Yugoslav Journal of Operations Research
Volume 19 (2009) Number 2, 239-248
DOI:10.2298/YUJOR0902239L

INTERIOR-POINT ALGORITHMS FOR A CLASS OF CONVEX OPTIMIZATION PROBLEMS

Goran LEŠAJA

goran@georgiasouthern.edu.

Verlynda N. SLAUGHTER

Department of Mathematical Sciences
Georgia Southern University
Statesboro, Georgia, USA

Received: March 2009 / Accepted: November 2009

Abstract: In this paper we consider interior-point methods (IPM) for the nonlinear, convex optimization problem where the objective function is a weighted sum of reciprocals of variables subject to linear constraints (SOR). This problem appears often in various applications such as statistical stratified sampling and entropy problems, to mention just few examples. The SOR is solved using two IPMs. First, a homogeneous IPM is used to solve the Karush-Kuhn-Tucker conditions of the problem which is a standard approach. Second, a homogeneous conic quadratic IPM is used to solve the SOR as a reformulated conic quadratic problem. As far as we are aware of it, this is a novel approach not yet considered in the literature. The two approaches are then numerically tested on a set of randomly generated problems using optimization software MOSEK. They are compared by CPU time and the number of iterations, showing that the second approach works better for problems with higher dimensions. The main reason is that although the first approach increases the number of variables, the IPM exploits the structure of the conic quadratic reformulation much better than the structure of the original problem.

Keywords: SOR problem, convex optimization problems, conic quadratic optimization problem, interior-point methods.

1. INTRODUCTION

In this paper we consider the problem

$$\begin{aligned} (SOR) \quad & \textit{Minimize} && \sum_{i=1}^{n} \frac{c_i}{x_i} \\ & \textit{subject to} && \sum_{i=1}^{n} a_i x_i \le b \\ & && l_i \le x_i \le u_i, \qquad i = 1,\dots,n \end{aligned} \tag{1.1}$$

where c_i, l_i, and $u_i \in R_+$.

The above problem is a nonlinear, optimization problem with convex objective function and simple linear constraints. The abbreviation SOR comes from Sum of Reciprocals because the objective function is a weighted sum of reciprocals of variables. The SOR appears often in various applications, such as in statistical stratified sampling [5] and entropy problems [3], to mention just a few.

The interior - point methods (IPMs) are best suited to solve many convex optimization problems especially with linear constraints. Thus, the SOR is solved using IPMs. In the past two decades the development of IPMs has had a profound impact on the optimization theory and practice. The renewed interest in IPMs originated with Karmarkar's paper [7] in 1984, and has been a very active area of research ever since. The list of references on the subject is extensive. We refer the interested reader to the classical monograph of Nesterov and Nemirovski [8] and the references therein.

First, the SOR is solved in a standard manner by applying a homogeneous IPM to the Karush-Kuhn-Tucker conditions of the original problem (which can be stated as monotone complementarity problem). Second, a homogeneous conic quadratic IPM is used to solve the SOR as a reformulated conic quadratic problem.

The two approaches are then numerically tested on a set of randomly generated problems. The code of choice is optimization solver MOSEK [1] which has very good subroutines that solve both approaches. The results are compared by CPU time and the number of iterations, showing that the first approach works better for problems with higher dimensions. The result confirms the experience from numerous numerical tests of variety of different optimization problems: The interior-point methods seem to work better whenever a problem can be reformulated as one with a "nice" structure such as linear, conic quadratic or semi definite optimization problem.

The outline of the paper is as follows. In Section 2 we discuss the conic quadratic reformulation of SOR. In Section 3 the IPMs used to solve the SOR are discussed. Numerical results are presented in Section 4 and the conclusions are stated in Section 5.

2. CONIC QUADRATIC REFORMULATION OF SOR

The SOR (1.1) can be transformed into a conic quadratic optimization problem. In general a conic optimization problem can be expressed in the form

$$\begin{array}{ll} \textit{Minimize} & c^T x \\ \textit{subject to} & Ax = b, \\ & x \in K \end{array} \tag{2.1}$$

where $A \in R^{mxn}$, $c, x \in R^n$, and $b \in R^m$. Set K is assumed to be a pointed closed convex cone. The types of cones most often considered are: the cone of nonnegative n-tuples, the quadratic cone, and the cone of semi definite symmetric matrices. Without loss of generality, it can be assumed that $K = K_1 \times \ldots \times K_k$, that is, the cone K is a direct product of several individual cones where each one of them is of one of the three mentioned types. All these cones are self-dual which makes working with the dual problem much easier. Moreover, it makes problem (2.1) well suited to be solved with Primal – Dual IPMs, which are considered to be the most effective of all types of IPMs.

The first step of transforming the stratified sampling SOR problem (1.1) into a conic quadratic programming problem is to transform the objective function in (1.1) by letting

$$\sum_{i=1}^{n} \frac{c_i}{x_i} = \sum_{i=1}^{n} t_i$$

where $\frac{c_i}{x_i} = t_i$; $i = 1, \ldots, n$.

Then, we introduce a new variable r_i in the following way

$$r_i = \sqrt{2d_i}, \quad r_i^2 \le 2x_i t_i, \quad x_i \ge 0, \quad t_i \ge 0 \; ; \quad i = 1, ..., n.$$

It is easy to see that $(x_i, t_i, r_i) \in K_i$, where K_i is a rotated quadratic cone. The rotated quadratic cone is defined as

$$K^r := \left\{ x \in R^n : \; 2x_1x_2 \ge \| x_{3:n} \|^2 ; \; x_1, x_2 \ge 0 \right\}.,$$

whereas the quadratic cone is given by

$$K^q := \left\{ x \in R^n : \; x_1^2 \ge \| x_{2:n} \|^2 , \; x_1 \ge 0 \right\}.$$

with the terms $\|x_{i:n}\|$; $i = 2,3$ corresponding to $\|(x_2, x_3, ..., x_n)\|$ and $\|(x_3, x_4, ..., x_n)\|$ respectively. The cone K^q is also referred to as an "ice-cream cone" because of its unique shape that in R^3 ~~it~~ resembles an ice-cream cone. It can be shown that the rotated cone can be reduced to the quadratic cone [4].

The next step is to transform the constraint of problem (1.1) by adding a slack variable, $s_{n+1} \ge 0$ to the constraint, which results in

$$\sum_{i=1}^{n} a_i x_i + s_{n+1} = b.$$

The final step is to transform the box constraints, $l_i \le x_i \le u_i, \quad i = 1,...,n$ by introducing slack variable, w_i and surplus variable, v_i for each $i = 1,..., n$

$$w_i = u_i - x_i \Rightarrow x_i + w_i = u_i \Rightarrow w_i \ge 0,$$
$$v_i = x_i - l_i \Rightarrow x_i - v_i = l_i \Rightarrow v_i \ge 0.$$

With these transformations the SOR (1.1) can be written in the conic quadratic form.

$$\begin{array}{lll} (SOR-CQ) & Minimize & \sum_{i=1}^{n} t_i \\ & subject\ to & \sum_{i=1}^{n} a_i x_i + s_{n+1} = b \\ & & r_i = \sqrt{2c_i} \\ & & x_i - v_i = l_i \\ & & x_i + w_i = u_i \qquad i = 1,...,n \\ & & (x_i, t_i, r_i) \in K_i \\ & & (v_i, w_i, s_{n+1}) \in R_+ \end{array} \tag{2.2}$$

This reformulation first appeared in the second author's master thesis [9] in 2004 (under guidance of the first author) on which this paper is based. As far as we know, this is the first time this reformulation has been used in the literature and it is the main result of the thesis and the paper. Subsequently, related problem and its reduction to conic quadratic form was discussed in [6]. However, the problem in [6] is different because the objective function is weighted sum of reciprocals of linear functions while the objective function in our paper is the weighted sum of reciprocals of each variable separately. In addition, the IPM used to solve the problem is different than the ones used in our paper.

Throughout the rest of the paper the above formulation of the SOR-CQ will be used with $c_i = 1$.

3. ALGORITHMS FOR SOLVING SOR

There are a number of nonlinear optimization methods that can be used to solve SOR, such as various feasible direction methods (Frank-Wolfe, or reduced gradient method, etc), penalty methods or IPMs. It is known that for many convex optimization problems and especially with linear constraints, the IPMs are the most efficient methods. Therefore, in this paper we focus on solving SOR with two different IPMs and finding the more efficient one.

The first (standard) approach is to use IPM on the original formulation of the problem, or more precisely, on the KKT conditions of the problem which lead to the monotone nonlinear complementarity problem. Specifically, the KKT conditions for SOR described in (1.1) are given below.

$$\begin{bmatrix} -\frac{1}{x_1^2} \\ -\frac{1}{x_2^2} \\ \vdots \\ -\frac{1}{x_n^2} \end{bmatrix} + \eta \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} - \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{aligned} &\sum_{i=1}^{n} a_i x_i \le b, \\ &l_i \le x_i \le u_i, \qquad i = 1,\dots,n \\ &\eta\left(\sum_{i=1}^{n} a_i x_i - b\right) = 0, \\ &\lambda_i (x_i - u_i) = 0, \qquad i = 1,\dots, n \\ &\nu_i(-x_i + l_i) = 0, \qquad i = 1,\dots, n \\ &\eta \ge 0, \\ &\nu_i \ge 0,\ \lambda_i \ge 0, \qquad i = 1,\dots,n \end{aligned} \tag{3.1}$$

The Lagrange multipliers $\eta \ge 0$ and $\nu_i \ge 0,\ \lambda_i \ge 0, i = 1,\dots,n$ correspond to the linear constraint and lower and upper part of the box constraints respectively.

We use the homogeneous IPM for nonlinear complementarity problems proposed by Andersen and Ye in [2]. Implementation version of that method is described in Andersen and Ye [3].

The second approach is to use a suitable version of IPM on the conic-quadratic reformulation of the SOR described in (2.2). We use the homogeneous IPM for conic quadratic problems proposed by Andersen et al. in [4].

It is not our intention to describe these algorithms in detail in this paper. We refer the interested reader to the cited papers and the references therein. We only mention briefly the main ideas on which they are based. The idea of the homogeneous model is to embed the original optimization problem into a slightly larger problem which always has a solution. Furthermore, an appropriate solution to the embedded problem either provides a certificate of the infeasibility or a scaled solution to the original problem. Consequently, instead of solving the original problem, the embedded problem is solved using an appropriate version of the IPM.

The theory behind the IPMs is based on the concept of the central path. Central path is a trajectory that is obtained by perturbing the complementarity equation of the KKT conditions for the given optimization problem using a positive parameter usually denoted by μ . For many important optimization problems, including the ones stated in this paper, it has been shown that the central path is well defined and it converges to the optimal solution when $\mu \to 0$. The main idea is to trace the central path using Newton's method while gradually reducing the parameter μ to zero. In general, it is not possible to compute a point on the central path, called μ-center, efficiently. The main achievement of the theory of IPMs was to show that it suffices to trace the central path approximately, that is, the global convergence of IPMs with a very good polynomial complexity can still be proved as long as the iterates are in the appropriate neighborhood of the central path.

A simplified diagram of the IPM is given in Figure 3.1.

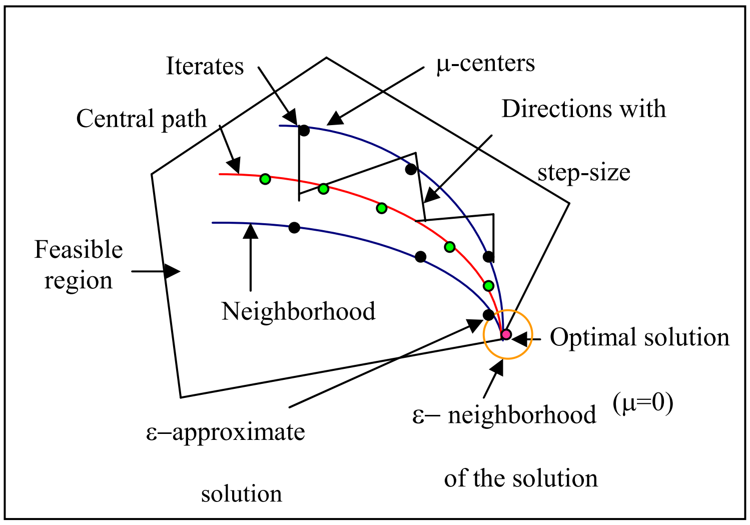

Figure 3.1

The code that successfully implements the above mentioned algorithms is MOSEK which has been created by E. Andersen [1]. The subroutine MOSEKOPT solves the conic quadratic problems, while the subroutine MSKSCOPT solves the nonlinear optimization problems directly (via KKT conditions). For details on MOSEK visit the MOSEK website at www.mosek.com.

4. NUMERICAL RESULTS

For the numerical testing we used the PC platform with 3.2 GHz Pentium(R) 4 processor and Window XP operating system.

The two approaches to solving SOR are tested on the set of randomly generated problems. The generator and the interface to MOSEK are written in MATLAB. The dimension of the problems varies from two to a hundred variables. The version of the MOSEK used in this paper was the release 4. In addition, we used the student version with a limit of three hundred variables. Since the conic quadratic reformulation of SOR triples the number of variables, the largest problems are limited to hundred variables.

In addition to the problems created by the generator there are a few problems in lower dimensions that are created "by hand". The problems are named by the dimension and if letter G is added it indicates that the problem was created by the generator. The problems are compared by the number of iterations and the CPU time (in seconds). The results are presented in four tables. First one contains the results for the problems with dimension up to twenty, the second one contains results of the problems whose dimension is between thirty and seventy, the third one contains the results of the problems with dimension between eighty and hundred and the fourth one contains the summary of the results expressed by average CPU and number of iterations.

		CONIC (MOSEKOPT)			**NON-CONIC (MSKSCOPT)**	
Problem	**Number of Variables**	**CPU Time**	**Number of Iterations**		**CPU Time**	**Number of Iterations**
Two 1	2	0.17	7		0.29	5
Two 2	2	0.27	8		0.10	6
Two 3 G	2	0.34	9		0.41	7
Three 1	3	0.28	9		0.20	11
Three 2 G	3	0.23	8		0.17	12
Four 1 G	4	0.23	9		0.12	8
Four 2	4	0.21	10		0.11	9
Five 1	5	0.19	8		0.15	12
Five 2	5	0.16	9		0.17	14
Ten 1	10	0.20	9		0.23	11
Ten 2	10	0.21	13		0.09	11
Ten 3	10	0.20	13		0.21	11
Ten 4 G	10	0.31	10		0.15	11
Ten 5	10	0.19	14		0.24	10
Twenty 1 G	20	0.28	12		0.24	10
Twenty 2 G	20	0.17	14		0.17	10
Twenty 3 G	20	0.17	14		0.17	10

Table 4.1

As can be seen from the Table 4.1, the MSKSCOPT optimizer appears to work better than the MOSEKOPT on the majority of the problems, One may be tempted to conclude that the homogeneous nonlinear solver, MSKSCOPT, is more efficient than the conic solver, MOSEKOPT. However, the situation is completely different as the dimension of the problems increases, which can be seen in the next two tables.

		CONIC (MOSEKOPT)			NON-CONIC (MSKSCOPT)	
Problem	**Number of Variables**	**CPU Time**	**Number of Iterations**		**CPU Time**	**Number of Iterations**
Thirty 1 G	30	0.15	14		0.98	108
Thirty 2G	30	0.35	15		0.93	98
Thirty 3G	30	0.24	14		0.77	89
Forty 2G	40	0.19	13		0.88	83
Forty 3G	40	0.28	14		0.57	66
Fifty 1 G	50	0.26	14		1.02	84
Fifty 2 G	50	0.29	16		0.54	57
Fifty 3 G	50	0.24	15		0.71	68
Fifty 4 G	50	0.23	14		0.73	90
Seventy 1 G	70	0.21	13		0.65	75
Seventy 2G	70	0.50	14		0.83	57

Table 4.2

This table shows that the conic quadratic solver, MOSEKOPT works better over the homogeneous nonlinear solver, MSKSCOPT on all problems. The results are similar in the next table.

		CONIC (MOSEKOPT)			NON-CONIC (MSKSCOPT)	
Problem	**Number of Variables**	**CPU Time**	**Number of Iterations**		**CPU Time**	**Number of Iterations**
Eighty 2 G	80	0.32	14		0.53	49
Eighty 3 G	80	0.22	15		0.20	11
Ninety 1 G	90	0.22	16		0.42	44
Ninety 2 G	90	0.22	14		0.55	72
Hundred 1 G	100	0.38	18		0.87	65
Hundred 2 G	100	0.27	14		0.55	45
Hundred 3 G	100	0.17	15		0.63	64
Hundred 4 G	100	0.20	17		0.12	11
Hundred 5 G	100	0.26	17		0.61	64

Table 4.3

It can be seen from the Table 4.3 that with the exception of two problems the conic quadratic solver MOSEKOPT works better over the homogeneous nonlinear solver, MSKSCOPT.

The results are summarized in the following table.

	CONIC (MOSEKOPT)			**NON-CONIC (MSKSCOPT)**	
Number of Variables	**Average CPU Time**	**Average Number of Iterations**		**Average CPU Time**	**Average Number of Iterations**
2 - 20	0.2241	10		0.1894	10
30 - 70	0.2662	14		0.7785	82
80 - 100	0.2379	15		0.4614	47

Table 4.4

This summary table shows that as the dimension of the problems increases the conic quadratic optimizer, MOSEKOPT solves the problems on average at least twice as fast as the homogeneous nonlinear optimizer, MSKSCOPT.

These results deserve further comment. The first impression is that probably the conic quadratic reformulation is not a good idea because it triples the number of variables, thus, increasing the dimension of the problem threefold. However, the numerical results show otherwise. The reason is due to the fact that the nonlinearity that appears in the objective function of the original problem has been transferred to the quadratic cones. Furthermore, IPMs handle cones very well and therefore they are very efficient and fast in solving conic quadratic problem, making up for the increased number of variables over the original problem. The iterations in IPM for conic quadratic reformulation are costlier but there are much fewer of them as opposed to those in the IPM for the original problem where iterations are cheaper but there are much more of them. Thus, the IPMs exploit the structure of the conic quadratic problem better than the structure of the original problem.

The results in this paper and many other numerical tests confirm the common view that there are problems that are considered "easy" when solved by IPMs. These are linear programming, semi-definite programming, and conic quadratic programming problems. The IPMs generally tend to perform better on these types of problems compared to general nonlinear programming problems even if they are convex.

5. CONCLUSION

In this paper we consider a convex optimization problem SOR defined by (1.1) which often appears in various applications such as stratified sampling and/or entropy problems.

The SOR is solved using two interior-point methods. First, a homogeneous interior point method of Andersen and Ye [2, 3] for monotone nonlinear complementarity problems was used to solve the KKT conditions of the original SOR listed in (3.1).

Second, a homogeneous conic quadratic IPM of Andersen et al [4] is used to solve the SOR as a reformulated conic quadratic problem stated in (2.2). Both of these algorithms were successfully implemented in optimization solver MOSEK [1] created by E. Andersen. The subroutine MOSEKOPT solves the conic quadratic problems, while the subroutine MSKSCOPT solves the nonlinear optimization problems directly (via KKT conditions).

The two approaches are then numerically tested using the above mentioned MOSEK subroutines on a set of randomly generated problems. They were compared by CPU time (in seconds) and the number of iterations showing that the second approach works better for problems with higher dimensions although the conic quadratic reformulation triples the number of variables. However, the IPM exploits the structure of the conic quadratic reformulation much better than the one of the original problem. The result confirms the experience from numerous numerical tests of variety of different optimization problems: The IPMs seem to work better whenever problem can be reformulated as a problem with the "nice" structure such as linear, conic quadratic or semi definite optimization problem.

As far as we are aware of it the approach of solving SOR as a reformulated conic quadratic problem is a novel approach not yet considered in the literature.

Acknowledgment

The authors would like to thank the referees and the editor for their valuable comments, which improved the earlier version of the paper, and for bringing the paper [6] to their attention.

REFERENCES

[1] Andersen, E. D., http://www.mosek.com.

[2] Andersen, E. D., and Ye, Y., "On a homogeneous algorithm for the monotone complementarity problem", *Mathematical Programming*, 84 (2) (1999) 375-399.

[3] Andersen, E.D., and Ye, Y., "A computational study of the homogeneous algorithm for large-scale convex optimization", *Computational Optimization and Applications,* 10 (1998) 243-269.

[4] Andersen, E.D., Roos, C., and Terlaky, T., "On implementing a primal-dual interior-point method for conic quadratic optimization", *Mathematical Programming,* 95 (2) (2003) 249-277.

[5] Bretthauer, K. M., and Shetty, B., "The nonlinear resource allocation problem", *Operations Research*, 43 (4) (1995) 670-683.

[6] Effati, S., and Moghadam, M.M., "Conversion of some classes of fractional programming to second order cone programming and solving it by potential reduction interior point method", *Applied. Mathematics and Computation,* 181 (1) (2006) 563-578.

[7] Karmarkar, N., A polynomial-time algorithm for linear programming, *Combinatorica*, 4 (1984) 373-395.

[8] Nesterov, Y. E., and Nemirovski, A. S., *Interior Point Polynomial Methods in Convex Programming,* SIAM Publications, Philadelphia, 1994.

[9] Slaughter, V., "Interior point methods for a class of stratified sampling problem", Master Thesis, Applied Mathematics, Georgia Southern University, December 2004.