Semidefinite Programming and Approximation Algorithms: First-Year Project Report NSF Grant DMI-9908077

Principal Investigator: Yinyu Ye

March 29, 2000

1 Introduction

The aim of the project is to develop, analyze and implement semidefinite programming approximation algorithms for solving large-scale combinatorial, discrete, and global optimization problems that arise in production management and network planning. The PI will do research to strengthen existing results and to explore new techniques for improving approximation accuracy and algorithm efficiency. He will also do research on implementation and application toward solving real-world problems. The research objectives include the following:

1. Approximation quality: Develop better approximation algorithms, in particular the semidefinite relaxation algorithm, for solving indefinite quadratic and graph partition optimization problems. These problems have been viewed the most challenging but promising problems in the research field.

2. Algorithm development: Design polynomial interior-point methods for solving semidefinite relaxation programs. In particular, develop efficient methods to take advantage of the structure and sparsity of problem data. This research possesses both theoretical and practical values.

3. Implementation: Implement the state of the art interior-point method in solving semidefinite programs arising from combinatorial and discrete optimization, and to develop public-domain computer codes based on these implementations. In particular, we study efficient iterative methods in solving a system of linear equations occurred in each iteration of interior-point methods.

4. Application: Apply the code to solving the network and circuit bisection model frequently used in the chip manufacture. We will test our program on a set of bench-mark problems collected in the industry.

5. Education: Educate undergraduate and graduate students at the University of Iowa via their participating in the project.

The PI believes that the project would result in one of the most complete and powerful solvers for approximating hard optimization problems with wide real-world applications. This development would also be valuable for education and human resource, intellectual knowledge, and basic scientific research. The partial discoveries and advances of the project during the first year include:

- Three new journal publications and six new working papers have been directly resulted from the project. These papers are listed in Section 4, where [J*] represents journal publication and [W*] working paper.
- Two new Ph. D. students are supported by the project and working with the PI on the project, which resulting in several working papers.
- One improved SDP computation code has been developed and released on our web side (http://dollar.biz.uiowa.edu/col/), and being widely used by academic and industrial institutions.
- The PI is invited to the 2000 International Symposium on Mathematical Programming to give a semi-plenary presentation on these results.

2 Summary of Progresses

In what follows I summarize several key results.

2.1 Progress on approximation

Given an undirected graph $G = (V, E)$ with $|V| = n$, non-negative weights w_{ij} on edges $(i, j) \in E$, and an integer k $(0 < k < n)$, the maximization graph partition (MAX-GP) problem is to determine a subset $S \subset V$ of k nodes such that an objective function $w(S)$ is maximized. Some examples of MAX-GP are: *Dense-k-Subgraph (DSP)*, where the total edge weights of the subgraph induced by S is maximized; *Max-Cut with size k (MC)*, where the total edge weights of the edges crossing between S and $V \setminus S$ is maximized; *Max-Not-Cut with size k (MNC)*, where the total edge weights of the non-crossing edges between S and $V \setminus S$ is maximized; *Max-Vertex-Cover with size k (MVC)*, where the total edge weights of the edges covered by S is maximized; etc.

Since these (MAX-GP) problems are NP-hard (e.g., see [7] for DSP, [1] for MC), one should not expect to find polynomial time algorithms for computing their optimal solutions. Therefore, we are interested in how close to optimality one can approach in polynomial time. A (randomized) polynomial time approximation algorithm for a maximization problem has a performance guarantee or worst case ratio $0 < r \leq 1$, if it outputs a feasible solution whose (expected) value is at least r times the maximal value for all instance of the problem. Such an algorithm is often called (randomized) r-approximation algorithm. A key step in designing a good approximation algorithm for such a maximization problem is to establish a good upper bound on the maximal objective value. Linear programming (LP) and semidefinite programming (SDP) have been frequently used to provide such upper bounds for many NP-hard problems.

There are several approximation algorithms for DSP. Kortsarz and Peleg [11] devised an approximation algorithm which has a performance guarantee $O(n^{-0.3885})$. Feige, Kortsarz and Peleg [8] improved it to $O(n^{-1/3+\epsilon})$, for some $\epsilon > 0$. The other approximation algorithms have performance guarantees which are the function of k/n, e.g., a greedy heuristic by Asahiro et.al. [3], and SDP relaxation based algorithms developed by Feige and Langberg [6] and Feige and Seltser [7], and Srivastav and Wolf [14]. The previously best performance guarantee, k/n for general k and $k/n + \epsilon_k$ for $k \sim n/2$, was obtained by Feige and Langberg [6].

For approximating the MC problem with size k, both LP and SDP based approximation algorithms have a performance ratio $1/2$ for all k, see Ageev and Sviridenko [1] and Feige and Langberg [6]. For $k = n/2$, i.e., the Max-Bisection, Frieze and Jerrum [9] obtained a 0.651-approximation algorithm (the same bound was also obtained by Anderson et.al [2] in their paper for max-p-section). For approximating the MNC problem with size k, the LP-based approximation algorithm has a performance ratio $1 - \frac{2k(n-k)}{n(n-1)}$ for all k, and the SDP-based algorithm has a ratio $.5 + \epsilon_k$ for $k \sim n/2$, see Feige and Langberg [6].

Our several working papers (W1 and W3-6) resulted from the project have made major contributions in this area. The papers present a new method to round an optimal solution of the SDP relaxation of the MAX-GP problem for general k. This rounding technique is related to the well-known rounding method introduced by Goemans and Williamson [10] in their seminal work for MAX-CUT, Feige and Goemans [5] for MAX-DICUT and MAX 2-SAT, Zwick [15] for constraint satisfaction problems, and Nesterov[13] and Zwick [16] for MAX-CUT. And this kind of randomized algorithm can be de-randomized by Mahajan and Ramesh[12].

As consequences of the new rounding method, we have yielded improved approximation performance ratios for DSP, MC, MNC and MVC, on a wide range of k. The following table shows our new improved ratios and previously best results for the special case $k = n/2$.

MAX-GP Problem	Previously Best Ratio	Our Improved Ratio
DSP	0.517	0.586
MC	0.651	0.699
MNC	0.54	0.602
MVC	0.80	0.812

2.2 Progress on algorithm

The approximation algorithm is to solve a semidefinite programming (SDP) relaxation problem, written in standard form as:

$$
\text{(SDP)} \qquad \begin{array}{ll} \text{Minimize} & C \bullet X \\ \text{Subject to} & A_i \bullet X = b_i,\ i = 1, \ldots, m. \\ & X \succeq 0. \end{array} \tag{1}
$$

where the given matrices $A_i \in \Re^{n \times n}$ and $C \in \Re^{n \times n}$ are symmetric, $b \in \Re^m$, and unknown $X \in \Re^{n \times n}$ is also symmetric. Furthermore, $C \bullet X = \text{trace} C^T X = \sum_{jk} C_{jk} X_{jk}$, and $X \succeq 0$

means that X is positive semidefinite. In most applications, A_i is a (sparse) rank-one matrix and C is sparse.

The dual of (SDP) can be written as:

$$\text{(DSDP)} \qquad \begin{array}{ll} \text{Maximize} & b^T y \\ \\ \text{Subject to} & \sum_{i=1}^m y_i A_i + S = C, \quad S \succeq 0, \end{array} \tag{2}$$

where y_i, $i = 1, \ldots, m$ are scalar variables.

There exist various interior-point algorithms that solve the semidefinite program in "polynomial" time, based on either the primal scaling (using X), the dual scaling (using S), or the primal-dual scaling (using both X and S). Recently, the dual-scaling algorithm (see Benson et al. [4]) has gained some attention, since it (using S) better exploits the sparsity structure of the problems than the other methods.

However, solving a linear system of a fully dense gram matrix in each step of the algorithm becomes the time-bottleneck of computational efficiency. To overcome this difficulty, in Working Paper (W7) we have tested using an iterative method, the conjugate gradient method with a simple preconditioner, to solve the linear system with a prescribed accuracy. In this report, we report computational results of solving semidefinite programs with dimension up to 14,000, which show that the iterative method could save computation time up-to 25 times of using the directed Cholesky factorization.

In Table 2.2 we compares the conjugate gradient method with the Cholesky factorization for solving the Max-Cut problem. "Dim" represents the size of the problem (the number of vertices in the graph), "Spars" is the density of the graphs, "SFden" is the density of the Cholesky factor of S (after reordering). The next two sections contain the solution times, the cut values, and their comparisons between the CG (conjugate gradient) and the CH (Cholesky factorization) solvers. Column "CG/CH" shows the ratio of the CG solution time to the CH time, and column "(CG-CH)/CH" is the ratio of the difference of the two resulting max-cut values to the max-cut value resulted from the CH solver. There, we see that the CG solution time is uniformly less than the CH solution time, while the two max-cut values remains extremely close.

This table shows a significant speed improvement for the dual-scaling algorithm from the CH factorization to the CG solver. For G77 (this graph has 14000 vertices), the max-cut solution takes 9 days to obtain when the CH factorization is used; while it takes about 9 hours when the CG solver is used. At the same time, the solution quality of the CG solver is even better than that of the CH factorization.

2.3 Progress on application

VLSI circuit partition techniques are used to reduce the VLSI chip size by carefully replacing components and interconnections in different parts, while enable efficient parallel simulation and communication and minimize time delays among these circuit parts. The circuit bisection is a basic technique in partitioning process. It divides a circuit into two parts with similar sizes, but at the same time minimizes the number of "connections" between the two circuit parts.

Name	Dim	Spars(%)	SFden(%)	Time CG	Time CH	Time CG/CH	Cut-Value CG	Cut-Value CH	Cut-Value CG/CH
G11	800	0.63	2.6	16.61	28.38	0.59	542	532	0.01
G12	800	0.63	3.64	17.66	29.84	0.59	540	534	0.01
G13	800	0.63	4.37	18.19	28.74	0.63	564	554	0.01
G14	800	1.59	14.49	35.23	47.76	0.74	2922	2982	-0.02
G15	800	1.58	13.88	32.11	54.9	0.58	2938	2975	-0.01
G20	800	1.59	14.05	32.03	58.46	0.55	838	876	-0.04
G21	800	1.58	13.86	37.56	61.29	0.61	841	855	-0.01
G22	2000	1.05	47.88	4123.31	4325.62	0.95	12960	12989	-0.00
G23	2000	1.05	47.68	3233.52	3402.86	0.95	13006	13002	0.00
G24	2000	1.05	48.29	3250.7	3549.09	0.92	12933	12985	-0.00
G30	2000	1.05	48.53	3718.93	3929.19	0.94	3038	3080	-0.01
G31	2000	1.05	48.70	3835.70	3784.45	1.01	2851	2936	-0.02
G32	2000	0.25	1.70	142.61	616.96	0.23	1338	1302	0.02
G33	2000	0.25	1.85	132.48	352.90	0.37	1330	1286	0.03
G34	2000	0.25	2.28	156.66	383.67	0.40	1334	1292	0.03
G48	3000	0.17	1.73	343.64	1113.6	0.31	6000	6000	0
G49	3000	0.17	1.57	303.3	1197.64	0.25	6000	6000	0
G50	3000	0.17	1.28	264.59	1706.73	0.16	5880	5880	0
G55	5000	0.1	1.04	1474.77	17854.76	0.08	9960	9960	0
G56	5000	0.12	8.76	15618.6	31341.04	0.50	3634	3649	-0.00
G57	5000	0.1	0.82	1819.76	15401.37	0.12	3320	3208	0.03
G60	7000	0.08	8.46	58535.13	89700.25	0.65	13610	13658	-0.00
G61	7000	0.08	8.46	52719.63	90491.23	0.58	5252	5273	-0.00
G62	7000	0.07	0.66	5187.23	63139.88	0.08	4612	4476	0.03
G63	7000	0.05	0.41	21386.79	62248.04	0.34	8017	8059	-0.00
G64	7000	0.18	11.19	102163.92	150017.27	0.68	7624	7861	-0.03
G65	8000	0.06	5.21	58309.63	133412.95	0.43	13261	13286	-0.00
G70	10000	0.03	0.31	33116.22	193932.17	0.17	9456	9499	-0.00
G72	10000	0.05	0.69	12838.07	265875.75	0.05	6644	6370	0.04
G77	14000	0.04	0.52	32643.36	804486.15	0.04	9418	9048	0.04

Table 1: Comparison of the CG and CH solvers for solving Max-Cut problems (Time is in seconds).

In Working Paper W2 we apply Semidefinite Programming (SDP) to solving the circuit bisection problem. Unlike other previous local search methods, we first translate the hypergraph into a weighted and undirected regular graph, and then solve it as a graph partition problem using the recent semidefinite program relaxation and heuristic or randomized approximation. Our preliminary computational results indicate that as the number of modules in the circuit increases, the quality of the partition resulted from SDP becomes superior to that resulted from previous methods.

We have also computed approximate solutions to the maximum stable set problem and the minimum graph coloring problem using a positive semidefinite relaxation, see W3. The positive semidefinite programs are solved using an implementation of the dual scaling algorithm that takes advantage of the sparsity inherent in most graphs and the structure inherent in the problem formulation. From the solution to the relaxation, we apply a randomized algorithm to find approximate maximum stable sets and a modification of a popular heuristic to find graph colorings. We obtained high quality answers for graphs with over 1000 vertices and more than 7000 edges.

3 Reports and Publications

3.1 New Publication

[J3] "Approximating global quadratic optimization with convex quadratic constraints," (Y. Ye), *Journal of Global Optimization* **15** (1999) 1-17.
[J2] "Mixed linear and semidefinite programming for combinatorial and quadratic optimization," (S. Benson, Y. Ye, and X. Zhang), *Optimization Methods and Software* **11&12** (1999) 515-544.
[J1] "Approximating quadratic programming with bound and quadratic constraints," (Y. Ye), *Mathematical Programming* **84** (1999) 219-226.

3.2 New Working Papers

[W7] "Solving Sparse Semidefinite Programs using the Dual Scaling Algorithm with an Iterative Solver," (C. Choi and Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (March, 2000).
[W6] "An improved rounding method and semidefinite relaxation for graph partition," (Q. Han, Y. Ye, and J. Zhang), Working Paper, Department of Management Sciences, The University of Iowa (February, 2000).
[W5] "On Approximation of Max-Vertex-Cover," (Q. Han, Y. Ye, H. Zhang and J. Zhang), Working Paper, Department of Management Sciences, The University of Iowa (February, 2000).
[W4] ".602 Approximation of the Complement of Min-Bisection and .586 Approximation of Dense-n/2-Subgraph," (Y. Ye and J. Zhang), Working Paper, Department of Management Sciences, The University of Iowa (November, 1999).
[W3] "Approximating Maximum Stable Set and Minimum Graph Coloring Problems with the Positive Semidefinite Relaxation," (S. Benson and Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (1999).

[W2] "Application of Semidefinite Programming to Circuit Partitioning," (C. Choi and Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (1999).
[W1] "A .699 approximation algorithm for Max-Bisection," Working Paper, Department of Management Sciences, The University of Iowa (March, 1999).

References

[1] A. A. Ageev and M. I. Sviridenko, "Approximation algorithms for maximum coverage and max cut with given sizes of parts", IPCO, G. Cornuejols, R. E. Burkard, G. J. Woeginger (Eds.), 17-30, 1999. Note

[2] G. Anderson, "An Approximation Algorithm for Max p-Section", Proceedings of *16th International Symposium on Theoretical Aspects of Computer Science*, volume 1563 of Lecture Notes in Computer Science, 237-247. Springer-Verlag, Berlin, 1999.

[3] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, "Greedily finding a dense subgraph," in *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, Lecture Notes in Computers Science, 1097, pages 136-148, Springer-Verlag, 1996.

[4] Steven J. Benson, Yinyu Ye, and Xiong Zhang "Mixed linear and semidefinite programming for combinatorial and quadratic optimization," *Optimization Methods and Software* **11&12** (1999) 515-544.

[5] U. Feige and M. X. Goemans, "Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT," in *Proceedings of the Third Israel Symposium on Theory of Computing and Systems*, (1995) pp. 182-189.

[6] U. Feige and M. Langberg, "Approximation Algorithms for Maximization Problems arising in Graph Partitioning" manuscript and Ph. D. thesis, 1999.

[7] U. Feige and M. Seltser, "On the densest k-subgraph problem," Technical report, Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot, September 1997.

[8] U. Feige, G. Kortsarz and D. Peleg, "The Dense k-Subgraph problem," to appear in *Algorithmica.*

[9] A. Frieze and M. Jerrum, "Improved approximation algorithms for max k-cut and max bisection," *Proc. 4th IPCO Conference* (1995) 1-13.

[10] M. X. Goemans and D. P. Williamson, " Improved approximation algorithms for Maximum Cut and Satisfiability problems using semidefinite programming," *Journal of ACM* 42 (1995) 1115–1145.

[11] G. Kortsarz and D. Peleg, "On choosing a dense subgraph," in *34th FOCS*, (1993) 692-701.

[12] S. Mahajan and H. Ramesh, "Derandomizing semidefinite programming based approximation algorithms," in *36th FOCS*, (1995) pp. 162-169.

[13] Y. E. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optimization Methods and Software* 9 (1998) 141-160.

[14] A. Srivastav and K. Wolf, "Finding dense subgraphs with semidefinite programming," in *Approximation Algorithms for Combinatorial Optimization*, K. Jansen and J. Rolim (Eds.) (1998) pp. 181-191.

[15] U. Zwick, "Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint," In *9th SODA* (1998) 201-210.

[16] U. Zwick, "Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to Max-Cut and other problems," in *31th STOC* (1999) 679-687.