Journal of Mathematical Research & Exposition
Jan., 2010, Vol. 30, No. 1, pp. 159–166
DOI:10.3770/j.issn:1000-341X.2010.01.015
Http://jmre.dlut.edu.cn

Truncated Smoothing Newton Method for l_∞ Fitting Rotated Cones

Yu XIAO[1], Bo YU[1,*], De Lun WANG[2]
1. *School of Mathematical Sciences, Dalian University of Technology, Liaoning* 116024*, P. R. China;*
2. *School of Mechanical Engineering, Dalian University of Technology, Liaoning* 116024*, P. R. China*

Abstract In this paper, the rotated cone fitting problem is considered. In case the measured data are generally accurate and it is needed to fit the surface within expected error bound, it is more appropriate to use l_∞ norm than l_2 norm. l_∞ fitting rotated cones need to minimize, under some bound constraints, the maximum function of some nonsmooth functions involving both absolute value and square root functions. Although this is a low dimensional problem, in some practical application, it is needed to fitting large amount of cones repeatedly, moreover, when large amount of measured data are to be fitted to one rotated cone, the number of components in the maximum function is large. So it is necessary to develop efficient solution methods. To solve such optimization problems efficiently, a truncated smoothing Newton method is presented. At first, combining aggregate smoothing technique to the maximum function as well as the absolute value function and a smoothing function to the square root function, a monotonic and uniform smooth approximation to the objective function is constructed. Using the smooth approximation, a smoothing Newton method can be used to solve the problem. Then, to reduce the computation cost, a truncated aggregate smoothing technique is applied to give the truncated smoothing Newton method, such that only a small subset of component functions are aggregated in each iteration point and hence the computation cost is considerably reduced.

Keywords rotated cone fitting; nonsmooth optimization; minimax problem; l_∞ fitting; smoothing Newton method.

Document code A
MR(2000) Subject Classification 90C30; 65H10
Chinese Library Classification O221.2

1. Introduction

Conical shaped parts, whole cones or sections of cones, are widely used in various mechanical equipments. Remodeling conical shaped parts by reverse engineering is a cones fitting problem: collecting data with three-coordinate measuring machine, and then find a cone that best fits the collected data. While collecting data with three-coordinate measuring machine, the measured coordinate system may departure from the coordinate system whose origin is located on the cone apex and z-axis parallels to the cone axis. Hence, rotations and translation operations to

Received July 26, 2009; Accepted November 2, 2009
Supported by the National Natural Science Foundation of China (Grant No. 10671029) and the Research Fund for the Doctoral Programme of Higher Education (Grant No. 20060141029).
* Corresponding author
E-mail address: fisherxiao@gmail.com (Y. XIAO); yubo@dlut.edu.cn (B. YU); dlunwang@dlut.ed.cn (D. L. WANG)

measured data should be taken into account, and thus the rotated cone fitting problem arises. If we fit the rotated cone in l_2 norm, it becomes a nonlinear least square problem. In case the measured data are generally accurate and it is needed to fit the surface within expected error bound, it is more appropriate to use l_∞ norm than l_2 norm, as that was pointed out in [1] and [8].

Let $(u_j, v_j, w_j) \in R^3$, $i = 1, \ldots, m$, be the measured data. Define the transformation matrix

$$T(x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_1 & x_2 & x_3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos x_4 & \sin x_4 & 0 \\ 0 & -\sin x_4 & \cos x_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos x_5 & 0 & -\sin x_5 & 0 \\ 0 & 1 & 0 & 0 \\ \sin x_5 & 0 & \cos x_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

then $(u_j, v_j, w_j, 1)$ is transformed to $(\bar{u}_j, \bar{v}_j, \bar{w}_j, 1) = (u_j, v_j, w_j, 1)T(x)$, i.e.,

$$\begin{aligned} \bar{u}_j(x) &= (u_j + x_1)\cos x_5 + ((v_j + x_2)\sin x_4 + (w_j + x_3)\cos x_4)\sin x_5, \\ \bar{v}_j(x) &= (v_j + x_2)\cos x_4 - (w_j + x_3)\sin x_4, \\ \bar{w}_j(x) &= -(u_j + x_1)\sin x_5 + ((v_j + x_2)\sin x_4 + (w_j + x_3)\cos x_4)\cos x_5. \end{aligned}$$

The fitting cone V is defined as $\bar{w} = r\sqrt{\bar{u}^2 + \bar{v}^2}$, $r \geq 0$. Denote the unknown variables x, r as x, and residual errors

$$\tilde{f}_j(x) = \left| \bar{w}_j(x) - x_6 \sqrt{\bar{u}_j^2(x) + \bar{v}_j^2(x)} \right|, \tag{1}$$

the fitting problem can be stated as:

$$\begin{aligned} &\min \quad \{f(x) = \max_{1 \leq j \leq m} \tilde{f}_j(x)\}, \\ &\text{s.t.} \quad x \in \mathscr{P} = \{x | -\pi \leq x_4 \leq \pi, -\pi \leq x_5 \leq \pi, 0 \leq x_6\}. \end{aligned} \tag{2}$$

It can be seen from (1) that $\tilde{f}_j(x)$ is nonsmooth function including the absolute value function and the square root function. Moreover, when large amount of measured data are to be fitted to a rotated cone, the number of components in the maximum function is large. Nonsmoothness and large number of components make the optimization problem complex and need to develop efficient solution methods.

Different algorithms have been proposed to solve minimax problems, such as subgradient methods (see [15] for details), SQP methods [13, 17, 21], bundle-type methods [5, 6, 9, 22], smooth approximation methods [2, 3, 7, 11, 14, 19] and etc.

In [10], for the nonsmooth max-function $f(x) = \max_{j \in \mathbf{q}} f_j(x)$, where $\mathbf{q} = \{1, \ldots, q\}$ and $f_j(x)$, $j = 1, \ldots, m$, are smooth functions, based on the Jaynes' maximum entropy principles, Li proposed the following aggregate function with parameter $p > 0$,

$$F_p(x) = \frac{1}{p} \ln \left(\sum\nolimits_{j \in \mathbf{q}} \exp(p f_j(x)) \right), \tag{3}$$

which is a smooth uniform and monotonic approximation to $f(x)$. However, the aggregate function is a single smooth but complex function, and its gradient and Hessian calculations are time-consuming. In [18], a truncated aggregate function was proposed and combined with stabilized Newton method to form a truncated aggregate smoothing Newton algorithm (TASN) for solving unconstrained minimax problems. For any given $\tilde{x} \in R^n$ and constant $\mu > 0$, denote

$$\bar{\mathbf{q}} = \{j \mid f(\tilde{x}) - f_j(\tilde{x}) \le \mu, j \in \mathbf{q}\}. \tag{4}$$

A truncated aggregate function with respect to $\bar{\mathbf{q}}$ was defined as

$$F_p^{\bar{\mathbf{q}}}(x) = \frac{1}{p} \ln \left(\sum\nolimits_{j \in \bar{\mathbf{q}}} \exp\left(p f_j(x)\right) \right). \tag{5}$$

At each iteration, only a small subset of the components in the max-function are aggregated, hence the number of gradient and Hessian calculations is reduced dramatically. The subset is adaptively updated with some truncating criterions, concerning only with computation of function values and not their gradients or Hessians, to guarantee the global convergence and locally quadratic convergence with as few computational cost as possible.

We can not utilize TASN algorithm in [18] directly to solve (2), for that $\tilde{f}_j$'s are not smooth functions. Here, we try to extend the idea of TASN algorithm to solve the rotated cone fitting problems, which minimize, under some bound constraints, the maximum function of some nonsmooth functions involving both absolute value and square root functions. At first, combining aggregate smoothing technique to the maximum function as well as the absolute value function and a smoothing function to the square root function, a monotonic and uniform smooth approximation to the objective function is constructed. Using the smooth approximation, a smoothing Newton method can be used to solve the problem. Then, to reduce the computation cost, the truncated aggregate smoothing technique is applied to give the truncated smoothing Newton method (TSN).

2. The TSN algorithm for l_∞ fitting rotated cone

Firstly, some deformations for (2) are necessary. Replace $\tilde{f}_j(x) = |f_j(x)|$ by $\max\{f_j(x), -f_j(x)\}$, then the fitting problem (2) is equivalent to

$$\begin{aligned} \min \quad & \{f(x) = \max_{1 \le j \le 2m} f_j(x)\}, \\ \text{s.t.} \quad & x \in \mathscr{P}, \end{aligned} \tag{6}$$

where

$$f_j(x) = \begin{cases} \bar{w}_j(x) - x_6 \sqrt{\bar{u}_j^2(x) + \bar{v}_j^2(x)}, & j = 1, \ldots, m, \\ -f_{j-m}(x), & j = m+1, \ldots, 2m. \end{cases} \tag{7}$$

Let $g(x) = \max\{g_1(x), \ldots, g_6(x)\}$, where $g_1(x) = x_4 - \pi$, $g_2(x) = -x_4 - \pi$, $g_3(x) = x_5 - \pi$, $g_4(x) = -x_5 - \pi$, $g_5(x) = -x_6$, $g_6(x) = 0$. Use penalty function with penalty parameter $C > 0$ to transform (8) into the following unconstrained programming problem

$$\min \left\{ H(x) = \left\{ \max_{1 \le j \le 2m} f_j(x) + C g(x) \right\} \right\}. \tag{8}$$

Proposition 2.1 *The functions $f_j(x)$, $j = 1, \ldots, 2m$, are locally Lipschitz at any $x \in R^5$.*

The following proposition concerning Clarke's subdifferential in nonsmooth optimization from [4] or [12] gives the first-order optimality condition for (8).

Proposition 2.2 *If (8) attains the extremum at x^*, then*

$$0 \in \partial H(x^*) \subset \underset{j\in\mathbf{q}(x^*)}{\mathrm{conv}}\{\partial f_j(x^*)\} + C \underset{s\in\mathbf{s}(x^*)}{\mathrm{conv}}\{\partial g_s(x^*)\},$$

where $\mathbf{q}(x^*) = \{j \in \mathbf{q} = \{1,\dots,2m\} \mid f_j(x^*) = f(x^*)\}$ *and* $\mathbf{s}(x^*) = \{s \in \mathbf{s} = \{1,\dots,6\} \mid g_s(x^*) = g(x^*)\}$.

Since $f_j(x)$ in (2) is nonsmooth in $\mathscr{D}_j = \{x|\bar{u}_j^2(x) + \bar{v}_j^2(x) = 0\}$, we try to smooth it by the following function

$$f_{j,p}(x) = \begin{cases} \bar{w}_j(x) - x_6\sqrt{\bar{u}_j^2(x) + \bar{v}_j^2(x) + 1/p} + x_6/\sqrt{p}, & j = 1,\dots,m; \\ x_6\sqrt{\bar{u}_j^2(x) + \bar{v}_j^2(x) + 1/p} - \bar{w}_j(x), & j = m+1,\dots,2m. \end{cases} \tag{9}$$

Proposition 2.3 *For any $p > 0$, $f_{j,p}(x)$ defined as (9) is twice continuously differentiable and, for any given $x \in \mathscr{P}$, $f_{j,p}(x)$ is monotonically decreasing with respect to $p > 0$, and $f_{j,p}(x) \to f_j(x)$ as $p \to \infty$.*

To smooth the objective function $H(x)$ in (8), the following aggregate function can be used

$$H_p(x) = F_p(x) + CG_p(x),$$

where

$$F_p(x) = \frac{1}{p}\ln\Big(\sum\nolimits_{j\in\mathbf{q}} \exp\left(pf_{j,p}(x)\right)\Big),\ G_p(x) = \frac{1}{p}\ln\Big(\sum\nolimits_{s\in\mathbf{s}} \exp\left(pg_s(x)\right)\Big).$$

We give some properties on the smoothing function $F_p(x)$. The proofs can be induced from Proposition 2.4 in [20] and the above Proposition 2.3.

Proposition 2.4 *(i) For any $p > 0$, $F_p(x)$ is twice continuously differentiable and,*

$$\nabla F_p(x) = \sum_{j\in\mathbf{q}} \zeta_{j,p}(x)\nabla f_{j,p}(x)$$

where

$$\zeta_{j,p}(x) = \frac{\exp\left(pf_{j,p}(x)\right)}{\sum_{j\in\mathbf{q}}\exp\left(pf_{j,p}(x)\right)} \in (0,1), \sum_{j\in\mathbf{q}} \zeta_{j,p}(x) = 1.$$

(ii) For any $x \in \mathscr{P}$, $F_p(x)$ is monotonically decreasing with respect to $p > 0$;

(iii) When $(x,t) \to (x^*,0)(\|x^*\| \neq \infty)$, it has $F_p(x) \to f(x^*)$, and $\zeta_{j,p}(x) \to 0(j \notin \mathbf{q}(x^*))$.

Now, the following algorithm is given for solving problem (8).

Algorithm 1 (Truncated Smoothing Newton Algorithm)
Data. $x^0 \in \mathscr{P}$.
Parameters. $C > 0$, $p_0 > 0$, $\hat{p} \gg 1$; α, β, $\kappa_1 \in (0,1)$; $\eta \in (0,(1-\alpha)\kappa_1^2/32)$, $\bar{\delta} > 0$; γ, ω are sufficient big numbers; functions $\epsilon_a(p)$, $\epsilon_b(p)$, $\tau(p)$, $\varepsilon(p)$: $(0,\infty) \to (0,\infty)$, satisfying $\epsilon_b(p) \geq \epsilon_a(p) > \tau(p)$ for all $p > 0$, $\lim_{p\to+\infty}\tau(p) = 0$, $\varepsilon_1(p) = \eta\tau(p)$, $\varepsilon_2(p) > 0$.

Step 1. Set $i = 0$, $k = 0$, $s = 1$, $x^{k,i} = x^0$.

Step 2. Compute μ

$$\mu = \frac{1}{p_k}\ln\Big(\max\Big\{1, (2\gamma - \varepsilon_1)(q-1)/\varepsilon_1, (2\omega + 6p_k\gamma^2 - \varepsilon_2)(q-1)/\varepsilon_2\Big\}\Big), \tag{10}$$

then compute $\bar{\mathbf{q}}$ according to (4) and compute

$$H_{p_k}^{\bar{\mathbf{q}}}(x^{k,i}) = F_{p_k}^{\bar{\mathbf{q}}}(x^{k,i}) + CG_{p_k}(x^{k,i}).$$

If $\|\nabla H_{p_k}^{\bar{\mathbf{q}}}(x^{k,i})\| > \tau(p_k)$, go to Step 3, else go to Step 8.

Step 3. Compute $B(x^{k,i})$

$$B(x^{k,i}) = \theta(x^{k,i})I + \nabla^2 H_{p_k}^{\bar{\mathbf{q}}}(x^{k,i}), \tag{11}$$

where $\theta(x) = \max\{0, \bar{\delta} - e(x)\}$ with $e(x)$ denoting the smallest eigenvalue of $\nabla^2 H_p^{\bar{\mathbf{q}}}(x)$. Then compute Cholesky factor R such that $B(x^{k,i}) = RR^{\mathrm{T}}$, and compute the reciprocal condition number $c(R)$ of R. If $c(R) \geq \kappa_1$, go to Step 4, else go to Step 5.

Step 4. Compute $h_{k,i} = -B(x^{k,i})^{-1}\nabla H_{p_k}^{\bar{\mathbf{q}}}(x^{k,i})$, go to Step 6.

Step 5. Set $h_{k,i} = -\nabla H_{p_k}^{\bar{\mathbf{q}}}(x^{k,i})$.

Step 6. Compute the step length $\lambda_{k,i} = \beta^l$, where $l \geq 0$ is the smallest integer satisfying

$$H_{p_k}(x^{k,i} + \lambda_{k,i}h_{k,i}) - H_{p_k}(x^{k,i}) \leq \alpha\lambda_{k,i}\langle\nabla H_{p_k}^{\bar{\mathbf{q}}}(x^{k,i}), h_{k,i}\rangle. \tag{12}$$

Step 7. Set $x^{k,i+1} = x^{k,i} + \lambda_{k,i}h_{k,i}$, $i = i + 1$. Compute $\mu, \bar{\mathbf{q}}$. If

$$\|\nabla H_{p_k}^{\bar{\mathbf{q}}}(x^{k,i})\| \leq \tau(p_k), \tag{13}$$

go to Step 8, else go to Step 3.

Step 8. If $s = 1$, compute p^* such that

$$\epsilon_a(p_k) \leq \|\nabla H_{p^*}^{\bar{\mathbf{q}}}(x^{k,i})\| \leq \epsilon_b(p_k), \tag{14}$$

go to Step 9, else set $p_{k+1} = s(k+2)$, $k = k+1$, $i = 0$, go to Step 2.

Step 9. If $p^* \leq \hat{p}$, set $p_{k+1} = \max\{p^*, p_k + 1\}$, $k = k+1$, $i = 0$, go to Step 2, else set $s = \max\{2, (\hat{p}+1)/(k+1)\}$, $p_{k+1} = \max\{p_k + 1, s(k+2)\}$, $k = k+1$, $i = 0$, go to Step 2. □

Let $\tilde{H}_p(x) = \max_{j\in\mathbf{q}} f_{j,p}(x) + Cg(x)$, $\Omega = \{x|\tilde{H}_{p_0}(x) \leq H_{p_0}(x^0),\}$. Then, it has

Proposition 2.5 *For any $x^0 \in R^n$ and $p_0 > 0$, it has $x^0 \in \Omega$; Define $\Omega_p = \{x|\tilde{H}_p(x) \leq H_{p_0}(x^0)\}$, then for any $p > p_0$, it has $\Omega_p \cap \mathscr{P} \subset \Omega$.*

Let $\gamma_p(x) = \max\{\|\nabla f_{j,p}(x)\| \mid j \in \mathbf{q}\}$, $\omega_p(x) = \max\{\|\nabla^2 f_{j,p}(x)\| \mid j \in \mathbf{q}\}$; Given the following assumptions:

Assumption 2.6 *The level set Ω is a bounded closed set.*

Under Assumption 2.6, it has the following proposition:

Proposition 2.7 *$\partial f_{j,p}$ is bounded in $\Omega \times [p_0, +\infty]$.*

Assumption 2.8 *γ is sufficiently big number such that $\gamma \geq \max\{\gamma_p(x), \mid x \in \Omega, p \in [p_0, \infty)\}$.*

With Assumptions 2.6 and 2.8, the similar conclusion of Corollary 2.3 in [18] still holds, i.e., for any $x \in \Omega$, $p > 0$, $\varepsilon_1(p) > 0$ and $\varepsilon_2(p) > 0$, if $\bar{\mathbf{q}}$ is set as (4) and (10), it has

$$\|\nabla H_p(x) - \nabla H_p^{\bar{\mathbf{q}}}(x)\| \leq \varepsilon_1(p).$$

Then, together with Propositions 2.2–2.5 and 2.7, we have

Lemma 2.9 *Suppose that sequences $\{p_k\}$ and $\{x^{1,i}\}$, $\{x^{2,i}\}, \ldots, \{x^{k,i}\}, \ldots$, are generated by Algorithm 1. Under Assumptions 2.6 and 2.8, for any $x^{k,i} \in \Omega$ such that $\|\nabla H^{\bar{\mathbf{q}}}_{p_k}(x^{k,i})\| > \tau(p_k)$, it has*

(i) $(1-\eta)\|\nabla H^{\bar{\mathbf{q}}}_{p_k}(x^{k,i})\| \le \|\nabla H_{p_k}(x^{k,i})\| \le (1+\eta)\|\nabla H^{\bar{\mathbf{q}}}_{p_k}(x^{k,i})\|$;

(ii) $\lambda_{k,i}$ *is computed using a finite number of function evaluations;*

(iii) $x^{k,i+1} \in \Omega$.

Lemma 2.10 *Suppose that sequences $\{p_k\}$ and $\{x^{1,i}\}$, $\{x^{2,i}\}, \ldots, \{x^{k,i}\}, \ldots$, are generated by Algorithm 1. Under Assumptions 2.6 and 2.8,*

(i) For any k, the sequence $\{x^{k,i}\}$ is finite, i.e., there exists a $i_k \in N$ such that (13) holds for $i = i_k$;

(ii) The sequence $\{p_k\}$ is strictly monotone increasing and $p_k \to \infty$ as $k \to \infty$.

The proofs of Lemmas 2.9–2.10 are similar with that in [18], and omitted here.

Theorem 2.11 *Under Assumptions 2.6 and 2.8, there exists an infinite subsequence $N' \subset N$ such that $x^{k,i_k} \to x^*$ as $k \to \infty$ with $k \in N'$, and*

$$0 \in \operatorname*{conv}_{j\in\mathbf{q}(x^*)}\{\partial f_j(x^*)\} + C \operatorname*{conv}_{s\in\mathbf{s}(x^*)}\{\partial g_s(x^*)\}.$$

Moreover, if $x^* \notin \mathscr{D} = \bigcup_{1\le j\le m} \mathscr{D}_j$, then $0 \in \partial H(x^*)$.

Proof From Lemma 2.9, it has $x^{k,i_k} \in \Omega$ for any $k \in N$, and hence there exists an infinite subsequence $N' \subset N$ such that $x^{k,i_k} \to x^*$ as $k \to \infty$ with $k \in N'$. Since $\|\nabla H^{\bar{\mathbf{q}}}_{p_k}(x^{k,i_k})\| \le \tau(p_k)$ and $\tau(p_k) \to 0$ as $k \to \infty$, we have $\|\nabla H^{\bar{\mathbf{q}}}_{p_k}(x^{k,i_k})\| \to 0, \quad$ as $k \to \infty$. Together with

$$\|\nabla H_{p_k}(x^{k,i_k}) - \nabla H^{\bar{\mathbf{q}}}_{p_k}(x^{k,i_k})\| \le \eta\tau(p_k),$$

it follows that $\|\nabla H_{p_k}(x^{k,i_k})\| \to 0$, as $k \to \infty$ with $k \in N'$, i.e.,

$$\lim_{\substack{k\to\infty\\ k\in N'}} \nabla H_{p_k}(x^{k,i_k}) = \sum_{j\in\mathbf{q}} \zeta_{j,p_k}(x^{k,i_k})\nabla f_{j,p_k}(x^{k,i_k}) + C\sum_{s\in\mathbf{s}} \xi_{s,p_k}(x^{k,i_k})\nabla g_s(x^{k,i_k}) = 0,$$

where $\zeta_{j,p_k}(x^{k,i_k})$ is defined as that in Proposition 2.4 and $\xi_{s,p_k}(x^{k,i_k}) = \frac{\exp(p_k g_s(x^{k,i_k}))}{\sum_{s\in\mathbf{s}}\exp\left(p_k g_s(x^{k,i_k})\right)}$. By the boundedness of $\nabla f_{j,p_k}(x^{k,i_k})$, it has

$$\lim_{\substack{k\to\infty\\ k\in N'}} \nabla f_{j,p_k}(x^{k,i_k}) = \eta_j \in \partial f_j(x^*).$$

Now, we have

$$\lim_{\substack{k\to\infty\\ k\in N'}} \nabla H_{p_k}(x^{k,i_k}) = \sum_{j\in\mathbf{q}(x^*)} \hat{\zeta}_j\eta_j + C\sum_{s\in\mathbf{s}(x^*)} \hat{\xi}_s\nabla g_s(x^*) = 0,$$

where $\hat{\zeta}_j$, $\hat{\xi}_s \ge 0$, $\sum_{j\in\mathbf{q}(x^*)}\hat{\zeta}_j = \sum_{s\in\mathbf{s}(x^*)}\hat{\xi}_s = 1$, hence

$$0 \in \operatorname*{conv}_{j\in\mathbf{q}(x^*)}\{\partial f_j(x^*)\} + C \operatorname*{conv}_{s\in\mathbf{s}(x^*)}\{\partial g_s(x^*)\}.$$

Moreover, if $x^* \notin \mathscr{D} = \bigcup_{1\le j\le m} \mathscr{D}_j$, then $f_j(x)$ is smooth at x^* for all $j \in \mathbf{q}$, and

$$\lim_{p\to\infty, x\to x^*} \nabla f_{j,p}(x) = \nabla f_j(x^*),$$

hence $0 \in \partial H(x^*)$. □

The above theorem proves the convergence of Algorithm 1. Moreover, for the fixed p, Algorithm 1 is locally quadratic convergent under some conditions, and the detailed analysis is similar to Theorem 3.5 in [18].

3. Numerical experiment

We have implemented truncated smoothing Newton algorithm in MATLAB programming language and compared it with SQP method (SQP method is implemented by calling matlab function fminimax directly. Theoretically, SQP method can only be used to solve minimax problems with smooth component functions, we tried to use it to solve the rotated cone fitting problem with nonsmooth component functions and it worked.) and smoothing Newton method (SN) which substitutes exact aggregate function smoothing for the truncated aggregate function in Algorithm 1.

Parameters in Algorithm 1 are set as $\alpha = 0.5$, $\beta = 0.8$, $\hat{p} = 10^5 \ln q$, $\kappa_1 = 10^{-7}$, $\kappa_2 = 10^{30}$, $\kappa_3 = 1000\hat{p}$, $p_0 = 1$, $(\epsilon_a, \epsilon_b) = (0.1, 0.45)$, $\gamma = 10^2, \omega = 10, \tau^2(p) = \min\{10^{-1}, 1000/p\}, \varepsilon_1 = 10^{-2}, \varepsilon_2 = 10^{-1}$, $\bar{\delta} = 0.02$, $C = 100$.

All the computations are done by running MATLAB 7.6.0 on a laptop with AMD Turion(tm) 64 ×2 Mobile Technology TL-58 CPU 1.9 GHz and 896M memory, and only the matlab functions chol, rcond and eigs (A,1, 'SA') are utilized to compute the Cholesky decomposition, reciprocal condition number and the smallest eigenvalue. The results are listed in the following tables, where x^* denotes the final approximate solution point, f^* is the maximum residual error obtained by x^*, time is the CPU time in seconds. Here, we also give the gradient computing number in Algorithm 1 and SN, denoted as N, except for SQP which is implemented by calling matlab internal function and difficult to count the amount.

In this section, we apply our algorithm to two fitting problems. The first one appears in manufactural parts matching and another is an artificial problem.

Example 3.1 This metrical data set is from input shaft of aerogenerator's accelerator.

Method	x^*	f^*	Time	N
TSN	$(2.955782, 0.441657, \ldots, 89.593986)$	0.748557	3.167	85525
SN	$(2.955844, 0.441514, \ldots, 89.593986)$	0.748557	5.828	416720
SQP	$(2.939973, 0.505359, \ldots, 89.892013)$	0.751025	6.143	–

Table 1 The numerical results of Example 3.1, $m = 5209$, $x^0 = (3, 0, -1558, 3.14, 0, 2)^{\mathrm{T}}$

Example 3.2 We also test our algorithm for the artificial rotated cone data points which are generated as that in [16]. At first, produce points $(\tilde{x}_j, \tilde{y}_j, \tilde{z}_j)$, $i = 1, \ldots, m$, on an unrotated cone by defining $\tilde{x}_j = r_j \tan(\pi/6)\cos\gamma_j, \tilde{y}_j = r_j \tan(\pi/6)\sin\gamma_j, \tilde{z}_j = r_j$, where r_j and γ_j are equally distributed pseudorandom numbers in $[0.1, 10]$ and $[0, 2\pi]$, respectively. Then, perturb $\tilde{z}_j$ by adding error item which follows Gaussian distribution $N(0, 0.3)$, and make rotations and translation to obtain the final data $(u_j, v_j, w_j, 1) = (\tilde{x}_j, \tilde{y}_j, \tilde{z}_j, 1)T^{-1}(2.1, -1.4, 1.3, \pi/20, \pi/25)$.

Method	x^*	f^*	Time	N
TSN	$(-2.146208, 1.408664, \ldots, 1.738858)$	0.784598	4.012	100344
SN	$(-2.146208, 1.408664, \ldots, 1.738858)$	0.784598	13.614	1120000
SQP	$(-2.140351, 1.403742, \ldots, 1.740741)$	0.784606	7.824	–

Table 2 The numerical results of Example 3.2, $m = 20000$, $x^0 = (-2, 2, -2, 0, 0, 0.6)^{\mathrm{T}}$

References

[1] AL-SUBAIHI I, WATSON G A. *Fitting parametric curves and surfaces by l_∞ distance regression* [J]. BIT, 2005, **45**(3): 443–461.

[2] BANDLER J W, CHARALAMBOUS C. *Practical least pth optimization of networks* [J]. IEEE Trans. Microwave Theory Tech., 1972, **20**: 834–840.

[3] CHARALAMBOUS C. *Acceleration of the least p-th algorithm for minimax optimization with engineering applications* [J]. Math. Programming, 1979, **17**: 270–297.

[4] CLARKE F H. *Optimization and Nonsmooth Analysis* [M]. John Wiley & Sons, Inc., New York, 1983.

[5] FRANGIONI A. *Generalized bundle methods* [J]. SIAM J. Optim., 2002, **13**(1): 117–156.

[6] GAUDIOSO M, MONACO M F. *A bundle type approach to the unconstrained minimization of convex nonsmooth functions* [J]. Math. Programming, 1982, **23**(2): 216–226.

[7] GÍGOLA C, GÓMEZ S. *A regularization method for solving the finite convex min-max problem* [J]. SIAM J. Numer. Anal., 1990, **27**(6): 1621–1634.

[8] HARTLEY R, SCHAFFALITZKY F. *L_∞ minimization in geometric reconstruction problems* [P]. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recoqnition, 2004, **1**: 504–509.

[9] KIWIEL K C. *Methods of Descent for Nondifferentiable Optimization* [M]. Springer-Verlag, Berlin, 1985.

[10] LI Xingsi. *An aggregate function method for nonlinear programming* [J]. Sci. China Ser. A, 1991, **34**(12): 1467–1473.

[11] MAYNE D Q, POLAK E. *Nondifferential optimization via adaptive smoothing* [J]. J. Optim. Theory Appl., 1984, **43**(4): 601–613.

[12] MÁKELÄ M M, NEITTAANMÄKI P. *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control* [M]. World Scientific, 1992.

[13] MURRAY W, OVERTON M L. *A projected Lagrangian algorithm for nonlinear minimax optimization* [J]. SIAM J. Sci. Statist. Comput., 1980, **1**(3): 345–370.

[14] POLAK E, ROYSET J O, WOMERSLEY R S. *Algorithms with adaptive smoothing for finite minimax problems* [J]. J. Optim. Theory Appl., 2003, **119**(3): 459–484.

[15] SHOR N Z. *Minimization Methods for Non-differentiable Functions* [M]. New York, Berlin, 1985.

[16] SPÄTH H. *Least squares fitting with rotated paraboloids* [J]. Math. Commun., 2001, **6**(2): 173–179.

[17] WOMERSLEY R S, FLETCHER R. *An algorithm for composite nonsmooth optimization problems* [J]. J. Optim. Theory Appl., 1986, **48**(3): 493–523.

[18] XIAO Yu, YU Bo. *A truncated aggregate smoothing Newton method for minimax problems* [J]. Appl. Math. Comput., 2009, doi:10.1016/j.amc.2009.11.034.

[19] XU Song. *Smoothing method for minimax problems* [J]. Comput. Optim. Appl., 2001, **20**(3): 267–279.

[20] YU Bo, FENG Guochen, ZHANG Shaoliang. *The aggregate constraint homotopy method for nonconvex nonlinear programming* [J]. Nonlinear Anal., Ser. A, 2001, **45**(7): 839–847.

[21] ZHOU J L, TITS A L. *An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions* [J]. SIAM J. Optim., 1996, **6**(2): 461–487.

[22] ZOWE J. *Nondifferentiable Optimization: A Motivation and a Short Introduction into the Subgradient and the Bundle Concept* [M]. Springer, Berlin, 1985.