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

Principal Investigator: Yinyu Ye

November 22, 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 three 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 general 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 was 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

2.1.1 Max Set Splitting

In the *Max Set Splitting* problem, we are given a finite set $U = \{1, 2, \cdots, m\}$ and collection $C = \{S_1, S_2, \cdots, S_n\}$ of subsets of U, and the goal is to find a partition of U into two subsets, $U = U_1 \cup U_2$, that maximizes the number of subsets in C that are split. A subset $S_j \in C$ is said to be split by the partition $U = U_1 \cup U_2$ if $S_j \cap U_1 \neq \emptyset$ and $S_j \cap U_2 \neq \emptyset$. This problem is also called *Max Hypergraph Cut*. *Max Ek-Set Splitting* is a special case of *Max Set Splitting* in which all subsets in C have cardinality exactly k.

For any fixed $k \geq 2$, *Max Ek-Set Splitting* problem was shown to be NP-hard by Lovasz [15]. It has also been shown to be APX-complete by Petrank [18]. *Max E2-Set Splitting* is exactly the extensively studied MAX CUT problem, which is known to be NP-hard to approximate within $\frac{16}{17} + \epsilon$, for any $\epsilon > 0$ [21]. Goemans and Williamson [9], in a major breakthrough, used semidefinite programming (SDP) to obtained an approximation algorithm for MAX CUT with a performance guarantee 0.87856. For *Max E3-Set Splitting* it has been shown by Kann et.al. [12] to be approximable within the same performance guarantee as MAX CUT; This bound has been improved to 0.87867 [25], and 0.90871 [24] by Zwick. Very recently, Guruswami [13] shows that *Max E3-Set Splitting* is not approximable within a factor of $\frac{21}{22} + \epsilon$. For $k \geq 4$, *Max Ek-Set Splitting* is approximable within $1 - 2^{1-k}$ [2, 12], and this is the best possible since Hastad [14] shows it is hard to approximate within a factor of $1 - 2^{1-k} + \epsilon$. Arora et.al. [4] designed a PTAS for dense instance of this problem. For general *Max Set Splitting* problem, the previously best known result is a 0.72405-approximation algorithm due to Andersson and Engebretsen [3]. Ageev and Sviridenko [1] give a 1/2-approximation algorithm for this problem with constraint $|U_1| = k$ for a given k.

We note that Zwick [25] has constructed an approximation algorithm for this problem. Based on the numerical evidence, he conjectured that the algorithm has performance guarantee 0.7977; but he is not able to prove it yet.

A closely related problem of *Max Set Splitting* is *Max NAE SAT* problem, which is a variant of the well-known *MAX SAT* problem, where the objective is to maximize the clauses that contain both true and false literals. *Max NAE SAT* has been shown to be APX-complete by Papadimitriou and Yannakakis [17]. *Max (Ek-)Set Splitting* is a special case of *Max (Ek)-NAE SAT* in which all literals appear unnegated. All of the results for *Max (Ek-)Set Splitting* mentioned above hold for *Max (Ek)-NAE SAT*.

In this paper, built on the algorithm of Andersson and Engebretsen [3], we present an improved 0.74996$-$approximation algorithm for *Max Set Splitting*. Recall that the algorithm of Andersson and Engebretsen [3] has three steps: (1) solve a semidefinite programming (SDP) relaxation of *Max Set Splitting*, (2) find a partition by using the rounding method of Goemans and Williamson [9], (3) add a probabilistic postprocessing step, where the partition of the second step is perturbed.

We improve the algorithm of Andersson and Engebretsen [3] in the following ways. First, we present a strengthened SDP relaxation for *Max Set Splitting* by adding some valid inequalities. Second, we round the SDP solution by an improved rounding method, which was introduced by Nesterov [16], Zwick [25] and Ye [23]. Our improvement also relies on a tighter analysis compared with [3].

We also consider the Satisfiable Instance of *Max Set Splitting* problem, i.e., all the subsets in C can be split in an optimal partition. In this case, the performance ratio of our algorithm can be improved to 0.758. The only known result for Satisfiable Instance is a 0.912$-$ approximation algorithm for *Max E3-Set Splitting* [25].

This paper is organized as follows. In the next section, we present an strengthened SDP relaxation based approximation algorithm for *Max Set Splitting*. In section 3, we analyze the quality of the partition resulted from rounding an optimal SDP solution. We analyze the performance ratio of the perturbation step and the final partition in section 4. In section 5, we extend our algorithm to *Max NAE SAT* problem.

2.1.2 Machine Scheduling

In the parallel machine scheduling problem, we are given a set of n jobs that have to be scheduled on m identical parallel machines. Each job J_j $(j = 1, 2, \cdots, n)$ has a positive processing time p_j and a positive weight w_j. In a feasible schedule, every job J_j is processed for p_j time units on one of the m machines without interruption. Every machine can process at most one job at a time, and every job can be processed on at most one machine at a time. Let C_j denote the completion time of job J_j in a given schedule. Then, two well studied optimality criteria in the literature are: to minimize the makespan $C_{\max} = \max_{1\le j\le n} C_j$, or to minimize the total weighted completion time $\sum_{1\le j\le n} w_j C_j$. See Chen, Potts and Woeginger [7] and Hall [10] for excellent surveys on parallel machine scheduling problems.

Most of the parallel machine scheduling models assume that each machine can process arbitrarily number of jobs. However, in many flexible manufacturing systems, as well as VLSI chip production, it is important to balance the number of jobs assigned to the different production facilities. In [22], Tsai proposed the following two-parallel machines scheduling problem with capacity constraints. The model is the two-parallel machines scheduling problem with an upper bound $q(n/2 \leq q \leq n)$ on the number of jobs that can be processed on a single machine. Tsai developed a heuristic for this problem with the goal to minimize the makespan (we denote it by $P2/q/C_{\max}$), and showed that it is asymptotically optimal when jobs processing times are independent and uniformly distributed. For the same problem, Bramel, Chan and Simchi-Levi [5] showed that a modified version of the well-known Longest Processing Time First heuristic (LPT) has a worst case ratio of 7/6 and this bound is tight.

In this paper, we study the two-parallel machines scheduling problem with capacity constraints and our goal is to minimize the total weighted completion time. We will denote this problem by $P2/q/\sum w_j C_j$. Without the capacity constraint, the problem is NP-hard but admits a *fully polynomial time approximation scheme* [19]. We present here an 1.1626-approximation algorithm for the $P2/q/\sum w_j C_j$ problem with the capacity constraint.

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.

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. [J3]) has gained some attention, since it (using S) better exploits the sparsity structure of the problems than the other methods. Recently, Benson and I [W3] have developed a new code, DSDP3, to solve general SDPs. The code maintains the feature of exploiting the sparse structure of problems and is competitive in solving other unstructured problems as well.

3 Reports and Publications

3.1 New Publication

[J3] "Solving large-scale sparse semidefinite programs for combinatorial optimization," (S. Benson, Y. Ye, and X. Zhang), *SIAM J. Optimization* **10** (2000) 443-461.
[J2] "An efficient algorithm for minimizing a sum of P-norms," (G. Xue and Y. Ye), *SIAM J. Optimization* **10** (2000) 551-579.
[J1] "On homotopy-smoothing methods for variational inequalities," (X. Chen and Y. Ye), *SIAM J. Control & Optimization* **37** (1999) 589-616.

3.2 New Working Papers

[W3] "DSDP3: Dual Scaling Algorithm for General Positive Semidefinite Programming," (S. Benson and Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (November, 2000).
[W2] "An Approximation Algorithm for the Two-Parallel Machines Scheduling Problem with Capacity Constraints," (H. Yang, Y. Ye and J. Zhang), Working Paper, Department of Management Sciences, The University of Iowa (August, 2000).
[W1] "Improved Approximation for Max Set Splitting and Max NAE SAT," (J. Zhang, Y. Ye, and Q. Han), Working Paper, Department of Management Sciences, The University of Iowa (June, 2000).

References

[1] A. A. Ageev and M. I. Sviridenko, "An Approximation Algorithm for Hypergraph Max k-Cut with Given Sizes of Parts," *BRICS Report Series*, RS 99-49, (1999).

[2] P. Alimonti, "Non-oblivious Local Search for Graph and Hypergraph Coloring Problems," In *Proc. of 21st International Workshop on Graph-Theoretic Concepts in Comp. Sci.*, Lecture Notes in Computer Sciences 1017, 167-180, Springer-Verlag, Berlin, (1995).

[3] G. Andersson and L. Engebrestsen, "Better Approximation Algorithms for Set Splitting and Not-All-Equal SAT," *Information Processing Letters* 65, 305-311, (1998).

[4] S. Arora, D. Karger and M. Karpinski, "Polynomial Time Approximation Schemes for Dense Instances of NP-hard Problems," *Journal of Computer and System Science*, 58, 193-210, (1999).

[5] J. Bramel, L. Chan and D. Simchi-Levi, "Worst-Case Analysis of the Modified LPT Heuristic for the Two-Machine Makespan Problem with Capacity Constraints," submitted to *Discrete Applied Mathematics*.

[6] D. Bertsimas and Y. Ye, "Semidefinite Relaxations, Multivariate Normal Distributions, and Order Statistics," *Handbook of Combinatorial Optimization*, Vol (3), D.-Z. Du and P.M. Pardalos (Eds.) 1-19, Kluwer Academic Publishers, (1998).

[7] B. Chen, C. N. Potts and G. J. Woeginger, "A Review of Machine Scheduling: Complexity, Algorithms and Approximability," *Handbook of Combinatorial Optimization (Vol. 3)*, D.-Z. Du and P.M. Pardalos (Eds.) pp. 21-170, (1998 Kluwer Academic Publishers).

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

[9] M. X. Goemans and D. P. Williamson, "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming," *Journal of ACM*, 42, 1115-1145, (1995).

[10] L. A. Hall, "Approximation Algorithm for Scheduling," in *Approximation Algorithms for NP-hard Problems*, D. S. Hochbaum (Eds.) pp 1-45, (1996 PWS Publishers).

[11] Q. Han, Y. Ye and J. Zhang, "An Improved Rounding Method and Semidefinite Programming Relaxation for Graph Partition," *Working Paper*, Department of Management Sciences, the University of Iowa, March 2000.

[12] V. Kann, J. Lagergren and A. Panconesi, "Approximability of Maximum Splitting of k-Sets and Some Other APX-complete Problems," *Information Processing Letters*, 58, 105-110, 1996.

[13] V. Guruswami, "The Approximability of Set Splitting Problems and Satisfiability Problems with no Mixed Clauses," *ECCC Report,* TR99-043, (1999).

[14] J. Hastad, "Some Optimal Inapproximability Results," *Manuscript*, (1999), Preliminary version in *STOC'97.*

[15] L. Lovasz, "Coverings and Colorings of Hypergraphs," In *Proc. 4th Southeastern Conf. on Combinatorics, Graph Theory, and Computing*, 3-12, Utilitas Mathematica Publishing, Winnipeg, (1973).

[16] Yu. E. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optimization Methods and Software*, 9:141–160, 1998.

[17] C. H. Papadimitriou and M. Yannakakis, "Optimization, Approximation, and Complexity Classes," *J. Comput. System. Sci.*, 43, 425-440, (1991).

[18] E. Petrank, "The Hardness of Approximation: Gap Location," *Computational Complexity*, 4, 133-157, (1994).

[19] S. Sahni, "Algorithms for Scheduling Independent Tasks," *Journal of ACM*, 23 (1976) 116-127.

[20] M. Skutella, "Semidefinite Relaxations for Parallel Machine Scheduling," in *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, (1998) pp. 472-481.

[21] L. Trevisan, G. B. Sorkin, M. Sudan and D. P. Williamson. "Gadgets, approximation and linear programming, " *SIAM Journal on Computing*, 29(6), 2074-2097, (2000).

[22] L. H. Tsai, "Asymptotic Analysis of an Algorithm for Balanced Parallel Processor Scheduling," *SIAM Journal on Computing*, 21, (1992) pp. 59-94.

[23] Y. Ye, "A .699-Approximation Algorithm for Max-Bisection," *Working Paper*, Department of Management Sciences, the University of Iowa, March 1999.

[24] U. Zwick, "Approximation Algorithms for Constraint Satisfaction Problems Involving at Most Three Variables Per Constraint," *9th SODA*, 201-210, (1998).

[25] U. Zwick, "Outward Rotations: A Tool For Rounding Solutions of Semidefinite Programming Relaxations, with Application to MAX CUT and Other Problems," *31th STOC*, 679-687, (1999).