms of the number n of variables.

Note that maximizing a quadratic form over the cube $[-1,1]^n$ is NP-hard and no PTAS can exist, since it contains the max-cut problem.

The complexity of approximating a polynomial over the unit sphere is not known. Of course, minimizing a quadratic form over the unit sphere is an easy problem, as it amounts to computing the minimum eigenvalue of a matrix. Moreover, minimizing an *even* form on the unit sphere has a PTAS, since it can be reformulated as the problem of minimizing an associated form on the simplex. On the other hand, Nesterov [4] shows that maximizing a cubic form on the unit sphere is a NP-hard problem, using a reduction from the maximum stable set problem.

References

[1] I. Bomze and E. De Klerk. Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Global Optimization*, 24(2):163–185, 2002.

[2] L. Faybusovich. Global optimization of homogeneous polynomials on the simplex and on the sphere. In C. Floudas and P. Pardalos, editors, *Frontiers in Global Optimization*. Kluwer Academic Publishers, 2003.

[3] T.S. Motzkin and E.G. Straus. Maxima for graphs and a new proof of a theorem of Túran. *Canadian J. Math.*, 17:533-540, 1965.

[4] Yu. Nesterov. Random walk in a simplex and quadratic optimization over convex polytopes. CORE Discussion Paper 2003/71, CORE-UCL, Louvain-La-Neuve, September 2003.

[5] G. Pólya. *Collected Papers*, volume 2, pages 309–313. MIT Press, Cambridge, Mass., London, 1974.
[6] V. Powers and B. Reznick. A new bound for Pólya's theorem with applications to polynomials positive on polyhedra. *Journal of Pure and Applied Algebra*, 164:221–229, 2001.
[7] B. Reznick. Sums of even powers of real linear forms. *Memoirs of the American Mathematical Society*, Number 463, 1992.

Boosted Sampling: Approximation Algorithms for Stochastic Optimization

R. Ravi
(joint work with Anupam Gupta and Amitabh Sinha)

Infrastructure planning problems involve making decisions under uncertainty about future requirements; while more effective decisions can be made after the actual set of clients have materialized, the decision-making costs are inflated if deferred until then. The following simple two-stage model captures this tradeoff effectively. Future requirements are uncertain, but are assumed to be drawn from a known probability distribution (e.g., from demand forecasts, industry outlooks). In light of this information, an *anticipatory* part of the solution may be constructed in a first-stage at the current costs. Subsequently, the requirements facing the planner materialize in the form of a client set (drawn from the distribution), and the first-stage solution must be *augmented* to satisfy the revealed requirements. The elements chosen in this second stage are costlier than when chosen earlier, reflecting the need for careful (first-stage) planning. Given the uncertainty of the requirements, the traditional minimum-cost goal may be adapted to minimize the total *expected* cost of the solution.

As an example, consider the Stochastic Steiner tree problem that specifies an inflation parameter σ and a probability distribution $p_{\min}^{(r)}$ on the set of terminal nodes (which are clients) that have to be connected to the root in a given rooted discrete metric space. A subset of edges E_0 may be bought by paying the original lengths in the first stage. Once the actual set of terminals S is revealed, we must then buy the recourse edges E_S at σ times their lengths so that S is connected to the root by edges in $E_0 \cup E_S$. The objective is to minimize $c(E_0) + \mathbf{E}[\sigma\, c(E_S)]$. Here the expectation is over $p_{\min}^{(r)}$, the randomness in the set of terminals revealed.

The framework is that of *two-stage stochastic optimization with recourse* [7, 6, 9] which may be paraphrased as "On Monday, we only know the input distribution on the clients, and we can buy some resources. On Tuesday, the client set is now completely specified, but things are now more expensive (in our case, by a factor σ); we must buy any additional resources needed to get a feasible solution to the instance."

Following this framework, in Stochastic Vertex Cover, the clients are edges to be covered, and we are given a probability distribution over sets of edges that will arrive; vertices become σ times more expensive after these edges are revealed. The Stochastic Facility Location problem on a metric space containing clients

Problem	Non-Stoc. Approx. Ratio α	Strictness β	Stoch. Approximation	
			General Distrib.	Indep. decisions
Steiner Tree	1.55 (Robins & Zelikovsky)	2	3.55	3.55
Vertex Cover	2 (Primal-dual)	6	8	3
Facility Location	3 (Mettu-Plaxton)	5.45	8.45	6
Steiner Network	4 (Gupta et al.)	$4^{\ddagger}$	-	8

FIGURE 0.1. Result Summary

and facilities with opening costs defines a probability distribution over the set of clients that will require connection to open facilities. Opening facilities becomes σ times costlier in the second stage. The objective, in addition to expected cost of opening facilities, also includes expected connection costs of the revealed clients to their closest open facilities.

Our Results: In the paper [3] on which the talk is based, we give a simple yet general framework to translate approximation algorithms for deterministic optimization problems into approximation algorithms for corresponding stochastic versions with second-stage inflation parameter σ. Given an α-approximation algorithm for the classical problem, one can use it in the following framework:

(1) *Boosted Sampling:* Sample σ times from the distribution $p_{\min}^{(r)}$ to get sets of clients $D_1, \ldots, D_\sigma$.
(2) *Building First Stage Solution:* Build an α-approximate solution for the clients $D = \cup_i D_i$.
(3) *Building Recourse:* When actual future in the form of a set S of clients appears (with probability $p_{\min}^{(r)}(S)$), augment the solution of Step 2 to a feasible solution for S.

Note that *we do not need to know the distribution* $p_{\min}^{(r)}$ *explicitly*; it could be a black-box from which we can draw samples. (In practice, these samples could be drawn from market predictions, or from Monte-Carlo simulations.) Thus we can sidestep the often-lethal problem of handling an exponential number of scenarios.

Informal Main Result If the α-approximation algorithm $\mathcal{A}$ satisfies some technical properties (the problem is *sub-additive*, and $\mathcal{A}$ admits a β-strict *cost-sharing* function, then the above framework yields an $\alpha + \beta$ approximation for the stochastic version of the problem.

Using our framework, we show that stochastic variants of STEINER TREE, FACILITY LOCATION, and VERTEX COVER have constant-factor approximation algorithms. The approximation ratios α and strictness of the corresponding cost-shares β, and the resulting guarantees for the stochastic variants are summarized in Figure 0.1.

We also consider the special case of *independent decisions*; in this, each client j has a probability π_j of arrival *independent* of other clients, and the probability $\pi(S)$ of the set S materializing is given by $\prod_{j \in S} \pi_j \prod_{j \notin S} (1 - \pi_j)$. For this model, we can also give a 8-approximation for the stochastic version of the Steiner Forest problem and improve the approximation ratios of the corresponding versions of Vertex Cover and Facility location to 3 and 6 respectively.

While a natural approach to utilizing an approximation algorithm for a deterministic problem is to set the client requirements at their expected value according to $p^{(r)}_{\min}$, we note that this approach cannot yield bounded approximation ratios even in simple examples. Rather, using the full power of sampling in building the first stage solution gives a provably good solution as we demonstrate.

Related Work: Very recently, there has been a surge of interest in stochastic versions of NP-hard problems, with papers on the topic by Ravi and Sinha [8], and independently, by Immorlica et al. [4]. Both these works look at versions of our model with some restrictions on the distribution $p^{(r)}_{\min}$, while we consider *arbitrary distributions* $p^{(r)}_{\min}$. In particular, they consider the following cases.

- the *scenario model*, where the distribution $p^{(r)}_{\min}$ has its support on a family $\mathcal{F}$ of possible subsets explicitly given as part of the input (and hence the algorithms are allowed to take time $\text{poly}(|\mathcal{F}|, n)$), and
- the *independent decisions* model, where each element j has an associated probability π_j, and the probability of a set $\pi(S) = \prod_{j \in S} \pi_j \prod_{j \notin S} (1 - \pi_j)$. (I.e., the sets are chosen by flipping a coin independently for each element.)

Since our algorithms work for *arbitrary distributions*, our theorems hold in both the above models as well. In particular, our 3.55- and 3-approximations for stochastic Steiner Tree and Vertex Cover in the independent model improve upon the $O(\log n)$- and 6.3-approximations in [4] respectively.

One can define (as in [8]) other stochastic variants of the problems we define here: e.g., one can imagine that there are multiple inflation parameters, and that instead of all things getting dearer by σ, different parts of the problem change in different ways. This work leaves open the question of whether our framework can be extended to handle such multiple-parameter stochastic problems.

Stochastic Steiner Tree appears similar to the *maybecast* problem of Karger and Minkoff [5]; however, the latter is a single-stage optimization problem. Finally, though some of our techniques, including strict cost-shares come from the work of [2, 1], the problems considered there are deterministic optimization problems.

References

[1] Anupam Gupta, Amit Kumar, Martin Pál, and Tim Roughgarden. Approximations via cost-sharing. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 606–615, 2003.

[2] Anupam Gupta, Amit Kumar, and Tim Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 365–372, 2003.

[3] Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 417–426, 2004.
[4] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004. to appear.
[5] David R. Karger and Maria Minkoff. Building Steiner trees with incomplete global knowledge. In *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science*, pages 613–623, 2000.
[6] Willem K. Klein Haneveld and Maarten H. van der Vlerk. Stochastic integer programming: general models and algorithms. *Ann. Oper. Res.*, 85:39–57, 1999. Stochastic programming. State of the art, 1998 (Vancouver, BC).
[7] Willem K. Klein Haneveld and Maarten H. van der Vlerk. *Stochastic Programming*. Department of Econometrics and OR, University of Groningen, Netherlands, 2003.
[8] R. Ravi and Amitabh Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. *GSIA Working Paper 2003-E68*, 2003.
[9] R. Schultz, L. Stougie, and M. H. van der Vlerk. Two-stage stochastic integer programming: a survey. *Statist. Neerlandica*, 50(3):404–416, 1996.

Robust Inference of Relevant Attributes

R. Reischuk
(joint work with Jan Arpe)

Given n Boolean input variables representing a set of attributes, consider Boolean functions that actually depend only on a small but unknown subset of these variables/attributes, in the following called *relevant*. The goal is to determine the relevant attributes given a sequence of examples - input vectors X with their values $f(X)$. Determining the number of relevant attributes is NP-hard. It has been known that this problem can be approximated within a logarithmic factor with a reduction to set cover problem. This ratio is best possible under the assumption that NP is not contained in subexponential deterministic time.To find the relevant attributes we analyze two simple greedy strategies and prove that they are able to achieve this goal for various kind of Boolean functions and input distributions according to which the examples are drawn at random. The analysis also provides explicit upper bounds on the number of examples required which grow only logarithmically in the total number of attributes, but exponentially in the number of relevant attributes. They further depend on the distribution and combinatorial properties of the function to be inferred. Then we extend these results to a situation in which the examples contain errors, that means a certain number of input bits may be wrong. We show that even in such an error-prove situation, reliable inference is still possible.

Average Case Analysis: Two Seemingly Simple Problems

Angelika Steger
(joint work with Jan Remy and Alexander Souza)

Closely related to the development of algorithms is the analysis of algorithms. Here one distinguishes between worst case analysis, where the behavior of the algorithm on the worst possible input is considered, and average case analysis, where the expected or average behavior of the algorithm is of interest. From a practical point of view the latter is of particular importance if the worst case analysis does not provide bounds that are meaningful in practice: there are algorithms that provide empirically very good results, even though their worst case behavior is not very promising. The sorting algorithm QuickSort is a well known example for this phenomenon as there are algorithms that have a much better worst-case behavior, but on average over all possible inputs QuickSort performs very well, and that explains why it is one of the most used sorting algorithms in practice. However, at the current state of art, QuickSort is also one of the few examples where good bounds on the average performance can be proven. It seems that one needs to develop new methods and tools for a successful average case analysis in a broader context. In this view we consider two very simple deterministic problems, one in the area of scheduling and one in graph theory, and study their average case behavior.

1. Completion Time Scheduling

Several deterministic completion time scheduling problems can be solved in polynomial time. Smith [8] shows that scheduling jobs in order of non-decreasing processing time and weight ratio (WSPT) is optimal for the single machine problem $1 \,|\, | \sum_j w_j C_j$. For the unweighted problem $P \,|\, | \sum_j C_j$ on identical parallel machines, the optimality of the shortest processing time first (SPT) strategy is shown by Conway, Maxwell, and Miller [3]. In contrast, Bruno and Sethi [1] establish that the problem $Pm \,|\, | \sum_j w_j C_j$ is $\mathcal{NP}$-hard in the ordinary sense for constant $m \geq 2$ machines.

Turning to stochastic scheduling, the expected value of the objective function to a deterministic problem is a natural choice as an objective for the probabilistic counterpart. Thus, in that preformance measure, an algorithm ALG is considered optimal if it minimises the value $\mathcal{E}\mathrm{ALG} = \int_x \mathrm{ALG}(x) f(x) dx$, where $\mathrm{ALG}(x)$ denotes the value of the objective function achieved by ALG on an instance x with density $f(x)$.

Apparently, models that are $\mathcal{NP}$-hard in a deterministic setting sometimes allow a simple priority policy to be optimal for the probabilistic counterpart. For example, scheduling jobs in order of non-decreasing expected processing time (SEPT) is known to be optimal for many problems with the objective $\mathcal{E}\sum_j C_j$. Moreover, for the problem $1 \,|\, P_j \sim \mathrm{Stoch}\,(\mu_j) \,|\, \mathcal{E}\sum_j w_j C_j$, scheduling jobs in non-decreasing order of $\frac{\mu_j}{w_j}$ ratio (WSEPT) is optimal in non-preemptive static and dynamic policies. By using LP relaxations, Möhring, Schulz, and Uetz [5] have recently shown

that $\mathbb{E}[\text{WSEPT}] \leq (2-\frac{1}{m})\mathbb{E}[\text{OPT}]$, for several variants of the scheduling problem $P \mid P_j \sim \text{Stoch}(\mu_j) \mid \mathcal{E}\sum_j w_j C_j$, where OPT denotes an optimum policy.

One property of the performance measure $\mathbb{E}[\text{ALG}]$ is that instances x with small value $\text{ALG}(x)$ tend to be neglected since they contribute few to the overall expected value. Hence, in this measure, algorithms are preferred that perform well on instances x with large optimum value $\text{OPT}(x)$. It depends on the application if such behaviour is desireable, but if one is interested in algorithms that perform well on "many" instances, this measure may seem inappropriate.

Regarding this problem, the expected competitive ratio $\mathbb{E}[\text{ALG/OPT}]$ seems to be interesting, as the ratio $\text{ALG}(x)/\text{OPT}(x)$ relates the value of the objective function achieved by some algorithm ALG to the optimum OPT for all instances x. However, it seems that in the context of stochastic scheduling, the measure $\mathbb{E}[\text{ALG/OPT}]$ has only been considered by Coffman and Gilbert [2] and in the recent work by Scharbrodt, Schickinger, and Steger [6].

In [9] we introduce the class of distributions that are new-better-than-used in expectation *relative* to a function h (NBUE_h), which generalises the new-better-than-used in expectation (NBUE) class. The NBUE_{OPT} class comprises the exponential, geometric, and uniform distribution. Allowing the adversary to choose NBUE_{OPT} processing time distributions, we derive bounds to online list scheduling algorithms. Our analysis depends on a quantity α which is an upper bound to the probability for any pair of jobs being in the wrong order in a list of an arbitrary online list algorithm ALG, compared to an optimum list.

In particular, we show that the expected competitive ratio is at most $\frac{1}{1-\alpha}$ for the single machine problem and at most $\frac{1}{1-\alpha}+1-\frac{1}{m}$ for m identical parallel machines. These results reflect well the intuition that an algorithm should perform the better, the lower its probability of sequencing jobs in a wrong order. As a special case, we show that the WSEPT algorithm yields $\mathbb{E}\left[\frac{\text{WSEPT}}{\text{OPT}}\right] \leq 3-\frac{1}{m}$ for m identical parallel machines and exponential distributed processing times. Simulations empirically demonstrate tightness of this bound.

2. Minimum Spanning Tree

Let $G=(V,E)$ be a complete graph on n vertices which is *randomly weighted*, i.e. the edge weights are uniformly and independently chosen from $[0,1]$. Let the random variable OPT denote the weight of the minimum spanning tree in G. It has been shown by Frieze [4] that $\mathbf{E}[\text{OPT}]$ is independent of n. In particular, he proved that

$$\lim_{n\to\infty} \mathbf{E}[\text{OPT}] = \sum_{j=1}^{\infty} \frac{1}{j} = \zeta(3).$$

Let $N=\binom{n}{2}$ and assume that the edges are indexed by increasing weight. Frieze bounds the indices of the edges Kruskal's algorithm takes from above and below. Since the expected weight of the i^{th} smallest edge in G is $i/(N+1)$, the value given above follows by linearity of expectation. Note that this argument heavily relies on the fact that the whole graph is known.

Consider the following online version of the minimum spanning tree problem, described as a game between an algorithm and an adversary. Assume that some adversary $\mathcal{ADV}$ administrates the set E and shows edge by edge to an online algorithm $\mathcal{A}$. Without any precise knowledge about the weights of unseen edges, the algorithm has to decide immediately whether to include the edge into the spanning tree or not. Let ALG denote the weight of the spanning tree generated by the algorithm. Of course, the goal of $\mathcal{A}$ is to minimize ALG, which means we seek an online algorithm that competes well with the offline optimum. Analogously to the deterministic definition of c-competitiveness the *strict competitiveness coefficient* of $\mathcal{A}$ against $\mathcal{ADV}$ is defined as

$$C_{\mathcal{A}}^{\mathcal{ADV}} \stackrel{\text{def}}{=} \frac{\mathbf{E}[\text{ALG}]}{\mathbf{E}[\text{OPT}]}.$$

It remains to specify how powerful the adversary is. In the standard literature, two basic types of adversary models are used to analyze online algorithms in a probabilistic context. *Oblivious adversaries* determine the edge order in advance, as opposed to *adaptive adversaries* who choose the next edge based on the player's actions so far. An adversary $\mathcal{ADV}$ is said to be *fair* if he discovers the weight of the edges at the same time as the algorithm. In contrast, $\mathcal{ADV}$ is *unfair* if he knows the edge weights in advance and uses this information to determine the edge order. We thus consider the following four adversaries: $\mathcal{FO}$ (fair, oblivious), $\mathcal{FA}$ (fair, adaptive), $\mathcal{UO}$ (unfair, oblivious) and $\mathcal{UA}$ (unfair, adaptive). It follows by inclusion that for every algorithm $\mathcal{A}$,

$$C_{\mathcal{A}}^{\mathcal{FO}} \leq C_{\mathcal{A}}^{\mathcal{FA}} \quad \text{and} \quad C_{\mathcal{A}}^{\mathcal{UO}} \leq C_{\mathcal{A}}^{\mathcal{UA}}. \tag{0.1}$$

In [7] we give an algorithm which is strictly $\mathcal{O}(1)$-competitive against $\mathcal{FA}$. In addition, we show that there is an algorithm which is strictly $\mathcal{O}(\log n)$-competitive against $\mathcal{UA}$. Furthermore, we demonstrate that for every algorithm $\mathcal{A}$, we have lower bounds $C_{\mathcal{A}}^{\mathcal{UO}} = \Omega(\log n)$ and $C_{\mathcal{A}}^{\mathcal{FO}} \geq c \cdot \zeta(3)$ for a fixed constant $c > 1$. Then, the relation given in (0.1) assures that we have found for each of the four adversaries the best (in order of magnitude) possible online algorithm.

References

[1] E. C. Bruno, Jr. and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM*, 17:382 – 387, 1974.

[2] E. G. Coffmann, Jr. and E. N. Gilbert. Expected relative performance of list scheduling rules. *Oper. Res.*, 33:548–561, 1985.

[3] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling*. Addison–Wesley Publishing Company, Reading, MA, 1967.

[4] A. Frieze. On the value of a random minimum spanning tree. *Discrete Applied Mathematics*, 10:47–56, 1985.

[5] R.H. Möring, A.S. Schulz, and M. Uetz. Approximation in Stochastic Scheduling: The Power of LP-based Priority Rules. *Journal of the ACM*, 46:924 – 942, 1999.

[6] T. Schickinger, M. Scharbrodt, and A. Steger. A new average case analysis for completion time scheduling. *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 170 – 178, 2002.

[7] J, Remy, A. Souza, and A. Steger. On an online spanning tree problem in randomly weighted graphs. Manuscript, submitted for publication, 2004.
[8] W. E. Smith. Various optimizers for single–stage production. *Naval Research and Logistics Quaterly*, 3:59–66, 1956.
[9] A. Souza, and A. Steger. The expected competitive ratio for weighted completion time scheduling. *Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 620 – 631, 2004.

Typical Properties of Winners and Losers in Discrete Optimization

Berthold Vöcking
(joint work with Rene Beier)

Many combinatorial optimization problems have an objective function or constraints specified in terms of real numbers representing natural quantities like time, weight, distance, or utility. This includes some well-studied optimization problems like, e.g., traveling salesperson, shortest path, minimum spanning tree as well as various scheduling and packing problems. When analyzing the complexity of algorithms for such problems, we usually assume that these numbers are integers or rational numbers with a finite length representation. The hope is that it suffices to measure and compute with some bounded precision in order to identify an optimal or close to optimal solution. In fact, if real numbers occur only in the objective function and if this objective function is well-behaved (e.g., a linear function) then calculating with reasonable approximations of the input numbers yields a feasible solution whose objective value is at least close to the optimal objective value. More problematically, however, if the constraints are defined by real numbers, then calculating with rounded input numbers might miss all interesting solutions or might even produce infeasible solutions.

How can one solve optimization problems (efficiently) on a computer when not even the input numbers can be specified exactly? – In practice, optimization problems in which real numbers occur in the input are solved by simply rounding the real numbers more or less carefully. Fortunately, this approach seems to yield reasonable results. We seek for a theoretically founded explanation why this rounding approach usually works. Studying this issue under worst case assumptions does not make very much sense as, in the worst case, the smallest inaccuracy might lead to an infeasible or utterly sub-optimal solution. This question needs to be studied in a stochastic model. In our probabilistic analysis, we show that, under some reasonable and quite general stochastic assumptions, one can usually round real-valued input numbers after only a logarithmic number of bits without changing the optimal solution. In fact, our probabilistic analysis goes far beyond the point of explaining phenomena occurring in practice. We are able to provide algorithms with polynomial average-case complexity (more precisely, polynomial smoothed complexity) for a quite general class of discrete optimization problems.

Our probabilistic analysis covers a large class of combinatorial optimization problems containing, e.g., all *binary optimization problems* defined by linear constraints and a linear objective function over $\{0,1\}^n$. By parameterizing which

constraints are of stochastic and which are of adversarial nature, we obtain a semi-random input model that enables us to do a general average-case analysis for a large class of optimization problems while at the same time taking care for the combinatorial structure of individual problems. Our analysis covers various probability distributions for the choice of the stochastic numbers and includes *smoothed analysis* with Gaussian and other kinds of perturbation models as a special case (cf. [2, 3, 4, 5, 7, 8]). In fact, we can exactly characterize the smoothed complexity of optimization problems in terms of their random worst-case complexity.

> A binary optimization problem has a *polynomial smoothed complexity* if and only if it has a pseudopolynomial complexity.

Our analysis is centered around structural properties of binary optimization problems, called *winner*, *loser*, and *feasibility gaps*. Using similar techniques as for the well-known Isolating Lemma [6] we show, when the coefficients of the objective function and/or some of the constraints are stochastic, then there usually exist a polynomial $n^{-\Omega(1)}$ gap between the best and the second best solution as well as a polynomial slack to the boundary of the constraints. Similar to the condition number for linear programming, these gaps describe the sensitivity of the optimal solution to slight perturbations of the input and can be used to bound the necessary accuracy as well as the complexity for solving an instance. We exploit the gaps in form of an adaptive rounding scheme increasing the accuracy of calculation until the optimal solution is found. The strength of our techniques is illustrated by applications to various NP-hard optimization problems from mathematical programming, network design, and scheduling for which we obtain the the first algorithms with polynomial average-case/smoothed complexity.

References

[1] C. Banderier, R. Beier and K. Mehlhorn. Smoothed Analysis of three combinatorial problems. *Proc. 28th Intern. Symp. on Math. Foundations of Computer Science, 198-207, 2003.*

[2] A. Blum and J. Dunagan. Smoothed Analysis of the Perceptron Algorithm. *Proc. of the 13th Annual ACM-SIAM Symp. on Discrete Algorithms*, 905-914, 2003.

[3] L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, G. Schäfer and T. Vredeveld. Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm. *Proc. of the 44th IEEE Symp. on Foundations of Computer Science, 462-471, 2003.*

[4] J. Dunagan, D. A. Spielman, and S.-H. Teng. Smoothed analysis of Renegar's condition number for linear programming. Available at http://arxiv.org/abs/cs.DS/0302011, 2002.

[5] V. Damerow, F. Meyer auf der Heide, H. Räcke, C. Scheideler and C. Sohler. Smoothed motion complexity. *Proc. of the 11th European Symp. on Algorithms, 2003.*

[6] K. Mulmuley, U. Vazirani and V.V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica 7(1):105-113, 1987.*

[7] D. A. Spielman and S.-H. Teng. Smoothed Analysis of algorithms: why the Simplex Algorithm usually takes polynomial time. *Proc. of the 33rd ACM Symp. on Theory of Computing*, 296-305, 2001.

[8] D. A. Spielman and S.-H. Teng. Smoothed Analysis: Motivation and discrete models. *Proc. of WADS 2003*, 256-270, 2003.

Special Sessions

On three recent results by Subhash Khot

(Lars Engebretsen, Uriel Feige and Johan Håstad)

Extended abstract

Given the unfortunate fact that Subhash Khot had visa problems we organized a mini-session to discuss three results which has the common feature that Khot is a (co)author. All three results have subsequently been accept to be presented at FOCS 2004.

Theorem 20. *(Presented by Engebretsen) For each $p > 1$ and assuming that $NP \not\subseteq BPP$ it is hard to approximate Shortest Vector Problem in a lattice in ℓ_p-norm within any constant. Assuming that $NP \not\subseteq BPTIME(2^{poly(\log n)})$ the same problem is hard to approximate within $2^{(\log n)^{1/2-\epsilon}}$ for any $\epsilon > 0$.*

This improves the previous inapproximability factors $2^{1/p} - \epsilon$ by Micciancio [6] and $p^{1-\epsilon}$ by Khot [3]. The key to this proof seems to be a new reduction from the closest vector problem to shortest vector problem. The reduction by itself does not give the result but the resulting lattice has the property that the length of the shortest vector behaves nicely with respect to a certain *augmented tensor product*. This makes it possible to get the increased bounds.

The second result requires slightly stronger assumptions. Let us state one of the theorems of that paper.

Theorem 21. *(Presented by Håstad) For each $\epsilon > 0$ there is a $c_\epsilon > 0$ such that the following is true. Assuming that $NP \not\subseteq BPTIME(2^{n^\epsilon})$ then Graph Min-Bisection cannot be approximated in polynomial time within a factor $1 + c_\epsilon$.*

In order to prove this Khot constructs a new probabilistically checkable proof (PCP) that has some novel features. Only a constant number of questions is asked and the query pattern looks random in the following sense. For each set containing half of the locations in the proof the probability that all d questions belong to this half is close to 2^{-d}. An interesting fact is that this property is only required for attempted proofs of incorrect NP-statements. The property does not hold for valid proofs for correct statements.

The constructed PCP relies, as all known efficient PCPs, heavily on encodings by polynomials. The "randomness" requirement mentioned above seems, however, to make it difficult to apply recursion in the construction of the PCP. This, together with the fact that we only want a constant number of questions, makes it difficult

to have polynomials of non-constant degree and this gives the size 2^{n^ϵ} for the constructed PCP.

Graph Min-Bisection has previously only been proved hard to approximate under other types of assumptions such as the hardness of proving non-satisfiability of random instances of 3SAT [1].

The paper also contains results for Densest Subgraph and Bipartite Clique. These are established using the same underlying PCP and adding a combinatorial construction at the end.

Let us turn to the last paper which is a joint result by Khot, Kindler, Mossel and O'Donnell. The main result of this paper needs two assumptions. One is the Unique Games conjecture proposed by Khot [4] and the other is a new conjecture saying that among all functions not heavily dependent on a single coordinate, majority is close to being the most stable under random perturbations of the input.

Theorem 22. *(Presented by Feige) Assume that the Unique Games conjecture and the Majority is Stablest conjecture both are true. Then it is NP-hard to approximate Max-Cut within a factor $\alpha_{GW} + \epsilon$ for any $\epsilon > 0$. Here α_{GW} is the approximability constant achieved by the Goemans-Williamson* [2] *approximation algorithm.*

A slightly weaker constant, still beating the best previous bound can be obtained assuming only the Unique Games conjecture. This result is obtained through a, given the assumptions, rather natural PCP. This gives additional evidence of the strength of the unique games conjecture and thus increases the importance of resolving this conjecture. It has already been established [5] that this conjecture implies that 2 is the correct inapproximability constant for Vertex Cover.

References

[1] U. Feige, Relations between average case complexity and approximation complexity. STOC 2002.
[2] M. Goemans and D. Williamson Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of ACM, 42:1115-1145, 1995.
[3] S. Khot, Hardness of approximating the shortest vector problem in high L_p norms. FOCS 2003.
[4] S. Khot, On the power of unique 2-prover 1-round games. in high L_p norms. STOC 2002.
[5] S. Khot and O.Regev, Vertex Cover might be hard to approximate within $2 - \epsilon$. Computational Complexity 2003.
[6] D. Micciancio, The shortest vector problem is NP-hard to approximate within some constant. FOCS 1998.

Special Session on Steiner Tree Problems

Mathias Hauptmann

Mathias Hauptmann gave a brief survey on dense instances of Steiner Tree Problems. He further mentioned the problem of obtaining lower bounds for approximability of the Steiner Forest Problem. Currently the best known lower bound is ≈ 1.01 due to Chlebik and Chlebikova, which is the same as for the Steiner

Tree Problem. Janka Chlebikova gave a brief survey on these hardness results and the methods used to achieve them. She points out that in order to obtain better approximation hardness results far beyond this value, one seemingly needs some fundamentally different approach, maybe based on a different kind of reduction.

R. Ravi told us about the status of a fundamental and very interesting question concerning the Steiner Forest Problem, namely the performance of greedy algorithms for this problem. One variant is the shortest path heuristic, where in each step two terminals of the same terminal set at minimum distance are connected by a shortest path. According to what R. Ravi told us, for this variant it is still open whether this yields a constant ratio approximation. The second variant he mentions is a shortest path heuristic between arbitrary active vertices, hence possibly connecting terminals from different terminal sets as log as these sets still consist of at least two terminals. For this variant, R. Ravi claims that it can be proved to be 4-approximative, based on a mixed-component approach.

Stefan Hougardy mentions a very interesting aspect of approximation algorithms for the Steiner Tree Problem: For most of the approximation algorithms published so far, one does not know whether the analysis of the algorithm is tight. Indeed Hougardy et al. gave were able to give explicit lower bounds on the performance of the 1.86-approximation algorithm of Zelikovsky, the Berman-Ramaiyer heuristic, the 1.69-approximation algorithm of Zelikovsky, the 1.66-approximation algorithm of Prömel and Steger and the 1.59-approximation algorithm of Prömel and Hougardy.

R. Ravi mentions some interesting variants of Steiner Tree and Steiner Forest Problems: The Minimum Diameter Steiner Tree Problem, the Minimum Max. Degree Steiner Tree Problem where in case the optimum value is Δ, one can always find some tree with value $\Delta+1$. In the Covering Steiner Tree Problem, we are given an instance of the Group Steiner Problem with terminal sets (*groups*) $S_1, \ldots, S_m$, and for each group a number $r_i \in \{0, \ldots, |S_i|\}$. The task is to construct a minimum cost tree containing at least r_i vertices from group S_i $(1 \leq i \leq m)$. Ravi mentions the following variant, called *Group Connector Problem*: Given some instance of the group Steiner probelm with groups $S_1, \ldots, S_m$ and additionally connection requirements $c_1, \ldots, c_m$, find a min cost subgraph F with connected components $F_1, \ldots, F_q$ such that for every $i \in \{1, \ldots, m\}$, $\max_j |F_j \cap S_i| \geq c_i$. This problem turns out to be at least as hard as Set Cover, even when points are given on a line.

Special Session on Near Optimal Decentralized Routing in Long Range Contact Networks

Nicolas Schabanel

In order to explain the ability of individuals to find short paths to route messages to an unknown destination, based only on their own local view of a social network (the small world phenomenon), Kleinberg (2000) proposed a network model based on a d-dimensional lattice of size n augmented with k long range directed links per

node. Individuals behavior is modeled by a greedy algorithm that, given a source and a destination, forwards the message to the neighbor of the current ladder, which is the closest to the destination according to the lattice distance. This algorithm computes paths of expected length $O(\log^2(n)/k)$ between any pair of nodes. Other topologies have been proposed later as to improve greedy algorithm performances. But Aspnes et al. (2002) show that for a wide class of long range link distributions the expected length of the path computed by this algorithm is always $\Omega(\log^2(n)/(k^2 \log\log n))$.

We design and analyze a new decentralized routing algorithm, in which nodes consult their neighbors nearby, before deciding to whom forward the message. Our algorithm uses similar amount of computational resources as Kleinberg's greedy algorithm: it is easy to implement, visits $O(\log^2(n)/\log^2(1+k))$ nodes on expectation and requires only $O(\log^2(n)/\log(1+k))$ bits of memory. Our algorithm computes however a $\Theta(\log(n)\cdot(\log\log(n)/\log(1+k))^2)$ - long path in expectation between any pair of nodes. Our algorithm might fit better some human behavior (such as web browsing) and may also have successful applications to peer - to - peer networks where the length of the path along which the files are downloaded is a critical parameter of the network performance.

Special Session on Approximating Combinatorial Auctions without Randomized Routing

Berthold Vöcking

(joint work with Piotr Krysta)

Combinatorial auctions allocate a number of non-identical items to bidders that express preferences about combinations of items. Such combinatorial auctions have been suggested for selling spectrum licenses, pollution permits, loading slots, network resources etc.

They essentially correspond to set packing problems or, more generally, to packing integer programs. These packing problems are NP-hard and, hence, polynomial-time algorithms can solve these problems only in an approximate sense. However, there are more difficulties to solve than only these algorithmic questions. The basic game-theoretic requirement is that auctions should be *truthful (incentive compatible)*. An auction is truthful if each bidder's best strategy is always to reveal her true valuation, regardless of the other bidders' bids and valuations. In other words, it is required that truthful bidding is the dominant strategy for each bidder.

Unfortunately, approximation algorithms often destroy truthfulness. In particular, the most general tool for approximating unsplittable packing problems – randomized rounding as introduced by Raghavan and Thompson [9, 10] does not give a truthful mechanism. To circumvent this problem, Archer et al. [1] introduce additional dropping probabilities for bidders, which makes their algorithm *monotone in the bids*, i.e., the probability for a bidder to receive a bundle increases with the bid. This approach yields a mechanism that is truthful but only

in a probabilistic sense (and only for "known bidders"). In this paper, we present a different approach to truthful mechanisms for combinatorial auctions that can be represented in form of general packing integer programs. We devise Greedy algorithms achieving the same approximation factors, up to constant factors, as randomized rounding. These algorithms naturally satisfy certain monotonicity properties, which immediately imply truthfulness.

At this point, let us remark that our Greedy algorithms are not only of interest from the point of view of mechanism design. In fact, we believe that they yield a very efficient alternative to randomized rounding on several standard packing problems like set packing, admission control, unsplittable flow, and multicast routing. Our algorithms rely on the concept of so-called "opportunity cost" introduced by Awerbuch et al. in [2] for the design of online algorithms. This concept has since then widely been applied to various online and offline packing problems. The analysis of these algorithms is usually based on potential function arguments. We use opportunity cost in a similar fashion for the design of Greedy algorithms. Our analysis, however, is based on the primal-dual method rather than potential function arguments. In fact, parts of our analysis are inspired by the work of Garg and Könenmann [6] on fractional packing problems.

In particular, we consider the following class of *packing integer programs (PIP)*:

$$\begin{aligned} \max \quad & cx \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \{0,1\}^n, \end{aligned}$$

where $A \in [0,1]^{m \times n}$, $b \geq 1$, and $c \geq 0$. Two important subcases are *column restricted PIPs* and $(0,1)$-*PIPs*. In a column restricted PIP all entries in each column of A take the same positive value or 0. In a $(0,1)$-PIP all entries of A take either the value 0 or 1, and b is assumed to be a vector of positive integers. Define $B = \min_i\{b_i\}$. We describe approximation ratios in terms of m and B. B is called the *multiplicity parameter*, and it has significant influence on the achievable approximation ratios.

Already the most restrictive form of these three variants, namely $(0,1)$-PIP, generalizes many weighted packing problems, including maximum clique, maximum weighted independent sets, b-matching in hypergraphs, as well as k-dimensional matching. In particular, $(0,1)$-PIPs correspond to the *weighted set packing problem with multiple elements*, the algorithmic problem underlying single-minded combinatorial auctions. Let us denote the universe of elements by U. The cardinality of U is m. The parameter b_i describes the number of copies that are available of the i-th element. Furthermore, there is a collection $\mathcal{S}$ of n sets over the elements in U. The j-th column of A specifies the elements in the j-th of these sets and the parameter c_j describes the profit of this set. The objective is to select a subcollection of sets from $\mathcal{S}$ maximizing the sum of the profits of the selected sets such that, for each element $e \in U$, the number of selected sets containing e does not exceed b_e.

We prove that a Greedy algorithm achieves an approximation ratio of $O(m^{\frac{1}{B+1}})$. The same approximation ratio is obtained by randomized rounding [9, 12]. In fact, the constants hidden in the Oh-notation of our approximation bound are even slightly better than the best constants known for randomized rounding [12]. Furthermore, it follows from the work of Håstad [7] and of Chekuri & Khanna [4] that a better approximation ratio in terms of m and b cannot be achieved, unless $NP = ZPP$. Let us remark that there are also other bounds on the approximation ratio of randomized rounding using other terms [11]. For example, [11] proves an upper bound of the form $O(d^{\frac{1}{B}})$ with d denoting the maximum cardinality over the sets in $\mathcal{S}$. These results, however, are incomparable to ours.

For general PIPs, we show that a Greedy algorithm achieves an approximation factor of $O(m^{\frac{1}{B}})$. Again this matches the best bound for randomized rounding [12], and this result cannot be improved unless $NP = ZPP$ [4]. Finally, for column restricted PIPs, we obtain an approximation factor of $O(m^{\frac{1}{\lfloor B+1 \rfloor}})$. In fact, in this case we need two calls to Greedy algorithms, one to the algorithm for (0,1)-PIPs and one to the algorithm for general PIPs. Our approximation factor for column restricted PIPs corresponds again to the best known approximation factor achieved by using randomized LP rounding techniques [8].

References

[1] A. Archer, C.H. Papadimitriou, K. Talwar and É. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In the *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, 2003.

[2] B. Awerbuch, Y. Azar, S. Plotkin. Throughput-competitive online routing. In *Proc. 34th ACM FOCS*, pages 32–40, 1993.

[3] A.Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Math. of Oper. Res.*, **25**, 2000.

[4] C. Chekuri and S. Khanna. On Multidimensional Packing Problems. In the *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 185–194, 1999.

[5] C. Chekuri and S. Khanna. Edge-disjoint paths revisited. In *Proc. 14th ACM-SIAM SODA*, 2003.

[6] N. Garg and J. Könemann. Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional In *Proc. 39th ACM FOCS*, pages 300–309, 1993.

[7] J. Habstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proc. IEEE FOCS*, 1996.

[8] S.G. Kolliopoulos and C. Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *Proc. 6th IPCO*, Springer LNCS, **1412**, pp. 153–162, 1998.

[9] Prabhakar Raghavan. Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs. *J. Comput. Syst. Sci.*, **37(2)**, 130-143, 1988.

[10] P. Raghavan and C.D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, **7**: 365–374, 1987.

[11] A. Srinivasan. An extension of the Lovasz Local Lemma, and its applications to integer programming, In ACM-SIAM SODA '96, 1996.

[12] A. Srinivasan. Improved Approximation Guarantees for Packing and Covering Integer Programs, *SIAM Journal on Computing*, Vol. 29, 648–670, 1999.

Open Problem Session

Open Problems on Random 3SAT

Wenceslas Fernandez de la Vega

The problem MAX R-3SAT is defined as follows: For parameters p, n, m an instance with m clauses of length 3 is obtained by picking at random 3 literals from the $2n$ literals in n variables. $opt(F)$ is the maximum number of clauses which can be satisfied simultaneously. Karpinski and Fernandez de la Vega (2002) show: For each fixed p there exists a polytime algorithm which on input F outputs an assignment which satisfies $A(F)$ clauses such that

$$Pr\left(\frac{opt(F)}{A(F)} \leq \frac{9}{8}\right) \longrightarrow 1 \quad \text{as } n \to \infty$$

Problem 1: What is the maximum ratio r for which there is a polynomial time algorithm with

$$Pr\left(\frac{A(F)}{opt(F)} \geq r\right) \longrightarrow 1 \quad \text{as } n \to \infty \text{ ?}$$

Problem 2: Can we approximate $opt(F)$? Broder, Frieze and Upfal (1993) have shown that $\frac{opt(F)}{n}$ concentrates around its expectation $E\left(\frac{opt(F)}{n}\right)$. Can we approximate $E(opt(F))$? Obtaining $\frac{A(F)}{opt(F)} \geq 7/8$ is trivial.

Steiner Tree Problems

Mathias Hauptmann

Given a graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{R}$ and pairwise disjoint sets $S_i \subseteq V$ $(1 \leq i \leq m)$ (called the groups), the Group Steiner Tree Problem asks for a mincost tree T in G containing at least one vertex from every group. The problem is well known to be at least as hard to approximate as Set Cover (Ihler 1992, Garg et al. 1998), hence using Feige's hardness result we obtain logarithmic lower bound for approximability. Recently, Krauthgamer et al. improved this to obtain a polylog. lowerr bound. For the Steiner Tree Problem (the special case when all groups are of cardinality 1) Arora (1996) gives a PTAS for geometric instances (i.e. point sets in some $\mathbb{R}^d$ with d being constant.

Problem 1: What is the approximation complexity of the geometric Group Steiner Tree Problem ?

The Steiner Forest Problem asks for a minimum cost forest F for a given set of pairwise disjoint terminal sets $S_1, \ldots, S_m$ in a graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{R}_+$ such that each S_i is inside a connected component of F. For the Steiner Tree Problem (the special case when $m = 1$, Chlebik and Chlebikova give a lower bound of ≈ 1.01 for approximability. Nothing better is known for the

Steiner Forest Problem.

Problem 2: Give better lower bounds for the approximability of the Steiner Forest Problem.

One way of looking at this problem is to consider fixed parameter complexity. The Steiner Tree Problem is known to be in FPT for both the parameters $|S|$ (the number of terminals) and $|V \setminus S|$ (the number of non-terminals in the graph. When we take the number of Steiner points used in the tree as a parameter, the problem becomes $W[1]$-hard.

Problem 2a: Can we characterize the Steiner Forest Problem in terms of fixed parameter complexity if the parameter depends on m,
the number of terminal sets ?

Consider the ϵ-Dense Steiner Tree Problem where the instance consists of a graph $G = (V, E)$, terminal set $S \subseteq V$ and such that for alll $s \in S$, the number of neighbors of s in $V \setminus S$ is at least $\epsilon \cdot |V \setminus S|$. Karpinski and Zelikovsky (1997) proved that for every fixed $\epsilon > 0$, the ϵ-Dense Steiner Tree Problem provides a PTAS. Recently we could prove that also efficient approximation schemes exist for this problem.

Problem 3: Give hardness results for the ϵ-Dense Steiner Tree Problem.

It is even not known whether the problem is NP-hard in the exact setting (i.e. solving it to optimality in polynomial time).

The Group Connector Problem

R. Ravi

Consider the following generalization of the Steiner Forest Problem and call it the *Group Connector Problem*: Given some graph $G = (V, E)$ with edge costs c: $E \to \mathbb{R}_+$ and pairwise disjoint terminal sets $S_1, \ldots, S_m$ with requirements $r_i \leq |S_i|$, find a min-cost forest F in G consisting of connected components $F_1, \ldots, F_q$ such that for every i, $\max_j |S_i \cap F_j| \geq r_i$.

Problem: What is the approximation complexity of this problem ?

The *Group Connector Problem* is log-hard to approximate even on a line. This can be seen by a reduction from the Set Cover Problem: For each element and each set containing it draw two vertices at some small distance on a line such that representatives of elements of the same set are close together and distances between the sets are very large. Let S_i consist of al representatives of element i in sets and $r_i = 2$.

Planar Min-Bisection

Marek Karpinski

The problem of approximating the minimum bisection of a graph, i.e., the problem of partitioning a given graph into two halfs so as to minimize the number of edges with exactly one end in each half, belongs to the most intriguing problems currently in the area of combinatorial optimization and approximation algorithms. The reason being that we are not able to cope efficiently at the moment with the global conditions imposed on the vertices of a graph like the condition that the two parts of the partition are of equal size. While the PLANAR MAX CUT problem is known to be in P (Hadlock 1975), for the PLANAR MAX-BISECTION problem, the question at the time is whether it is NP-hard in the exact setting. Jansen, Karpinski, Lingas and Seidel (2002) provide polynomial time approximation schemes for MAX-BISECTION on planar and geometric graphs, showing that PLANAR MAX-BISECTION $\in$ PTAS.
The problem of PLANAR MIN-BISECTION arises profoundly in contexts ranging from partitioning functions in statistical physics to some combinatorial and geometric optimization problems.

Problem 1: What is the status of the PLANAR MIN-BISECTION problem in the exact setting ?

Problem 2: Does the PLANAR MIN-BISECTION problem admit a PTAS ?

Bicriteria Minimum Spanning Tree Problem

Berthold Vöcking

In the Bicriteria MST problem we are given weights and costs on the edges of a graph. The task is to construct a spanning tree such as to optimize simultaneously weight and cost of the tree.

In the constrained version of this problem there is a specified upper bound on the cost of the spanning tree. The problem is to find a minimum weight spanning tree obeying the budget constraint.

Problem 1: Does this problem have an FPAS ?

Problem 2: Enumerate all Pareto-optimal solutions (in time polynomial in the number of solutions).

Having Problem 2 solved, one can also solve Problem 1.

Reporter: Mathias Hauptmann

Participants

Prof. Dr. Alexandre I. Barvinok
barvinok@umich.edu
Dept. of Mathematics
The University of Michigan
525 East University Avenue
Ann Arbor, MI 48109-1109
USA

Markus Bläser
mblaeser@inf.ethz.ch
Theoretische Informatik
ETH-Zürich
ETH-Zentrum
CH-8092 Zürich

Prof. Dr. Norbert Blum
blum@cs.uni-bonn.de
Institut für Informatik V
Universität Bonn
Römerstr. 164
53117 Bonn

Dr. Janka Chlebikova
chlebikj@dcs.fmph.uniba.sk
jch@informatik.uni-kiel.de
Institut für Informatik
Universität Kiel
Olshausenstr. 40
24118 Kiel

Dr. Amin Coja-Oghlan
coja@informatik.hu-berlin.de
Institut für Informatik
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin

Mary Cryan
mcryan@inf.ed.ac.uk
Laboratory for Foundations of
Computer Science, School of
Informatics, University Edinburgh
Mayfield Road
GB-Edinburgh EH9 3JZ

Prof. Dr. Artur Czumaj
czumaj@cis.njit.edu
Dept. of Computer Science
New Jersey Inst. of Technology
University Heights
Newark, NJ 07102-1982
USA

Prof. Dr. Petros Drineas
drineas@cs.yale.edu
drinep@cs.rpi.edu
drienep@cs.rpi.edu
Department of Computer Science
Rensselaer Polytechnic Institute
104 Amos Eaton Hall
Troy, NY 12180-3590
USA

Prof. Dr. Martin E. Dyer
dyer@comp.leeds.ac.uk
The School of Computing
University of Leeds
GB-Leeds LS2 9JT

Prof. Dr. Lars Engebretsen
enge@nada.kth.se
Numerical Analysis and Computer
Science
Royal Institute of Technology
S-10044 Stockholm

Prof. Dr. Uriel Feige
uriel.feige@weizmann.ac.il
Department of Computer Science
and Applied Mathematics
The Weizmann Institute of Science
P.O.Box 26
Rehovot 76100
ISRAEL

Prof. Dr. Wenceslas L. Fernandez de la Vega
lalo@lri.fr
Wenceslas.Fernandez-De-La-Vega@lri.fr
Laboratoire de Rech. Informatique
Universite de Paris Sud (Paris XI)
Centre d'Orsay, Bat. 490
F-91405 Orsay Cedex

Prof. Dr. Sudipto Guha
sudipto@cis.upenn.edu
Dept. of Computer and Information
Science
University of Pennsylvania
3330 Walnut St.
Philadelphia, PA 19104-6389
USA

Prof. Dr. Johan Hastad
johanh@nada.kth.se
Dept. of Numerical Analysis and
Computing Science
Royal Institute of Technology
S-100 44 Stockholm

Dr. Mathias Hauptmann
hauptman@cs.uni-bonn.de
Institut für Informatik V
Universität Bonn
Römerstr. 164
53117 Bonn

Thomas P. Hayes
hayest@cs.uchicago.edu
Toyota Technological Institute
at Chicago
1427 E. 60th St.
Chicago, IL 60637
USA

Prof. Dr. Dorit S. Hochbaum
hochbaum@ieor.berkeley.edu
Department of Industrial
Engineering and Operations Research
University of California
Etcheverry Hall
Berkeley, CA 94720-1777
USA

Dr. Stefan Hougardy
hougardy@informatik.hu-berlin.de
Institut für Informatik
HU-Berlin
Unter den Linden 6
10099 Berlin

Prof. Dr. Klaus Jansen
kj@informatik.uni-kiel.de
Institut für Informatik
Universität Kiel
Olshausenstr. 40
24118 Kiel

Prof. Dr. Ravindran Kannan
kannan-ravi@cs.yale.edu
kannan@cs.yale.edu
Department of Computer Science
Yale University
P.O.Box 208285
New Haven CT 06520-8285
USA

Prof. Dr. Marek Karpinski
marek@cs.uni-bonn.de
Institut für Informatik
Universität Bonn
Römerstraße 164
53117 Bonn

Dr. Michael Langberg
mikel@cs.caltech.edu
Department of Computer Science
California Institute of Technology
1200 East California Boulevard
Pasadena CA 91125
USA

Dr. Monique Laurent
monique@cwi.nl
M.Laurent@cwi.nl
Centrum voor Wiskunde en Informatica
P.O. Box 94079
NL-1090 GB Amsterdam

Prof. Dr. Hans Jürgen Prömel
proemel@informatik.hu-berlin.de
Institut für Informatik
Humboldt-Universität zu Berlin
10099 Berlin

Prof. Dr. Ramamurthy Ravi
ravi@cmu.edu
Tepper School of Business
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
USA

Prof. Dr. Rüdiger Reischuk
reischuk@tcs.uni-luebeck.de
rr@reischuk.de
reischuk@informatik.mu-luebeck.de
Institut für Theoretische Informatik
Universität Lübeck, Geb. 64
Ratzeburger Allee 160
23538 Lübeck

Prof. Dr. Nicolas Schabanel
Nicolas.Schabanel@ens-lyon.fr
Laboratoire d'Informatique du Parallelisme
Ecole Normale Superieure de Lyon
F-69364 Lyon Cedex 07

Dr. Alex Scott
scott@math.ucl.ac.uk
Department of Mathematics
University College London
Gower Street
GB-London, WC1E 6BT

Prof. Dr. Angelika Steger
steger@inf.ethz.ch
Institut für Theoretische Informatik, ETH Zentrum
IFW E49.2
Haldeneggsteig 4
CH-8092 Zürich

Martin Thimm
thimm@informatik.hu-berlin.de
Institut für Informatik
Humboldt-Universität zu Berlin
10099 Berlin

Prof. Dr. Siavash D. Vahdati
vahdati@informatik.uni-mannheim.de
Theoretische Informatik
Universität Mannheim
D7,27
68131 Mannheim

Prof. Dr. Berthold Vöcking
voecking@ls2.cs.uni-dortmund.de
Universität Dortmund
Informatik 2
Baroper Str. 301
44221 Dortmund