Interior-Point Algorithms: 1997 Annual Progress Report on NSF Grant DMI-9522507

Principal Investigator: Yinyu Ye

February 19, 1998

1 Introduction

This document is the second year progress report on an optimization project funded by NSF Grant DMI-9522507. The aim of the proposal is to develop and analyze interior-point algorithms and their variants for solving large-scale linear, quadratic, and nonlinear optimization problems that arise in production management and resource allocation. It is a continuation of the PI's previous projects funded by NSF, and it is motivated by the demand to resolve important remaining theoretical and practical issues in the field. In the new project the PI will do research to strengthen existing results and to explore new techniques for improving linear and nonlinear programming algorithms. He will also do research on implementation and application to solve real-world problems.

In general, results in the area of developing efficient algorithms for solving large-scale optimization problems will be of great importance in improving the efficiency of manufacturing systems, communication networks, aircraft routing, multi-commodity-flow operations, and resources planning. Strengthening research in this area will definitely contribute to the national interest in industrial competitiveness and scientific knowledge.

More specific, my research topics include the following.

1. Complexity issues: Continue to develop complexity theories for linear programming (LP) and approximation algorithms for combinatorial and indefinite quadratic (global) optimization.

2. Algorithm developments: Extend LP interior-point algorithms to solving semidefinite programs. In particular, develop efficient techniques to solve large-scale and sparse semidefinite relaxations arisen from max-cut and box-constrained quadratic problems.

3. Implementation and Application: Continue to implement the state of the art interior-point algorithms for both linear and nonlinear programming, and to develop public-domain computer programs based on these implementations. Continue to apply the algorithms to solving real world problems.

4. Education: Continue to educate undergraduate and graduate students at the University of Iowa via their participating in the project. Three Ph. D. students graduated during the past and they are now faculties or post-docs in various US or overseas

institutions. The PI has published a text book on this topic and directing a computational optimization laboratory (COL) in his department. Currently, the lab has one post-doc associate and two graduate students.

Progresses have been made in each of these areas in 1997:

- Four new publications have been resulted from the project in *Journal of Combinatorial Optimization* and *Mathematical Programming.* Six working papers were submitted for possible publication. These papers are listed in Section 5, where [J*] for journal publication and [W*] for working paper.
- One implementation program is being developed to solve large-scale semidefinite programs for combinatorial and nonconvex quadratic optimization problems.
- We continue a network design project for the Rockwell International at Cedar Rapids and start a Protein Folding project, where some of our computer programs will be used.

2 Project Progress

I briefly summarize major results obtained in 1997 and discuss ongoing research activities.

2.1 Progress in Combinatorial and Global Optimization

Recently, there were several theoretical results on approximating combinatorial and quadratic optimization problems by using semidefinite programming. (see, e.g., Goemans and Williamson [9] and Nesterov [13]). The positive semi-definite relaxation was first proposed by Lovász and Shrijver [12], also see recent papers by Fujie and Kojima [7], Helmberg and Rendl [10], Karisch, Rendl, and Clausen [11], and Polijak, Rendl and Wolkowicz [15].

Consider the quadratic programming (QP) problem with (diagonal) quadratic and "box" constraints

$$
\text{(QP)} \qquad \begin{array}{lll} \bar{q}(Q) := & \text{Maximize} & q(x) := x^T Q x \\ & \text{Subject to} & \sum_{j=1}^n a_{ij} x_j^2 = b_i, \; i = 1, \ldots, m, \\ & & -e \le x \le e, \end{array}
$$

where symmetric matrix $Q \in \Re^{n \times n}$, $A = \{a_{ij}\} \in \Re^{m \times n}$ and $b \in \Re^m$ are given and $e \in \Re^n$ is the vector of all ones. Let $\bar{x}(Q)$ be a maximizer of the problem. The (QP) problem plays an important role in combinatorial and nonlinear optimization, and it is extremely difficult to find its (global) maximizer.

The function $q(x)$ has a minimizer and a maximizer over the bounded feasible set. Let $\underline{q} := -\bar{q}(-Q)$ and $\bar{q} := \bar{q}(Q)$ denote their minimal and maximal objective values, respectively. An ϵ-maximal solution or ϵ-maximizer, $\epsilon \in [0, 1]$, for (QP) is defined as an $x \in \mathcal{F}$ such that

$$
\frac{\bar{q} - q(x)}{\bar{q} - \underline{q}} \le \epsilon.
$$

Goemans and Williamson [9] proved an approximation result for the max-cut problem (all coefficients in Q are nonnegative) where $\epsilon = 1 - 0.878$. Nesterov [13] generalized their result to general Q and proved a result of $\epsilon = 4/7$.

However, approximation algorithms for (QP) were open. Recently, we [W5] established an $\epsilon = 4/7$ result to solving (QP), i.e., we developed a fast (polynomial algorithm) to deliver a solution for (QP) that is guaranteed to be 4/7-maximal. In a subsequent paper, Bertsimas and I [W2] developed a number of other approximation results for solving combinatorial problems.

2.2 Progress in Algorithm Development

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 are actually several polynomial algorithms for solving SDP. One is the primal-scaling algorithm (Nesterov and Nemirovskii [14], Alizadeh [1], Vandenberghe and Boyd [17], and Ye [19]), which is the analogue of the primal potential reduction algorithm for linear programming. This algorithm uses X to generate the iterate direction. Another is the dual-scaling algorithm (Vandenberghe and Boyd [17], Anstreicher and Fampa [2], and Ye [19]), which is the analogue of the dual-scaling algorithm for linear programming. The dual-scaling algorithm uses only S to generate the iterate direction. The third is the primal-dual scaling algorithm which uses both X and S to generate iterate directions (see Todd [16], Fujisawa et al. [8], and references therein). All these algorithm possess $O(\sqrt{n} \log(1/\epsilon))$ iteration complexity to yield accuracy ϵ.

Although they can be solved in polynomial time, large-scale SDP problems are difficult to solve. In practice, only problems with dimension (n) in hundreds are being solved and reported. This is because most of those primal-dual algorithms have to handle very dense matrices and hardly to use the structure of the original problems. On the other hand, large-scale combinatorial and quadratic optimization problems have features which make the dual-scaling algorithm the most suitable choice:

1. For large scale problems, S tends to be very sparse and structured since it is the linear combination of C and the A_is. This sparsity allows considerable savings in both memory and computation time. Since X, the primal matrix, may be much less sparse and its structure not known beforehand, primal or primal-dual algorithms cannot fully exploit the sparseness and structure of the data.

2. The approximation problems under consideration require less accuracy than some other applications. Therefore, the superlinear convergence, exhibited by the primal-dual algorithm, may not be utilized in our applications. The dual-scaling linear programming algorithm has been shown to perform equally well when only a lower precision answer is required.

3. In most combinatorial applications, we need only a lower bound for the optimal objective value of (SDP). Solving (DSDP) alone would be sufficient to provide such a lower bound. Moreover, in most applications an interior-feasible point is available to start with. Thus, we may not need to generate an X at all.

4. Even if an optimal primal solution is necessary, our dual-scaling algorithm can generate a sparsely structured optimal X at the termination of the algorithm.

Based on this observation, Benson, Zhang and I [W3] developed a dual potential-reduction algorithm for SDP with the best computational complexity as well as efficient practical performance.

2.3 Progress in Implementation and Application

We have recently developed a *c*-code to solve semidefinite relaxations for max-cut and box-constrained QP optimization. (It is available on *http://dollar.biz.uiowa.edu/col/*.) We applied the code to solve the well-known $G-set$ problems, see Helmberg and Rendl [10]. These problems were generated by **rudy**, a machine independent graph generator written by G. Rinaldi. This set of problems became a standard test set for graph optimization, and they had never been solved before by interior-point algorithms.

Our code was run on a Pentium PC-GATEWAY2000/233(MH) with 64M memory plus additional virtual hard-disk memory. The following Table show the performance of the code. Here, *Dim* is the dimension, n, of the problem and *Spar* is the density ratio of Q. *PriObj* and *DualObj* are the primal and dual objective values and *Rgap* is the relative duality gap at termination. *Iter* is the number of interior-point iterations and *ElapTime* is the computer time used to solve the problem. As we can see, in general, the sparser of the problem, the faster of the solution.

We have also started a Protein Folding project and continue a network design project for the Rockwell International at Cedar Rapids, where some of our computer programs will be used.

3 Education and Human Resources

Beneficiaries, educationally, of the project are the Ph.D. students at the University of Iowa. We have supported several Ph.D. students at UI from this NSF grant. Some results men-

Index	Dim.	Spar.	Iter.	PriObj	DualObj	Rgap	ElapTime
G1	800	6.12%	15	-4.832990e+04	-4.833285e+04	6.1e-05	500.27
G2	800	6.12%	14	-4.835480e+04	-4.835786e+04	6.3e-05	476.59
G3	800	6.12%	16	-4.833436e+04	-4.833737e+04	6.2e-05	547.78
G4	800	6.12%	18	-4.844417e+04	-4.844583e+04	3.4e-05	599.19
G5	800	6.12%	16	-4.839649e+04	-4.839959e+04	6.4e-05	533.62
G6	800	6.12%	17	-1.062436e+04	-1.062466e+04	2.8e-05	577.73
G7	800	6.12%	17	-9.956730e+03	-9.957059e+03	3.3e-05	581.51
G8	800	6.12%	17	-1.002747e+04	-1.002776e+04	2.9e-05	574.18
G9	800	6.12%	17	-1.011446e+04	-1.011494e+04	4.7e-05	587.25
G10	800	6.12%	17	-9.940009e+03	-9.940260e+03	2.5e-05	582.96
G11	800	0.62%	21	-2.516600e+03	-2.516659e+03	2.3e-05	85.54
G12	800	0.62%	17	-2.495313e+03	-2.495499e+03	7.4e-05	72.87
G13	800	0.62%	16	-2.588380e+03	-2.588546e+03	6.4e-05	71.13
G14	800	1.59%	30	-1.276547e+04	-1.276627e+04	6.3e-05	198.94
G15	800	1.58%	36	-1.268558e+04	-1.268623e+04	5.2e-05	229.67
G16	800	1.58%	34	-1.269941e+04	-1.270008e+04	5.3e-05	220.91
G17	800	1.58%	38	-1.268467e+04	-1.268532e+04	5.1e-05	248.39
G18	800	1.59%	32	-4.663839e+03	-4.664042e+03	4.3e-05	213.18
G19	800	1.58%	37	-4.327809e+03	-4.328042e+03	5.4e-05	239.56
G20	800	1.58%	36	-4.445332e+03	-4.445572e+03	5.4e-05	231.22
G21	800	1.58%	33	-4.416982e+03	-4.417138e+03	3.5e-05	216.90
G22	2000	1.05%	24	-5.654114e+04	-5.654383e+04	4.8e-05	8993.44
G23	2000	1.05%	23	-5.657943e+04	-5.658206e+04	4.7e-05	8562.08
G24	2000	1.05%	21	-5.656189e+04	-5.656343e+04	2.7e-05	8044.10
G25	2000	1.05%	21	-5.657391e+04	-5.657700e+04	5.5e-05	8056.97
G26	2000	1.05%	23	-5.652882e+04	-5.653150e+04	4.7e-05	8785.46
G27	2000	1.05%	23	-1.656621e+04	-1.656665e+04	2.6e-05	8710.60
G28	2000	1.05%	24	-1.640239e+04	-1.640317e+04	4.8e-05	8949.05
G29	2000	1.05%	25	-1.683427e+04	-1.683556e+04	7.7e-05	9699.64
G30	2000	1.05%	46	-1.686052e+04	-1.686153e+04	6.0e-05	17645.26
G31	2000	1.05%	24	-1.646627e+04	-1.646673e+04	2.8e-05	9182.20
G32	2000	0.25%	20	-6.270123e+03	-6.270559e+03	7.0e-05	1100.11
G33	2000	0.25%	20	-6.176944e+03	-6.177250e+03	5.0e-05	1117.56
G34	2000	0.25%	24	-6.186503e+03	-6.186750e+03	4.0e-05	1380.50
G35	2000	0.64%	48	-3.205730e+04	-3.205896e+04	5.2e-05	5785.18
G36	2000	0.64%	60	-3.202194e+04	-3.202386e+04	6.0e-05	7582.25
G37	2000	0.64%	51	-3.207258e+04	-3.207451e+04	6.0e-05	6360.85
G38	2000	0.64%	65	-3.205796e+04	-3.205990e+04	6.0e-05	8017.37
G39	2000	0.64%	58	-1.150993e+04	-1.151059e+04	5.8e-05	6984.55
G40	2000	0.64%	69	-1.145841e+04	-1.145916e+04	6.6e-05	8754.52
G41	2000	0.64%	50	-1.146014e+04	-1.146088e+04	6.4e-05	6180.25
G42	2000	0.64%	53	-1.178427e+04	-1.178501e+04	6.3e-05	6539.46
G43	1000	2.10%	20	-2.812811e+04	-2.812892e+04	2.9e-05	911.69
G44	1000	2.10%	20	-2.810978e+04	-2.811155e+04	6.3e-05	982.57
G45	1000	2.10%	21	-2.809789e+04	-2.809913e+04	4.4e-05	946.19
G46	1000	2.10%	18	-2.811848e+04	-2.811978e+04	4.6e-05	832.17
G47	1000	2.10%	19	-2.814537e+04	-2.814665e+04	4.5e-05	857.11
G48	3000	0.17%	6	-2.399899e+04	-2.400000e+04	4.2e-05	1246.14
G49	3000	0.17%	6	-2.399899e+04	-2.400000e+04	4.2e-05	1250.24
G50	3000	0.17%	6	-2.395165e+04	-2.395269e+04	4.3e-05	1212.98
G51	1000	1.28%	37	-1.602412e+04	-1.602503e+04	5.6e-05	493.00
G52	1000	1.28%	40	-1.603769e+04	-1.603856e+04	5.4e-05	539.19
G53	1000	1.28%	37	-1.603796e+04	-1.603888e+04	5.7e-05	492.74
G54	1000	1.28%	33	-1.602403e+04	-1.602478e+04	4.7e-05	446.17

Table 1: Performance of the c code on solving the $G-set$ semidefinite programs; $Toler = 10^{-4}$.

tioned above are part of their Ph.D. theses.

Another outgrowth of our research has been a new seminar, meeting once a week, among graduate students and faculty members from several departments. I have also brought some of the new research topics and results to an optimization and mathematical programming course at the graduate level. I have published a text book "Interior-Point Algorithms: Theory and Analysis."

4 Publications and Technical Reports

4.1 New Journal Papers

[J1] "Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems," (Yu. Nesterov, M.J. Todd, and Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (April, 1996), to appear in *Mathematical Programming.*
[J2] "Approximation algorithms for quadratic programming," (M. Fu, Z.-Q. Luo, and Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (July, 1996), to appear in *Journal of Combinatorial Optimization.*
[J3] "Bounded error parameter estimation: a sequential analytic center approach," (E. Bai, Y. Ye and R. Tempo), to appear in *IEEE, Decision and Control, 1997.*
[J4] "Approximate Farkas lemmas and stopping rules for iterative infeasible-point algorithms for linear programming," (M. J. Todd and Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (September, 1994), to appear in *Mathematical Programming.*

4.2 New Working Papers

[W1] "Mixed linear and semidefinite programming for combinatorial and quadratic optimization," (S. Benson, Y. Ye, and X. Zhang), Working Paper, Department of Management Sciences, The University of Iowa (February, 1998).
[W2] "Semidefinite relaxations, multivariate normal distributions, and order statistics," (D. Bertsimas and Y. Ye), Working Paper, Sloan School, MIT (December, 1997).
[W3] "Solving large-scale sparse semidefinite programs for combinatorial optimization," (S. Benson, Y. Ye, and X. Zhang), Working Paper, Department of Management Sciences, The University of Iowa (September, 1997).
[W4] "An efficient algorithm for minimizing a sum of P-norms," (G. Xue and Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (September, 1997).
[W5] "Approximating quadratic programming with quadratic constraints," (Y. Ye), Working Paper, Department of Management Sciences, The University of Iowa (Mar, 1997).
[W6] "On homotopy-smoothing methods for variational inequalities," (X. Chen and Y. Ye), AMR 96/39, Applied Mathematics Report, The University of New South Wales, Sydney (December, 1996).

References

[1] F. Alizadeh, "Combinatorial optimization with interior point methods and semi–definite matrices," Ph.D. Thesis, University of Minnesota, Minneapolis, MN, 1991.

[2] K. M. Anstreicher and M. Fampa, "A long-step path following algorithm for semidefinite programming problems," Working Paper, Department of Management Science, The University of Iowa, Iowa City, 1996.

[3] K. E. Atkinson, *An Introduction to Numerical Analysis* (Second Edition John Wiley & Sons, New York, 1989).

[4] S. Benson, Y. Ye, and X. Zhang, "Solving sparse, large scale positive semidefinite programs," Working Paper, Department of Management Science, The University of Iowa, Iowa City, 1997.

[5] D. Bertsimas and Y. Ye, "Semidefinite relaxations, multivariate normal distributions, and order statistics," Working Paper, Sloan School, MIT, December, 1997.

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

[7] T. Fujie and M. Kojima, "Semidefinite programming relaxation for nonconvex quadratic programs," Research Report B-298, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Meguro, Tokyo 152, May 1995. To appear in *Journal of Global Optimization.*

[8] K. Fujisawa, M. Fukuda, M. Kojima and K. Nakata, "Numerical Evaluation of SDPA (SemiDefinite Programming Algorithm)," Research Report B-330, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, September 1997.

[9] 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.

[10] C. Helmberg and F. Rendl, "A spectral bundle method for semidefinite programming," ZIB Preprint SC 97-37, Konrad-Zuse-Zentrum fuer Informationstechnik Berlin, Takusstrasse 7, D-14195 Berlin, Germany, August 1997.

[11] S. E. Karisch, F. Rendl, and J. Clausen, "Solving graph bisection problems with semidefinite programming," Technical Report DIKU-TR-97/9, Department of Computer Science, University of Copenhagen, July 1, 1997.

[12] L. Lovász and A. Shrijver, "Cones of matrices and setfunctions, and $0-1$ optimization," *SIAM Journal on Optimization* 1 (1990) 166-190.

[13] Yu. E. Nesterov, "Quality of semidefinite relaxation for nonconvex quadratic optimization," CORE Discussion Paper, #9719, Belgium, March 1997.

[14] Yu. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming : Theory and Algorithms* (SIAM Publications, SIAM, Philadelphia, 1993).

[15] S. Polijak, F. Rendl and H. Wolkowicz, "A recipe for semidefinite relaxation for 0-1 quadratic programming," *Journal of Global Optimization* 7 (1995) 51-73.

[16] M. J. Todd, "On search directions in interior-point methods for semidefinite programming," Technical Report No. 1205, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1997.

[17] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review* 38(1) (1996) 49–95.

[18] Y. Ye, "Approximating quadratic programming with quadratic constraints," Working Paper, Department of Management Science, The University of Iowa, Iowa City, 1997.

[19] Y. Ye, *Interior Point Algorithms : Theory and Analysis* (Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, 1997).