# An Iterative Algorithm for the Conic Trust Region Subproblem
# 圆锥信任区子问题的迭代算法

Maziar Salahi*

Published online: 26 September 2016
在线发表：2016 年 9 月 26 日


*Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran
吉兰大学数学科学学院应用数学系，拉什特，伊朗

## Abstract

In Salahi (Optim Lett, 2016), using a variant of S-Lemma, an SOCP/SDP formulation is given for the conic trust region subproblem. Howeve, since currently SOCP/SDP formulation is not applicable for large scale problems, in this paper we present a generalized Newton algorithm for the problem using the parametrization approach of Dinkelbach. On several examples the new method and SOCP/SDP formulation are compared showing that the new method is extremely faster and moreover it is applicable for large scale instances.

## 摘要

在 Salahi (Optim Lett, 2016) 中，利用 S-引理的变体给出了圆锥信任区子问题的 SOCP/SDP 公式。然而，由于目前 SOCP/SDP 公式不适用于大规模的问题，在本文中，我们使用 Dinkelbach 的参数化方法为该问题提出了一种广义牛顿算法。在几个例子中，我们比较了新方法和 SOCP/SDP 公式，发现新方法的速度非常快，而且适用于大规模的实例。

**Keywords** Conic trust region subproblem · Newton method · S-lemma · Semidefinite program · Second order cone program

**关键词** 圆锥信任区子问题 · 牛顿法 · S-引理 · 半定程序 · 二阶锥体程序

## 1 Introduction

Conic Trust Region Subproblem (CTRS) introduced by Davidon [4] has been used as a subproblem within several nonlinear programming problems algo-

## 1 简介

Davidon 引入的圆锥信任区子问题（CTRS）[4] 已被用作若干非线性规划问题算法中的一个子问题 [5, 8, 9, 11, 16]。它的形式如下：

rithms [5, 8, 9, 11, 16]. It is given in the following form:

$$\min \frac{c^T x}{1 - a^T x} + \frac{x^T A x}{(1 - a^T x)^2}$$
$$\|x\|^2 \leq 1 \tag{1}$$

where $a, c \in \mathbb{R}^n$ such that $1 - a^T x > 0$ and $A$ is a symmetric but not necessarily positive definite. It is obvious that when $a = 0$, then (CTRS) reduces to the classical trust region subproblem [1, 3, 7, 12]. One can note that the objective function in (1) may not be bounded in the feasible region, thus modifications of it have been studied in the literature, for example see [10]. In this paper, like [13], we consider the following modification of (1):

其中 $a, c \in \mathbb{R}^n$ 使得 $1 - a^T x > 0$，并且 $A$ 是一个对称的但不一定是正定的。很明显，当 $a = 0$ 时，那么（CTRS）就会还原为经典的信任区子问题 [1, 3, 7, 12]。我们可以注意到，在可行区域内，(1) 中的目标函数可能不受约束，因此文献中研究了它的修改，例如见 [10]。在本文中，与 [13] 一样，我们考虑对 (1) 进行如下修改：

$$\min \frac{c^T x}{1 - a^T x} + \frac{x^T A x}{(1 - a^T x)^2}$$
$$\|x\|^2 \leq 1, 1 - a^T x \geq \beta \tag{2}$$

where $\beta$ is a positive constant. For this modification, obviously the feasible region is compact, thus attains the minimum. Recently, in [13] using a variant of S-Lemma and the parametrization approach of Dinkelbach, the author has proposed a two steps exact SOCP/SDP formulation for (CTRS). However it is not currently applicable for large scale instances of (CTRS). Thus in this paper, using the parametrization approach of Dinkelbach [14], we present an iterative generalized Newton algorithm which is able to solve large scale instances. The rest of the paper is organized as follows. In section “Preliminaries”, we review the existing results. The generalized Newton algorithm is presented in section “Generalized Newton Algorithm” and finally some numerical experiments are presented in section “Numerical Examples”.

其中 $\beta$ 是一个正常数。对于这种修改，显然可行区域是紧的，因此达到了最小值。最近，在 [13] 中，作者使用 S-引理的变体和 Dinkelbach 的参数化方法，提出了（CTRS）的两步精确 SOCP/SDP 公式。然而，它目前还不适用于（CTRS）的大规模实例。因此在本文中，利用 Dinkelbach 的参数化方法，我们提出了一种能够解决大规模实例的迭代式广义牛顿算法。本文的其余部分组织如下。在「预备工作」部分，我们回顾了现有的结果。在「广义牛顿算法」一节中介绍了广义牛顿算法，最后在「数值实例」一节中介绍了一些数值实验。

## 2 Preliminaries

First let us rewrite (2) as follows:

$$\min \frac{x^T\left(A-\frac{ca^T+ac^T}{2}\right)x+c^Tx}{1-2a^Tx+x^Taa^Tx}$$
$$\|x\|^2\le 1, 1-a^Tx\ge\beta \tag{3}$$

and further let use denote $q_1(x):=x^T\left(A-\frac{ca^T+ac^T}{2}\right)x+c^Tx$ and $q_2(x):=1-2a^Tx+x^Taa^Tx$.

The following result due to Dinkelbach [6] gives an equivalent approach to find the optimal solution of (CTRS) which also plays an important role in the rest of the paper.

**Theorem 1.** *The following two statements are equivalent:*

$$(i)\quad \min \frac{q_1(x)}{q_2(x)}=\alpha$$
$$\|x\|^2\le 1, 1-\alpha^Tx\ge\beta.$$
$$(ii)\quad F(\alpha):=\min\{q_1(x)-\alpha q_2(x)\}=0$$
$$\|x\|^2\le 1, 1-\alpha^Tx\ge\beta. \tag{4}$$

In [13], using a variant of S-Lemma and Theorem 1, the author has proposed the following SOCP/SDP formulation for (CTRS):

**Theorem 2.** *Suppose that the Slater condition holds for (CTRS). Then we have*

$$\min\left\{\frac{q_1(x)}{q_2(x)}:\|x\|^2\le 1, 1-a^Tx\ge\beta\right\}$$
$$=\max\left\{\lambda:\begin{bmatrix} A-\frac{ca^T+ac^T}{2}+\mu I-\lambda aa^T+\frac{ua^T+au^T}{2} & \frac{c}{2}+\lambda a+\frac{(\beta-1)u-u_0a}{2} \\ \frac{c^T}{2}+\lambda a^T+\frac{(\beta-1)u^T-u_0a^T}{2} & -\lambda-\mu-(\beta-1)\mu_0\end{bmatrix}\succeq 0, \mu\ge 0, (-u_0;u)\in L^{n+1}\right\} \tag{5}$$

*where $L^{n+1}$ denotes the Lorentz cone in $\mathbb{R}^{n+1}$.*

By solving (5) we have the optimal objective value of (CTRS), let us denote it by $\alpha^*$. Now, in order to find the optimal solution $x^*$ of (3), using Theorem 2

## 2 预备工作

首先让我们按如下方式重写 (2)：

并进一步表示 $q_1(x):=x^T\left(A-\frac{ca^T+ac^T}{2}\right)x+c^Tx$ 和 $q_2(x):=1-2a^Tx+x^Taa^Tx$。

以下 Dinkelbach 的结果 [6] 给出了一个寻找（CTRS）最优解的等效方法，并且在本文的其余部分也起到了重要作用。

**定理 1.** 以下两句话是等价的：

在 [13] 中，作者利用 S-引理和定理 1 的变体，为（CTRS）提出了以下 SOCP/SDP 表述：

**定理 2.** 假设 *Slater* 条件对（*CTRS*）成立。那么我们有

其中 $L^{n+1}$ 表示 $R^{n+1}$ 中的 *Lorentz* 锥体。

通过求解 (5)，我们得到了（CTRS）的最优目标值，让我们用 $\alpha^*$ 来表示。现在，为了找到 (3) 的最优解 $x^*$，利用定理 2，只需解决以下（eTRS），它在最优

it is sufficient to solve the following (eTRS) , which should have zero objective value at optimality:

时的目标值应该是零：

$$\begin{aligned} &\min q_1(x) - \alpha^* q_2(x) \\ &\|x\|^2 \leq 1, 1 - \alpha^T x \geq \beta \end{aligned} \tag{6}$$

In [13], the author has used the recent exact SOCP/SDP relaxation of [2] to solve (6) as follows:

在 [13] 中，作者利用最近对 [2] 的精确 SOCP/SDP 松弛来解决 (6)，具体如下。

$$\begin{aligned} &\min \operatorname{trace}\left(\left(A - \frac{ca^T + ac^T}{2} - \alpha^* aa^T\right) X\right) + (c + 2\alpha^* a)^T x - \alpha^* \\ &\operatorname{trace}(X) \leq 1, X \succeq xx^T, \\ &\|(1-\beta)x - Xa\| \leq 1 - \beta - a^T x \end{aligned}$$

As we see, in both steps we need to solve an SOCP/SDP formulation which is not currently applicable for large scale instances. Thus in the next section, using Theorem 1, we present an efficient iterative generalized Newton algorithm to solve large instances of (CTRS).

正如我们所看到的，在这两个步骤中，我们需要解一个目前并不适用于大规模的实例的 SOCP/SDP 公式。因此，在下一节中，利用定理1，我们提出了一种高效的迭代广义牛顿算法来解决（CTRS）的大实例。

## 3 Generalized Newton Algorithm

## 3 广义牛顿算法

The function $F$ in (2) is continuous, concave and strictly decreasing and has a unique root [17]. However, it is not differentiable, thus we can not apply Newton method to take advantage of its faster convergence in finding the root. Therefore, we propose an algorithm using its subgradient. The subgradient of $F$ at $\alpha_k$ is $-q_2(x^k)$, where

在 (2) 中的函数 $F$ 是连续的、凹的和严格下降的，并且有一个唯一的根 [17]。然而，它不是可微的，因此我们不能应用牛顿方法来利用它在寻找根时的快速收敛性。因此，我们提出了一种利用其次梯度的算法。$F$ 在 $\alpha_k$ 处的次梯度是 $-q_2(x^k)$，其中

$$\begin{aligned} x^k = \arg\min_{x \in \mathbb{R}^n} \{q_1(x) - \alpha_k q_2(x)\} \\ \|x\|^2 \leq 1, 1 - \alpha^T x \geq \beta \end{aligned}$$

The iterations of the new algorithm are as follows:

新算法的迭代过程如下。

$$\alpha_{k+1} := \alpha_k - \frac{F(\alpha_k)}{-q_2(x^k)} = \alpha_k - \frac{q_1(x^k) - \alpha_k q_2(x^k)}{-q_2(x^k)} = \frac{q_1(x^k)}{q_2(x^k)}$$

We call the algorithm based on these iterations, the Generalized Newton Algorithm which is outlined as follows.

我们把基于这些迭代的算法称为广义牛顿算法，其概要如下。

**Algorithm 1** Generalized Newton Algorithm

**Inputs:** $A \in \mathbb{R}^{n\times n}, a, c \in \mathbb{R}^n, \beta \in \mathbb{R}, k = 0$, starting point $\alpha_0 \in \mathbb{R}$, and an accuracy parameter $\varepsilon > 0$.

**while** $|F(\alpha_k)| \geq \varepsilon$ **do**

**Step 1:** Solve the following minimization subproblem to obtain a global optimum $x^k$:

$$\min_{x\in\mathbb{R}^n} \{q_1(x) - \alpha_k q_2(x)\}$$
$$\|x\|^2 \leq 1, q - a^T x \geq \beta.$$

**Step 2:** Calculate $F(\alpha_k) = q_1(x^k) - \alpha_k q_2(x^k)$.

**if** $|F(\alpha_k)| \leq \varepsilon$ **then**

Stop.

**else**

Set $\alpha_{k+1} := \dfrac{q_1(x^k)}{q_2(x^k)}$ and $k : k+1$.

**end if**

**end while**

**Algorithm 1** 广义牛顿算法

**输入：** $A \in \mathbb{R}^{n\times n}, a, c \in \mathbb{R}^n, \beta \in \mathbb{R}, k = 0$，起点 $\alpha_0 \in \mathbb{R}$，精度参数 $\varepsilon > 0$。

**while** $|F(\alpha_k)| \geq \varepsilon$ **do**

**步骤 1:** 解决以下最小化子问题，得到全局最优的 $x^k$。

$$\min_{x\in\mathbb{R}^n} \{q_1(x) - \alpha_k q_2(x)\}$$
$$\|x\|^2 \leq 1, q - a^T x \geq \beta.$$

**步骤 2:** 计算 $F(\alpha_k) = q_1(x^k) - \alpha_k q_2(x^k)$。

**if** $|F(\alpha_k)| \leq \varepsilon$ **then**

停止。

**else**

令 $\alpha_{k+1} := \dfrac{q_1(x^k)}{q_2(x^k)}$ 和 $k : k+1$。

**end if**

**end while**

As we see, the main computational cost of the algorithm is solving an extended trust region subproblem which has been the focus of several research recently [2, 15]. In this paper, we take advantage of the recent efficient algorithm developed in [15] to solve it.

我们看到，该算法的主要计算成本是求解一个扩展的信任区子问题，这也是最近几个研究的重点 [2, 15]。在本文中，我们利用最近在 [15] 中开发的高效算法来求解它。

## 4 Numerical Examples

In this section, we compare the Generalized Newton Algorithm with the two steps SOCP/SDP formulations of [13] on several randomly generated test problems. Test problems are generated using the following code and all computations are performed in MATLAB R2015b on a Mac mini 2.6GHz with 16GB of memory.

## 4 数值实例

在这一节中,我们在几个随机生成的测试问题上比较了广义牛顿算法与 [13] 的两步 SOCP/SDP 公式。测试问题是用以下代码生成的，所有的计算都是在 Mac mini 2.6GHz、16GB 内存上的 MATLAB R2015b 中进行的。

**Test problems generator 测试问题发生器**

- $n$ = input('enter the size of the problem =');
- $density1$ = input('enter the density of the matrix =');
- $density2$ = input('enter the density of the vectors =');
- $A = sprandsym(n, density1)$;
- $a = sprandn(n, 1, density2)$;
- $c = sprandn(n, 1, density2)$;
- $\beta = 0.1$;

First on several small dimensional examples, we compare the two approaches, then we solve large instances using the Generalized Newton Algorithm. For each dimension, we have generated 10 test problems and the computational results are summarized in Tables 1, 2, 3 and 4. In Table 1, we compare the Generalized Newton Algorithm with the SOCP/SDP formulation. In this table, we report the dimension of problem, CPU time in seconds, accuracy as defined in the sequel and $|F(\alpha)|$. The accuracy is the relative objective function difference as $\frac{\alpha^*-\alpha^{\text{best}}}{\alpha^{\text{best}}}$, where $\alpha^*$ is the computed solution by each method and $\alpha^{\text{best}}$ is the solution with the smallest objective value among two methods. In the rest of the tables, as we are just solving problem using Generalized Newton Algorithm, we report time and $|F(\alpha)|$. Moreover, we should note that in the Generalized Newton Algorithm, for all test problems we set initial $\alpha = 1$ and $\varepsilon = 10^{-6}$. In Table 4, in order to preserve the sparsity of the objective function matrix, we consider $a = c$.

首先在几个小维度的例子上,我们比较了这两种方法，然后我们用广义牛顿算法解决大的实例。对于每个维度，我们都产生了 10 个测试问题，计算结果汇总在表 1、2、3 和 4 中。在表 1 中，我们比较了广义牛顿算法和 SOCP/SDP 公式。在这个表格中，我们报告了问题的维度、以秒为单位的 CPU 时间、后文中定义的精度和 $|F(\alpha)|$。准确度是相对目标函数差值，即 $\frac{\alpha^*-\alpha^{\text{best}}}{\alpha^{\text{best}}}$，其中 $\alpha^*$ 是每种方法计算出的解，$\alpha^{\text{best}}$ 是两种方法中目标值最小的解。在其余的表格中，由于我们只是用广义牛顿算法来解决问题，我们报告了时间和 $|F(\alpha)|$。此外，我们应该注意到，在广义牛顿算法中，对于所有的测试问题，我们都设定初始 $\alpha = 1$，$\varepsilon = 10^{-6}$。在表 4 中，为了保持目标函数矩阵的稀疏性，我们考虑 $a = c$。

**Table 1** Comparison of SOCP/SDP formulation and the generalized Newton algorithm for $density1 = 0.1$ and $density2 = 1$
对于 $density1 = 0.1$ 和 $density2 = 1$ 的 SOCP/SDP 公式和广义牛顿算法的比较

| Dimension | Algorithm | Accuracy | Time (s) |
|---|---|---|---|
| $n = 100$ | Generalized Newton | 0 | 0.6 |
| | SOCP/SDP | $2.9 \times 10^{-9}$ | 3.9 |
| $n = 200$ | Generalized Newton | 0 | 0.9 |
| | SOCP/SDP | $1.3 \times 10^{-8}$ | 19.9 |
| $n = 300$ | Generalized Newton | 0 | 1.4 |
| | SOCP/SDP | $1.9 \times 10^{-9}$ | 84.1 |
| $n = 400$ | Generalized Newton | 0 | 2.2 |
| | SOCP/SDP | $4.2 \times 10^{-9}$ | 222.8 |

**Table 2** Results of generalized Newton algorithm for problems with $A = sprandsym(n, 0.001), a = sprandn(n, 1, 0.1); c = sprandn(n, 1, 0.1)$
广义牛顿算法对于 $A = sprandsym(n, 0.001), a = sprandn(n, 1, 0.1); c = sprandn(n, 1, 0.1)$ 的问题的结果

| Size | Time (s) | $\lvert F(\alpha^*)\rvert$ |
|---|---|---|
| 2000 | 6.4 | $1.5 \times 10^{-11}$ |
| 4000 | 16.4 | $1.3 \times 10^{-11}$ |
| 6000 | 33.4 | $7.1 \times 10^{-10}$ |
| 8000 | 61.6 | $5.5 \times 10^{-12}$ |

**Table 3** Results of generalized Newton algorithm for problems with $A = sprandsym(n, 0.0001), a = sprandn(n, 1, 0.01); c = sprandn(n, 1, 0.01)$
广义牛顿算法对于 $A = sprandsym(n, 0.0001), a = sprandn(n, 1, 0.01); c = sprandn(n, 1, 0.01)$ 问题的结果

| Size | Time (s) | $\lvert F(\alpha^*)\rvert$ |
|---|---|---|
| 2000 | 4.6 | $4.5 \times 10^{-8}$ |
| 4000 | 14.1 | $8.3 \times 10^{-7}$ |
| 6000 | 19.4 | $7.3 \times 10^{-10}$ |
| 8000 | 31.1 | $7.4 \times 10^{-10}$ |
| 10,000 | 56.9 | $3.1 \times 10^{-9}$ |

**Table 4** Results of generalized Newton algorithm for problems with $A = sprandsym(n, 0.0001), a = sprandn(n, 1, 0.01); c = a$
广义牛顿算法对于 $A = sprandsym(n, 0.0001), a = sprandn(n, 1, 0.01); c = a$ 的问题的结果

| Size | Time (s) | $\lvert F(\alpha^*)\rvert$ |
|---|---|---|
| 2000 | 2.1 | $4.5 \times 10^{-11}$ |
| 4000 | 4.3 | $2.7 \times 10^{-9}$ |
| 6000 | 7.1 | $1.3 \times 10^{-11}$ |
| 8000 | 9.2 | $1.4 \times 10^{-10}$ |
| 10,000 | 16.7 | $1.1 \times 10^{-7}$ |
| 15,000 | 29.3 | $1.6 \times 10^{-7}$ |
| 20,000 | 61.3 | $1.4 \times 10^{-7}$ |

As we see from Table 1, the Generalized Newton Algorithm finds high accuracy solutions compared to the SOCP/SDP formulation in much shorter time.

正如我们从表 1 中看到的，与 SOCP/SDP 公式相比，广义牛顿算法在更短的时间内找到了高精度的解。此外，表 2、3 和 4 中的结果也表明了新算法在解决

Moreover, results in Tables 2, 3, and 4 also show the capability of the new algorithm on solving large instances where SOCP/SDP formulation is not applicable.

SOCP/SDP 公式不适用的大型实例方面的能力。

**Acknowledgements** The author would like to thank University of Guilan for supporting his sabbatical leave 2015-2016, hosted by Prof. H. Wolkowicz at the Department of Combinatorics and Optimization, University of Waterloo.

**致谢** 作者要感谢吉兰大学支持他在滑铁卢大学组合学和优化系的 H. Wolkowicz 教授主持的 2015-2016 年公休假。

## Reference 参考文献

[1] Adachi, S., Iwata, S., Nakatsukasa, Y., Takeda, A.: Solving the trust region subproblem by a generalized eigenvalue problem. In: Mathematical Engineering Technical Report (METR 2015-14), Department of Mathematical Informatics Graduate School of Information Science and Technology the University of Tokyo

[2] Burer, S., Anstreicher, K.M.: Second-order-cone constraints for extended trust-region subproblems. SIAM J. Optim. *23*(1), 432–451 (2013)

[3] Conn, A.R., Gould, N.I., Toint, P.L.: Trust Region Methods. SIAM, Philadelphia (2000)

[4] Davidon, W.C.: Conic approximations and collinear scalings for optimizers. SIAM J. Numer. Anal. *17*, 268–281 (1980)

[5] Di, S., Sun, W.Y.: A trust region method for conic model to solve unconstrained optimization. Optim. Methods Softw. *6*, 237–263 (1996)

[6] Dinkelbach, W.:On nonlinear fractional programming. Manag. Sci. *13*, 492–498 (1967)

[7] Fortin, C., Wolkowicz, H.: The trust region subproblem and semidefinite programming. Optim. Methods Softw. *19*(1), 41–67 (2004)

[8] Han, Q., Sun, W., Han, J., Sampaio, R.J.B.: An adaptive conic trust-region method for unconstrained optimization. Optim. Methods Softw. *20*(6), 665–677 (2005)

[9] Lu, X., Ni, Q.: A quasi-Newton trust region method with a new conic model for the unconstrained optimization. Appl. Math. Comput. *204*, 373–384 (2008)

[10] Ni, Q.: Optimality conditions for trust-region subproblems involving a conic model. SIAM J. Optim. *15*(3), 826–837 (2005)

[11] Qua, S.J., Jiang, S.D., Zhu, Y.: A conic trust-region method and its convergence properties. Comput. Math. Appl. *57*, 513–528 (2009)

[12] Rendl, F., Wolkowicz, H.: A semidefinite framework for trust region subproblems with applications to large scale minimization. Math. Program. *77*(1), 273–299 (1997)

[13] Salahi, M.: Exact two steps SOCP/SDP formulation for a modified conic trust region subproblem. Optim. Lett. (2016). doi:10.1007/s11590-016-1080-y

[14] Salahi, M., Fallahi, S.: Parametric approach for solving quadratic fractional optimization with a linear and a quadratic constraint. Comput. Appl. Math. *35*(1), 439–446 (2016)

[15] Salahi, M., Taati, A., Wolkowicz, H: Local nonglobal minima for solving large scale extended trust region subproblems. Comput. Optim. Appl. (2016). doi:10.1007/s10589-016-9867-4

[16] Sun, W.Y., Yuan, Y.X.: A conic trust-region method for nonlinearly constrained optimization. Ann. Oper. Res. *103*, 175–191 (2001)

[17] Zhang, A., Hayashi, S.: Celis-Dennis-Tapia based approach to quadratic fractional programming problems with two quadratic constraints. Numer. Algebra Control Optim. *1*, 83–98 (2011)