Optimization Methods & Software, 2014
Vol. 29, No. 1, 160–176, http://dx.doi.org/10.1080/10556788.2012.762365

Truncated aggregate homotopy method for nonconvex nonlinear programming

Y. Xiao[a,c], H.J. Xiong[b,c] and B. Yu[c]*

[a]*School of Basic Science, East China Jiaotong University, Nanchang, People's Republic of China;*
[b]*College of Science, Huazhong Agricultural University, Wuhan, People's Republic of China;*
[c]*School of Mathematics, Dalian University of Technology, Dalian, People's Republic of China*

(*Received 18 July 2011; final version received 20 December 2012*)

In this paper, an efficient implementation of aggregate homotopy method for nonconvex nonlinear programming problems is proposed. Adopting truncated aggregate technique, only a small subset of the constraints is aggregated at each iteration, hence the number of gradient and Hessian calculations is reduced dramatically. The subset is adaptively updated with some cheaply implementable truncating criterions, to guarantee the locally quadratic convergence of the correction process with as few computation cost as possible. Numerical tests with comparison to some other methods show that the new method is very efficient, especially for the problems with large amount of constraints and computationally expensive gradients or Hessians.

Keywords: nonlinear programming; nonconvex programming; truncated aggregate function; aggregate homotopy method

AMS Subject Classification: 90C34; 90C46; 49J52; 90C51; 90C90

1. Introduction

Consider the following nonlinear programming problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega = \{x \in R^n \mid g_j(x) \leq 0, j \in \mathbf{q}\}, \end{aligned} \tag{1}$$

where $f, g_j \in C^r (r > 2)$ and $\mathbf{q} = \{1, \ldots, q\}$.

It is well-known that if $x^* \in \Omega$ is a local solution of (1), and linear independence constraint qualification or Mangasarian–Fromovitz constraint qualification (MFCQ) holds, then there exists $\xi^* = (\xi_1^*, \ldots, \xi_q^*) \in R^q$, such that (x^*, ξ^*) is a solution of the Karush–Kuhn–Tucker (KKT) point system

$$\begin{gathered} \nabla f(x) + \sum_{j \in \mathbf{q}} \xi_j \nabla g_j(x) = 0, \\ \xi_j g_j(x) = 0, \qquad \xi_j \geq 0, \quad g_j(x) \leq 0. \end{gathered} \tag{2}$$

*Corresponding author. Email: yubo@dlut.edu.cn

If $(\hat{x}, \hat{\xi})$ is a solution of (2), then $\hat{x}$ is called a KKT point of (1) and $\hat{\xi}$ is called the Lagrangian multiplier vector corresponding to $\hat{x}$.

In [4,5,15], a combined homotopy interior point (CHIP) method was presented for (1). Under a normal cone condition of the constraints, it was proved that a smooth interior path from an interior point of the feasible set to a KKT point of (1) exists. However, the dimension of the linear systems arising in the process of numerically tracing the CHIP path is $n + q + 1$. In [25], by using the aggregate function (also known as exponential penalty function, see [8]), which is induced from the Jayne's maximum entropy principle by Li in [10] and has been used to solve linear programming, nonlinear programming, nonsmooth programming and generalized complementarity problems, etc. (see [11–14,17,18,20,21]), an aggregate constraint homotopy (ACH) method was given for solving (1). Let $g(x) = \max\{g_1(x), \ldots, g_q(x)\}$ and $\Omega^0 = \{x \in R^n : g(x) < 0\}$. The ACH is as follows:

$$H(x, y, t) \equiv \begin{pmatrix} (1-t)(\nabla_x f(x) + y\nabla_x g(x, \theta t)) + t(x - x^0) \\ yg(x, \theta t) - ty^0 g(x^0, \theta) \end{pmatrix} = 0, \tag{3}$$

where $x^0 \in \Omega^0$, $y^0 > 0$ and $\theta \in (0, 1]$ are chosen sufficiently small, and

$$g(x, t) = t \ln \left(\sum_{j \in \mathbf{q}} \exp \left(\frac{g_j(x)}{t} \right) \right), \tag{4}$$

called the aggregate function of $g(x)$, is a smooth uniform and monotonic approximation of $g(x)$ with $g(x) \leq g(x, t) \leq g(x) + t \ln q$. In [25], under a weak normal cone condition with some standard constraint qualifications, it was proved that the ACH determines a smooth interior path from the given interior point to a KKT point of (1). Because the dimension of the linear systems arising in the process of numerically tracing the ACH path is $n + 2$ and not $n + q + 1$ as in CHIP methods, it is reasonable that the ACH method is more efficient for solving nonlinear programming problems with large amount of constraints.

Notice that although $g(x, t)$ is a single smooth function, it is composed of large number of functions $g_j(x)$. Hence, calculation of its gradient and Hessian

$$\begin{aligned}
\nabla_x g(x, t) &= \sum_{j \in \mathbf{q}} \lambda_j(x, t) \nabla g_j(x), \\
g'_t(x, t) &= \frac{1}{t} \left(g(x, t) - \sum_{j \in \mathbf{q}} \lambda_j g_j(x) \right), \\
\nabla_x^2 g(x, t) &= \sum_{j \in \mathbf{q}} \lambda_j(x, t) \nabla^2 g_j(x) + \frac{1}{t} \sum_{j \in \mathbf{q}} \lambda_j(x, t) \nabla g_j(x) (\nabla g_j(x))^{\mathrm{T}} \\
&\quad - \frac{1}{t} \left(\sum_{j \in \mathbf{q}} \lambda_j(x, t) \nabla g_j(x) \right) \left(\sum_{j \in \mathbf{q}} \lambda_j(x, t) (\nabla g_j(x)) \right)^{\mathrm{T}}, \\
\nabla_{xt}^2 g(x, t) &= \frac{1}{t^2} \left(\sum_{j \in \mathbf{q}} \lambda_j(x, t) \nabla g_j(x) \sum_{j \in \mathbf{q}} \lambda_j(x, t) g_j(x) - \sum_{j \in \mathbf{q}} \lambda_j(x, t) g_j(x) \nabla g_j(x) \right),
\end{aligned} \tag{5}$$

where

$$\lambda_j(x, t) = \frac{\exp(g_j(x)/t)}{\sum_{j \in \mathbf{q}} \exp(g_j(x)/t)} \in (0, 1), \quad \sum_{j \in \mathbf{q}} \lambda_j(x, t) = 1, \tag{6}$$

is very expensive when q is very large, and we need more effort to gain efficient performance of the ACH method.

In [24], we proposed a truncated aggregate smoothing stabilized Newton method for unconstraint minimax problem. To reduce the computation cost, in each iteration point $\bar{x} = x^{k,i}$, choose parameter $\epsilon > 0$, denotes

$$\mathbf{q}(\bar{x}, \epsilon) = \{j \mid g(\bar{x}) - g_j(\bar{x}) \leq \epsilon, j \in \mathbf{q}\}. \tag{7}$$

The truncated aggregate function with respect to $\mathbf{q}(\bar{x}, \epsilon)$ was defined as

$$g_\epsilon(x, t) = t \ln \left(\sum_{j \in \mathbf{q}(\bar{x}, \epsilon)} \exp\left(\frac{g_j(x)}{t}\right) \right), \tag{8}$$

where the parameter ϵ was adaptively updated in each iteration point with some criterions to guarantee the convergence. For conciseness, we write $\mathbf{q}_\epsilon = \mathbf{q}(\bar{x}, \epsilon)$.

In this paper, using the truncated aggregate function, we propose a truncated aggregate homotopy (TAH) algorithm to implement the ACH method for (1). At each iteration, only a small subset of components in max-function is aggregated, hence the number of gradient and Hessian calculations is dramatically reduced, and hence the computation cost is greatly reduced especially for the problems with large q and computationally expensive gradients or Hessians. Moreover, we give some new truncating criterions, concerning only with computation of function values and not their gradients or Hessians, for adaptively updating the subset to guarantee the locally quadratic convergence of the correction process with as few computational cost as possible.

The paper is organized as follows. In Section 2, the ACH method and generic path following algorithm are restated, and some necessary propositions are given. In Section 3, the TAH algorithm and the convergence are presented. Some test results are given in Section 4. The paper ends with concluding remarks.

In the following, R^n, R^n_+ and R^n_{++} denote the n-dimensional Euclidean space, the nonnegative orthant of R^n and the positive orthant of R^n, respectively. Let $G(x) : R^n \to R^q$ be a smooth map, denote $DG(x) = (\nabla G_1(x), \ldots, \nabla G_q(x))^{\mathrm{T}}$, where $\nabla G_j(x) \in R^n$ is the gradient of $G_j(x), j = 1, \ldots, q$. A point $x \in R^n$ is called a regular point of G if the Jacobian $DG(x)$ has maximal rank $\min\{n, q\}$. A value $z \in R^q$ is called a regular value of G if x is a regular point of G for all $x \in G^{-1}(z)$. $V(z, \delta)$ means the set $\{x \mid \|x - z\| < \delta\}$ and $\bar{V}(z, \delta)$ means the set $\{x \mid \|x - z\| \leq \delta\}$. $f(h) = O(h^m)$ means that $\|f(h)\| \leq C|h|^m$ for small h and a constant $C > 0$. $\Omega_\theta(t) = \{x \in R^n \mid g(x, \theta t) \leq 0\}$. Finally, the symbol $\|\cdot\|$ denotes the Euclidean norm.

2. Aggregate homotopy method

We first restate the ACH method in [25]. Given the following assumptions [25]:

- (C_1) $f(x)$ and $g_j(x)$ are three times continuously differentiable functions.
- (C_2) Ω^0 is nonempty and bounded.
- (C_3) For any $x \in \partial\Omega$, the matrix $\{\nabla g_j(x) : j \in J(x) = \{j : g_j(x) = g(x)\}\}$ has a full column rank (*regularity of* $\partial\Omega$).
- (C_4) There exists a closed subset $\hat{\Omega} \subset \Omega^0$ with nonempty interior $\hat{\Omega}^0$, such that Ω satisfies the weak normal cone condition w.r.t. $\hat{\Omega}$.

DEFINITION 2.1 (The weak normal cone condition of Ω w.r.t. $\hat{\Omega}$, see [25]) *Let $\hat{\Omega}$ be a closed subset of Ω^0, if the normal cone of Ω at any $x \in \partial\Omega$ does not meet $\hat{\Omega}$, i.e.*

$$\left\{x + \sum_{j\in J(x)} \lambda_j \nabla g_j(x) : \lambda_j \geq 0, \sum_{j\in J(x)} \lambda_j > 0\right\} \cap \hat{\Omega} = \emptyset,$$

then Ω is said to satisfy the weak normal cone condition w.r.t. $\hat{\Omega}$.

THEOREM 2.2 ([25], Theorem 2.9) *Suppose that assumptions* (C_1)–(C_4) *hold, then for any $\tilde{x} \in \hat{\Omega}^0$, there exists a neighbourhood $N(\tilde{x})$ of $\tilde{x}$ such that $N(\tilde{x}) \subset \hat{\Omega}^0$, and there exists a $\theta \in (0, 1]$ such that $N(\tilde{x}) \subset \Omega(1)^0$, $\partial\Omega(t)$ is regular and $\Omega(t)$ satisfies the normal cone condition w.r.t. $N(\tilde{x})$ for any $t \in (0, 1]$. Furthermore, for almost all $x^{(0)} \in N(\tilde{x})$ and $y^{(0)} > 0$,*

$$H^{-1}(0) = \{(x, y, t) \in \Omega \times R^1_+ \times (0, 1] : H(x, y, t) = 0\}$$

contains a smooth curve Γ which starts from $(x^{(0)}, y^{(0)}, 1)$ and terminates in approaches to the hyperplane at $t = 0$. Moreover, let $(x^*, y^*, 0)$ be any limit point of Γ on the hyperplane at $t = 0$, and $\xi_i^* = y^*\lambda_i(x^*, 0)$ (where $\lambda_i(x, t)$ is defined as (6)*), $i = 1, \dots, q$, then y^* is finite and $(x^*, \xi_1^*, \dots, \xi_q^*)$ is a solution of* (2)*, x^* is a KKT point of* (1) *and $\xi_1^*, \dots, \xi_q^*$ are corresponding Lagrangian multipliers.*

Remark 1 Assumptions (C_1)–(C_4) are sufficient conditions for aggregate homotopy method to solve nonconvex smooth programming (see [25]). Actually, (C_1) and (C_2) are common assumptions to make the problem solvable. Assumption (C_3) is given for bounded Lagrange multiplier. Moreover, after publication of the paper [25], authors found that the assumption (C_3) can be weakened as: (C_3^*) For any $x \in \partial\Omega$, $\{\nabla g_j(x) : j \in J(x) = \{j : g_j(x) = g(x)\}\}$ are positively independent, i.e. $\sum_{j\in J(x)} \xi_j \nabla g_j(x) = 0, \xi_j \geq 0 \Rightarrow \xi_j = 0$, which is an equivalent formulation of MFCQ. In the later related works, the assumption was changed to (C_3^*) (see [26]). In the rest of this paper, we will use the weaker assumption (C_3^*) too. (C_4) is specialized to guarantee the existence of smooth interior point homotopy path for nonconvex programming. If the problem is convex, (C_4) holds naturally.

By numerically tracing the smooth path Γ, a globally convergent algorithm can be established. The predictor–corrector (PC) method (see [2]) is usually adopted to numerically trace the homotopy path Γ by generating a sequence of points (x^k, y^k, t_k), $k = 1, 2, \dots$, along the curve satisfying a chosen tolerance criterion, say $\|H(x^k, y^k, t_k)\| \leq H_{\text{tol}}$ for some $H_{\text{tol}} > 0$. Here, the Euler_Newton method for tracing ACH is outlined briefly as follows. For conciseness, we write $v = (x, y, t)$.

Algorithm 1 (Euler_Newton method for tracing aggregate constraint homotopy)
Data. $\theta \in (0, 1]$, $x^0 \in \hat{\Omega}^0$, $y^0 > 0$, $t_0 = 1$, initial predictor direction d^0.
Parameters. Initial steplength $h > 0$.

$k = 0$.

repeat

Predictor step

Compute unit predictor vector d^k satisfying the following system

$$DH(v^k)d = 0,$$

$$\|d\| = 1, \tag{9}$$
$$\mathrm{sign}\left(\det\begin{bmatrix} DH(v^k) \\ d^{\mathrm{T}} \end{bmatrix}\right) = \mathrm{sign}\left(\det\begin{bmatrix} DH(v^0) \\ d^{0\mathrm{T}} \end{bmatrix}\right).$$

Set $v^{k,0} = v^k + hd^k$;
Corrector step
Let $v^{k,*}$ solve the following corrector equation

$$R_k(v) = \begin{bmatrix} H(v) \\ {d^k}^{\mathrm{T}}(v - v^{k,0}) \end{bmatrix} = 0, \tag{10}$$

with Newton iteration $v^{k,i+1} = v^{k,i} - DR_k(v^{k,i})^{-1}R_k(v^{k,i})$;
Set $v^{k+1} = v^{k,*}, k = k + 1$;
Choose a new steplength $h > 0$;
until traversing is stopped.

Remark 2 The initial predictor direction d^0 is

$$d^0 = \frac{-d}{\|d\|\mathrm{sign}(d_{n+2})}, \tag{11}$$

where d is the tangent vector satisfying $DH(v^0)d = 0$. In predictor step, predictor direction is usually taken as the unit tangent vector at the preceding point v^k (see [9,16]). Another popular choice is unit scant vector $d^k = (v^k - v^{k-1})/\|v^k - v^{k-1}\|$ which is able to avoid solving system (9) (see [6]). In corrector step, the predictor point is usually drawn back to $H^{-1}(0)$ satisfying additional equation ${d^k}^{\mathrm{T}}(v - v^{k,0}) = 0$ or $\|v - v^k\|^2 - h^2 = 0$ (see [7,16]). In some other papers, the corrector point is obtained by approximately solving $\min_v\{\|v - v^{k,0}\| | H(v) = 0\}$ (see [2]). Moreover, an efficient algorithm needs to incorporate an automatic strategy for controlling the steplength h. In [2], some strategies are outlined in detail. Here, we adopt a simple and heuristic strategy introduced in [6]: if, for various reasons, a predictor step with its subsequent corrector step is not accepted, then the steplength is reduced by a factor $\bar{\alpha} < 1$; if the corrector step with Newton iterations succeeds fast, the steplength is increased by the factor $\bar{\alpha} > 1$. For extensive literature on these methods, see [2,6,9,16]. Under the assumptions that H is a smooth map having zero as a regular value and the step size $h_{\max} > 0$ is sufficiently small, the convergence discussions are fairly classical and may be found in [2,7,16,19].

Since zero is a regular value of H and $v^k \in H^{-1}(0)$, then $DH(v^k)$ has full row rank, and hence $DR_k(v^k) = \begin{bmatrix} DH(v^k) \\ {d^k}^{\mathrm{T}} \end{bmatrix}$ is nonsingular if stepsize h is small which can be controlled by choosing small $h_{\max}$. Now, some necessary preliminaries are stated firstly. Substitute $v^{k,0}$ with $v^k + hd^k$ in correction equation (10) and use $\|d^k\| = 1$, we can rewrite the correction equation as follows:

$$R_k(v) = \begin{bmatrix} H(v) \\ {d^k}^{\mathrm{T}}(v - v^k) - h \end{bmatrix} = 0. \tag{12}$$

Proposition 2.3 *Assume $v^k \in H^{-1}(0)$ and $DR_k(v^k)$ is nonsingular. Then there exists $h_1 > 0$, $0 < \delta_1 < 1$, such that $DR_k(v)$ is nonsingular in the neighborhood $V_k = V(v^k, \delta_1)$ and the corrector equation* (12) *has unique solution $v^{k+1}(h) \in \bar{V}_k$ for any $h \in [0, h_1]$. Moreover, $v^{k+1} : (0, h_1) \to R^{n+2}$ is continuous.*

Proof Take h as a variable in (12) and denote (12) as $R_k(v,h)=0$. Since $R_k(v^k,0)=0$ and the partial derivative $D_vR_k(v^k,0)=\begin{bmatrix} DH(v^k) \\ d^{k\mathrm{T}} \end{bmatrix}$ is nonsingular, then the assertion follows from the implicit function theorem. ■

It follows directly from Proposition 2.3 that

PROPOSITION 2.4 *For any* $0<\delta_2<\delta_1/2$, *there exists* $0<h_2<h_1$, *such that for any* $h\in(0,h_2]$, *it has* $V(v^{k+1}(h),\delta_2)\subset V_k$.

PROPOSITION 2.5 *Let* $U_k=\bigcup_{h\in(0,h_2]}V(v^{k+1}(h),\delta_2)$. *Denote*

$$\bar{\alpha}=\max\{\|DR_k(v)^{-1}\|\mid v\in\bar{U}_k\},$$
$$\bar{\gamma}=\max\{\|D^2R_k(v)\|\mid v\in\bar{U}_k\},$$
$$\bar{\delta}=\min\left\{\frac{\delta_2}{2},\frac{1}{5\bar{\alpha}\bar{\gamma}}\right\}.$$

Then, there exists an $\bar{h}_k<h_2$ *such that for any* $0<h<\bar{h}_k$, *it has* $\|v^{k+1}-v^{k,0}\|<\bar{\delta}$.

Proof It's easy to see that

$$\|v^{k+1}-v^{k,0}\|=\sqrt{\|v^{k+1}(h)-v^k\|^2-h^2}.$$

Then from the continuity of $v^{k+1}(h)$ and $v^{k+1}(0)=v^k$, we know the proposition holds. ■

3. Truncated aggregate homotopy algorithm

As shown in (5), the gradient and Hessian calculations of aggregate function $g(x,t)$ are time-consuming, hence, we consider to use the truncated aggregate technique to trace the homotopy path Γ efficiently. Set $V_\Gamma=\bigcup_{v\in\Gamma}\bar{V}(v,1)$ such that $\bigcup_{v^k\in\Gamma}V_k$ is contained in V_Γ. Denote $\Omega_\Gamma=\{x\mid(x,y,t)\in V_\Gamma\}$. Since the homotopy curve Γ is bounded (see Theorem 2.2), it is obvious that V_Γ and Ω_Γ are bounded. Denote $A_0(x)=\max_{j\in\mathbf{q}}|g_j(x)|$, $A_1(x)=\max_{j\in\mathbf{q}}\|\nabla g_j(x)\|$ and $A_2(x)=\max_{j\in\mathbf{q}}\|\nabla^2g_j(x)\|$, then $A_i(x)$, $i=0,1,2$, are bounded in Ω_Γ.

Algorithm 2 (Truncated aggregate homotopy algorithm)
Data. $\theta>0, x^0\in\hat{\Omega}^0, y^0=1, t_0=1$.
Parameters. Initial steplength $h_0>0$; tolerance $t_{\mathrm{tol}}=10^{-7}$, $t_c=10^{-6}$, $H_{\mathrm{tol}}=10^{-3}$; maximum steplength $h_{\max}$; maximum inner iteration number N_{in}; A_1,A_2 are big numbers such that $A_i\geq\max\{A_i(x)\mid x\in\Omega_\Gamma\}, i=1,2$; error parameters $\{\eta_{k,i}\}_{k,i=0}^{\infty}$, $\{\mu_{k,i}\}_{k,i=0}^{\infty}$.

Step 0. Unit tangent vector d^0, $k=0, i=0$.

(Predictor step)

Step 1. If $0\leq t_k\leq t_{\mathrm{tol}}$, end the procedure; else go to Step 2.
Step 2. If $k>0$, compute $d^k=(v^k-v^{k-1})/\|v^k-v^{k-1}\|$. If $t_k>t_c$, go to Step 3, else go to Step 4.
Step 3. Set $v^{k,0}=v^k+h_kd^k, i=0$, go to Step 5.
Step 4. Let $h_k=(t_{\mathrm{tol}}-t_k)/d^k_{n+2}$ and $v^{k,0}=v^k+h_kd^k$, then correct $v^{k,0}$ on the hyperplane $t=t_{\mathrm{tol}}$.

(Corrector step)

Step 5. If $v^{k,i} \notin \Omega \times R^1_+ \times [0,1]$, set $h_k = 0.5h_k$, go to Step 2; else, let $v = (x, y, t) = (x^{k,i}, y^{k,i}, t_{k,i})$, $\eta = \eta_{k,i}$, $\mu = \mu_{k,i}$, calculate

$$\epsilon = \epsilon^{k,i} = \max\{\bar{\epsilon}, \bar{\bar{\epsilon}}\}, \tag{13}$$

where

$$\begin{aligned} \bar{\epsilon} &= \theta t \ln\left(\max\left\{\frac{(q-1)}{\eta} y(2(1-t)A_1 + \theta t), 1\right\}\right), \\ \bar{\bar{\epsilon}} &= \theta t \ln\left(\max\left\{\frac{q-1}{\mu}\left((1-t)y\left(2A_2 + \frac{6A_0(x)A_1}{\theta t^2} + \frac{6A_1^2}{\theta t}\right)\right.\right.\right. \\ &\quad \left.\left.\left. + \; 2(2y+1-t)A_1 + \theta(t+y) + \frac{2yA_0(x)}{t}\right), 1\right\}\right), \end{aligned} \tag{14}$$

update $\mathbf{q}_\epsilon$ according to (7) with $\bar{x} = x^{k,i}$, then go to Step 6.

Step 6. Compute $\|R_{k,\epsilon}(v)\|$, where

$$R_{k,\epsilon}(v) = \begin{bmatrix} H_\epsilon(v) \\ {d^k}^{\mathrm{T}}(v - v^{k,0}) \end{bmatrix},$$

with

$$H_\epsilon(v) = \begin{bmatrix} (1-t)(\nabla_x f(x) + y\nabla_x g_\epsilon(x, \theta t)) + t(x - x^0) \\ y g_\epsilon(x, \theta t) - t y^0 g(x^0, \theta) \end{bmatrix}.$$

If $\|R_{k,\epsilon}(v)\| \le H_{\mathrm{tol}}$, go to Step 7; else go to Step 8.

Step 7. Set $v^{k+1} = v^{k,i}$, $h_{k+1} = \begin{cases} \min\{1.5h_k, h_{\max}\}, & i<3, \\ h_k, & \text{else}, \end{cases}$ $k = k+1$, go to Step 1.

Step 8. If $i \ge N_{\mathrm{in}}$, set $h_k = 0.5h_k$, go to Step 3; else, obtain $v^{k,i+1}$ using truncated aggregate iteration,

$$v^{k,i+1} = v^{k,i} - DR_{k,\epsilon}(v^{k,i})^{-1} R_{k,\epsilon}(v^{k,i}), \tag{15}$$

set $i = i + 1$ and go to Step 5.

At first, we give some estimates of difference between the aggregate function (4) and the truncated aggregate function (8), which are induced from Proposition 2.2 in [24] and important for the efficient implementation of the aggregate homotopy method.

COROLLARY 3.1 *Suppose that assumption* (C_1) *holds. For any given* $\bar{x} \in R^n, 0 < t \le 1, \epsilon > 0$, *let* $g(x, \theta t), \boldsymbol{q}_\epsilon$ *and* $g_\epsilon(x, \theta t)$ *be defined as in* (4), (7) *and* (8). *Then the following error estimates hold*

(i) $0 \le g(\bar{x}, \theta t) - g_\epsilon(\bar{x}, \theta t) \le \theta t(q-1)\exp(-\epsilon/\theta t)$;
(ii) $\|\nabla_x g(\bar{x}, \theta t) - \nabla_x g_\epsilon(\bar{x}, \theta t)\| \le 2(q-1)\exp(-\epsilon/\theta t)A_1(\bar{x})$;
(iii) $\|g'_t(\bar{x}, \theta t) - (g_\epsilon)'_t(\bar{x}, \theta t)\| \le (q-1)\exp(-\epsilon/\theta t)(\theta + (2/t)A_0(\bar{x}))$;
(iv) $\|\nabla^2_x g(\bar{x}, \theta t) - \nabla^2_x g_\epsilon(\bar{x}, \theta t)\| \le 2(q-1)\exp(-\epsilon/\theta t)(A_2(\bar{x}) + (3/\theta t)A_1^2(\bar{x}))$;
(v) $\|\nabla^2_{xt} g(\bar{x}, \theta t) - \nabla^2_{xt} g_\epsilon(\bar{x}, \theta t)\| \le (6/\theta t^2)(q-1)\exp(-\epsilon/\theta t)A_0(\bar{x})A_1(\bar{x})$.

Proof Here we give the proof for items (iii) and (v), and others can be found in [24].

(iii)

$$\|g_t'(x,\theta t)-(g_\epsilon)_t'(x,\theta t)\| = \frac{1}{t}\left\| g(x,\theta t) - \sum_{j\in\mathbf{q}} \lambda_j(x,\theta t) g_j(x) - g_\epsilon(x,\theta t) \right.$$

$$\left. + \sum_{j\in\mathbf{q}_\epsilon} \zeta_j(x,\theta t) g_j(x) \right\|, \tag{16}$$

where

$$\zeta_j(x,\theta t) = \frac{\exp(g_j(x)/\theta t)}{\sum_{j\in\mathbf{q}_\varepsilon}\exp(g_j(x)/\theta t)} \in (0,1), \quad \sum_{j\in\mathbf{q}_\epsilon}\zeta_j(x,\theta t)=1. \tag{17}$$

For conciseness, we write $\zeta_j(x,\theta t)$, $\lambda_j(x,\theta t)$, $\zeta_j(\bar{x},\theta t)$ and $\lambda_j(\bar{x},\theta t)$ as ζ_j, λ_j, $\bar{\zeta}_j$ and $\bar{\lambda}_j$, respectively. From (i) and the following inequality which has been proved in [24],

$$\sum_{j\in\mathbf{q}_\epsilon} |\bar{\zeta}_j - \bar{\lambda}_j| = \sum_{j\notin\mathbf{q}_\epsilon} \bar{\lambda}_j \le (q-1)\exp\left(-\frac{\epsilon}{\theta t}\right), \tag{18}$$

it has

$$\begin{aligned}
\|g_t'(\bar{x},\theta t)-(g_\epsilon)_t'(\bar{x},\theta t)\| &\le \frac{1}{t}\left(\|g(\bar{x},\theta t)-g_\epsilon(\bar{x},\theta t)\| + \left\|\sum_{j\in\mathbf{q}}\bar{\lambda}_j g_j(\bar{x}) - \sum_{j\in\mathbf{q}_\epsilon}\bar{\zeta}_j g_j(\bar{x})\right\|\right)\\
&\le \frac{1}{t}\left(\|g(\bar{x},\theta t)-g_\epsilon(\bar{x},\theta t)\| + \left\|\sum_{j\in\mathbf{q}_\epsilon}(\bar{\lambda}_j-\bar{\zeta}_j) g_j(\bar{x})\right\| + \left\|\sum_{j\notin\mathbf{q}_\epsilon}\bar{\lambda}_j g_j(\bar{x})\right\|\right)\\
&\le \frac{1}{t}\left(\theta t(q-1)\exp\left(-\frac{\epsilon}{\theta t}\right) + 2(q-1)\exp\left(-\frac{\epsilon}{\theta t}\right)A_0(\bar{x})\right)\\
&= (q-1)\exp\left(-\frac{\epsilon}{\theta t}\right)\left(\theta + \frac{2}{t}A_0(\bar{x})\right).
\end{aligned} \tag{19}$$

(v) Since

$$\begin{aligned}
&\nabla^2_{xt} g(x,\theta t) - \nabla^2_{xt} g_\epsilon(x,\theta t)\\
&\quad = \frac{1}{\theta t^2}\left(\left(\sum_{j\in\mathbf{q}}\lambda_j g_j(x)\right)\left(\sum_{j\in\mathbf{q}}\lambda_j \nabla g_j(x)\right) - \sum_{j\in\mathbf{q}}\lambda_j g_j(x)\nabla g_j(x)\right)\\
&\qquad - \frac{1}{\theta t^2}\left(\left(\sum_{j\in\mathbf{q}_\epsilon}\zeta_j g_j(x)\right)\left(\sum_{j\in\mathbf{q}_\epsilon}\zeta_j \nabla g_j(x)\right) - \sum_{j\in\mathbf{q}_\epsilon}\zeta_j g_j(x)\nabla g_j(x)\right)\\
&\quad = \frac{1}{\theta t^2}\left(\left(\sum_{j\in\mathbf{q}_\epsilon}\lambda_j g_j(x)\right)\left(\sum_{j\in\mathbf{q}_\epsilon}\lambda_j \nabla g_j\right) - \left(\sum_{j\in\mathbf{q}_\epsilon}\zeta_j g_j(x)\right)\left(\sum_{j\in\mathbf{q}_\epsilon}\lambda_j \nabla g_j(x)\right)\right.\\
&\qquad + \left(\sum_{j\in\mathbf{q}_\epsilon}\zeta_j g_j(x)\right)\left(\sum_{j\in\mathbf{q}_\epsilon}\lambda_j \nabla g_j(x)\right) - \left(\sum_{j\in\mathbf{q}_\epsilon}\zeta_j g_j(x)\right)\left(\sum_{j\in\mathbf{q}_\epsilon}\zeta_j \nabla g_j(x)\right)
\end{aligned}$$

$$+\left(\sum_{j\in\mathbf{q}_\epsilon}\lambda_j g_j(x)\right)\left(\sum_{j\notin\mathbf{q}_\epsilon}\lambda_j\nabla g_j(x)\right)+\left(\sum_{j\notin\mathbf{q}_\epsilon}\lambda_j g_j(x)\right)\left(\sum_{j\in\mathbf{q}}\lambda_j\nabla g_j(x)\right)$$

$$+\left(\sum_{j\in\mathbf{q}_\epsilon}\zeta_j g_j(x)\nabla g_j(x)-\sum_{j\in\mathbf{q}_\epsilon}\lambda_j g_j(x)\nabla g_j(x)\right)-\sum_{j\notin\mathbf{q}_\epsilon}\lambda_j g_j(x)\nabla g_j(x)\Bigg), \tag{20}$$

then together with (18), $\sum_{j\in\mathbf{q}_\epsilon}\bar{\zeta}_j=1$ and $\sum_{j\in\mathbf{q}}\bar{\lambda}_j=1$, we have

$$\|\nabla^2_{xt}g(\bar{x},\theta t)-\nabla^2_{xt}g_\epsilon(\bar{x},\theta t)\|\le\frac{6}{\theta t^2}(q-1)\exp\left(-\frac{\epsilon}{\theta t}\right)A_0(\bar{x})A_1(\bar{x}).$$

■

From Corollary 3.1, the following proposition holds.

PROPOSITION 3.2 *Suppose that assumption* (C_1) *holds. In Algorithm* 2, *for any* $v\in V_\Gamma$ *with* $y\ge 0$, *and any error parameters* $\eta,\mu>0$, *if* ϵ *is set as* (13) *and* (14), *it has*

$$\|H(v)-H_\epsilon(v)\|\le\eta, \tag{21}$$

$$\|DH(v)-DH_\epsilon(v)\|\le\mu. \tag{22}$$

Proof For concise, denote

$$\begin{aligned}&H_1(x,y,t)=(1-t)(\nabla_x f(x)+y\nabla_x g(x,\theta t))+t(x-x^0),\\&H_2(x,y,t)=yg(x,\theta t)-ty^0g(x^0,\theta).\end{aligned} \tag{23}$$

Since

$$\epsilon\ge\bar{\epsilon}=\theta t\ln\left(\max\left\{\frac{(q-1)}{\eta}y(2(1-t)A_1+\theta t),1\right\}\right), \tag{24}$$

the following inequality holds

$$(q-1)y(2(1-t)A_1+\theta t)\exp\left(-\frac{\epsilon}{\theta t}\right)\le\eta. \tag{25}$$

According to Corollary 3.1, we have

$$\begin{aligned}\|H_1(v)-H_{1,\epsilon}(v)\|&=(1-t)\|y(\nabla_x g(x,\theta t)-\nabla_x g_\epsilon(x,\theta t))\|\\&\le 2(1-t)(q-1)yA_1(x)\exp\left(-\frac{\epsilon}{\theta t}\right)\end{aligned} \tag{26}$$

and

$$\|H_2(v)-H_{2,\epsilon}(v)\|=\|y(g(x,\theta t)-g_\epsilon(x,\theta t))\|\le y\theta t(q-1)\exp\left(-\frac{\epsilon}{\theta t}\right). \tag{27}$$

Now, since $A_i\ge A_i(x)$, $i=1,2$, it immediately follows from (25)–(27) that

$$\|H(v)-H_\epsilon(v)\|\le\|H_1(v)-H_{1,\epsilon}(v)\|+\|H_2(v)-H_{2,\epsilon}(v)\|\le\eta. \tag{28}$$

Similarly, the assertion $\|DH(v)-DH_\epsilon(v)\|\le\mu$ also holds. Since $\epsilon\ge\bar{\bar{\epsilon}}$, it has

$$\begin{aligned}&(q-1)\left((1-t)y\left(2A_2+\frac{6A_0(x)A_1}{\theta t^2}+\frac{6A_1^2}{\theta t}\right)\right.\\&\left.\quad+2(2y+1-t)A_1+\theta(t+y)+\frac{2yA_0(x)}{t}\right)\exp\left(-\frac{\epsilon}{\theta t}\right)\le\mu.\end{aligned} \tag{29}$$

According to Corollary 3.1, we have

$$
\begin{aligned}
\left\| \frac{\partial H_1(v)}{\partial x} - \frac{H_{1,\epsilon}(v)}{\partial x} \right\| &\leq 2(q-1)(1-t)y\left(A_2 + \frac{3}{\theta t}A_1^2\right)\exp\left(-\frac{\epsilon}{\theta t}\right),\\
\left\| \frac{\partial H_1(v)}{\partial y} - \frac{H_{1,\epsilon}(v)}{\partial y} \right\| &\leq 2(q-1)(1-t)A_1\exp\left(-\frac{\epsilon}{\theta t}\right),\\
\left\| \frac{\partial H_1(v)}{\partial t} - \frac{H_{1,\epsilon}(v)}{\partial t} \right\| &\leq (q-1)yA_1\left(2 + (1-t)\frac{6A_0(x)}{\theta t^2}\right)\exp\left(-\frac{\epsilon}{\theta t}\right),\\
\left\| \frac{\partial H_2(v)}{\partial x} - \frac{H_{2,\epsilon}(v)}{\partial x} \right\| &\leq 2(q-1)yA_1\exp\left(-\frac{\epsilon}{\theta t}\right),\\
\left\| \frac{\partial H_2(v)}{\partial y} - \frac{H_{2,\epsilon}(v)}{\partial y} \right\| &\leq (q-1)\theta t\exp\left(-\frac{\epsilon}{\theta t}\right),\\
\left\| \frac{\partial H_2(v)}{\partial t} - \frac{H_{2,\epsilon}(v)}{\partial t} \right\| &\leq (q-1)y\left(\theta + \frac{2}{t}A_0(x)\right)\exp\left(-\frac{\epsilon}{\theta t}\right).
\end{aligned}
\tag{30}
$$

Then from (29) and (30), the assertion $\|DH(v) - DH_\epsilon(v)\| \leq \mu$ holds. ■

Under the assumptions that H is a smooth map having zero as a regular value and the step size $h_{\max} > 0$ is sufficiently small, the global convergence proofs for PC algorithms have been given in [2,7,16,19] and some other papers. Compared with those existing algorithms, the main difference in our algorithm is the adopting of truncated aggregate technique in Newton-corrector steps. Here, we only discuss the convergence of corrector steps. The notations used in the following theorem are the same as in Propositions 2.3–2.5 if no special explanation is given.

THEOREM 3.3 *Suppose that assumptions* (C_1), (C_2), (C_3^*) *and* (C_4) *hold. In Algorithm* 2, *assume* $v^k \in H^{-1}(0)$, $R_k(v) = 0$ *is defined as* (12) *with* $h < \bar{h}_k$ *and* $DR_k(v^k)$ *nonsingular. Then*

(i) $R_k(v) = 0$ *has unique solution* v^{k+1} *in the neighborhood* V_k *with* $DR_k(v^{k+1})$ *nonsingular and* $\|v^{k+1} - v^{k,0}\| < \bar{\delta}$;

(ii) *Denote*

$$\alpha = \max\{\|DR_k(v)^{-1}\| \mid v \in \bar{V}(v^{k+1}, \delta_2)\} \quad \textit{and} \quad \beta = \max\{\|DR_k(v)\| \mid v \in \bar{V}(v^{k+1}, \delta_2)\}.$$

If $\eta_{k,i} \leq \min\{1/30\alpha\beta, 1/2\}H_{\text{tol}}$, $\mu_{k,i} \leq 1/5\alpha$, *then the sequence* $\{v^{k,i}\}$ *generated by the corrector step in Algorithm* 2 *is well defined and finite, i.e. there exists* $i_k \in N$ *such that* $\|R_{k,\epsilon^{k,i_k}}(v^{k,i_k})\| \leq H_{\text{tol}}$ *holds for* $i = i_k$;

(iii) *For sufficiently small* H_{tol}, *if* $\eta_{k,i} \leq \min\{H_{\text{tol}}/30\alpha\beta, O(H_{\text{tol}}^2)\}$ *and* $\mu_{k,i} \leq \min\{1/5\alpha, O(H_{\text{tol}})\}$, *then for* $v^{k,i}$ *sufficiently close to* v^{k+1}, *it has* $\|v^{k,i+1} - v^{k+1}\| = O(\|v^{k,i} - v^{k+1}\|^2)$.

Proof (i) The assertion holds from Propositions 2.3–2.5.

(ii) From Propositions 2.3–2.4, we have that $DR_k(v)$ is nonsingular for $v \in S_0 = V(v^{k+1}, \delta_2)$. Define $\gamma = \max\{\|D^2R_k(v)\| \mid v \in \bar{S}_0\}$. Let $\delta = \min\{\delta_2/2, 1/5\alpha\gamma\}$ and $S = V(v^{k+1}, \delta)$. For all $v \in S$, we have

$$\|R_k(v) - R_k(v^{k+1})\| \leq \beta\|v - v^{k+1}\|,$$

$$\|DR_k(v) - DR_k(v^{k+1})\| \le \gamma \|v - v^{k+1}\| \le \gamma\delta \le \frac{1}{5\alpha},$$

$$\|R_k(v) - R_k(v^{k+1}) - DR_k(v^{k+1})(v - v^{k+1})\| \le \frac{\gamma}{2}\|v - v^{k+1}\|^2 \le \frac{1}{10\alpha}\|v - v^{k+1}\|. \tag{31}$$

Since $S_0 \subset U_k$, hence $\alpha \le \bar{\alpha}$ and $\gamma \le \bar{\gamma}$, and then $\delta \ge \bar{\delta}$. From Proposition 2.5, we obtain $v^{k,0} \in S$. Suppose $v^{k,i} \in S$, then by $S \subset V_k$, we have $A_j(x^{k,i}) \le A_j, j = 1, 2$. According to Proposition 3.2, the following inequalities hold

$$\|R_k(v^{k,i}) - R_{k,\epsilon^{k,i}}(v^{k,i})\| = \|H(v^{k,i}) - H_{\epsilon^{k,i}}(v^{k,i})\| \le \eta_{k,i}, \tag{32}$$

$$\|DR_k(v^{k,i}) - DR_{k,\epsilon^{k,i}}(v^{k,i})\| = \|DH(v^{k,i}) - DH_{\epsilon^{k,i}}(v^{k,i})\| \le \mu_{k,i}. \tag{33}$$

Hence, it follows from (33) that $DR_{k,\epsilon^{k,i}}(v^{k,i})$ is nonsingular and

$$\|(DR_{k,\epsilon^{k,i}}(v^{k,i}))^{-1}\| \le \frac{\alpha}{1 - \alpha\mu_{k,i}}, \tag{34}$$

i.e. $\{v^{k,i}\}$ is well-defined.

Next, we would like to prove there exists a $i_k \in N$ such that $\|R_{k,\epsilon^{k,i_k}}(v^{k,i_k})\| \le H_{\text{tol}}$ holds. For $\|R_{k,\epsilon^{k,i}}(v^{k,i})\| > H_{\text{tol}}$ from (32), it has

$$\|R_k(v^{k,i})\| \ge \|R_{k,\epsilon^{k,i}}(v^{k,i})\| - \|R_k(v^{k,i}) - R_{k,\epsilon^{k,i}}(v^{k,i})\| \ge \frac{H_{\text{tol}}}{2}, \tag{35}$$

and hence

$$\|R_{k,\epsilon^{k,i}}(v^{k,i}) - R_k(v^{k,i})\| \le \frac{H_{\text{tol}}}{30\alpha\beta} \le \frac{\|R_k(v^{k,i})\|}{15\alpha\beta},$$

then, together with (31)–(34), it has

$$\begin{aligned}
\|v^{k,i+1} - v^{k+1}\| &= \|v^{k,i} - (DR_{k,\epsilon^{k,i}}(v^{k,i}))^{-1}R_{k,\epsilon^{k,i}}(v^{k,i}) - v^{k+1}\| \\
&= \|(DR_{k,\epsilon^{k,i}}(v^{k,i}))^{-1}(R_k(v^{k,i}) - R_k(v^{k+1}) - DR_k(v^{k+1})(v^{k,i} - v^{k+1}) \\
&\quad + R_{k,\epsilon^{k,i}}(v^{k,i}) - R_k(v^{k,i}) + (DR_k(v^{k,i}) - DR_{k,\epsilon^{k,i}}(v^{k,i}))(v^{k,i} - v^{k+1}) \\
&\quad + (DR_k(v^{k+1}) - DR_k(v^{k,i}))(v^{k,i} - v^{k+1}))\| \\
&\le \|(DR_{k,\epsilon^{k,i}}(v^{k,i}))^{-1}\| \Big(\|R_k(v^{k,i}) - R_k(v^{k+1}) - DR_k(v^{k+1})(v^{k,i} - v^{k+1})\| \\
&\quad + \frac{1}{15\alpha\beta}\|R_k(v^{k,i}) - R_k(v^{k+1})\| + \|(DR_k(v^{k,i}) - DR_{k,\epsilon^{k,i}}(v^{k,i}))(v^{k,i} - v^{k+1})\| \\
&\quad + \|(DR_k(v^{k+1}) - DR_k(v^{k,i}))(v^{k,i} - v^{k+1})\| \Big) \\
&\le \frac{\alpha}{1 - \alpha\mu_{k,i}} \left(\frac{1}{15\alpha} + \frac{1}{5\alpha} + \frac{1}{5\alpha} + \frac{1}{10\alpha} \right) \|v^{k,i} - v^{k+1}\|.
\end{aligned}$$

Since

$$\frac{\alpha}{1 - \alpha\mu_{k,i}} \left(\frac{1}{15\alpha} + \frac{1}{5\alpha} + \frac{1}{5\alpha} + \frac{1}{10\alpha} \right) < \frac{5}{6},$$

we obtain $\|v^{k,i+1} - v^{k+1}\| < \frac{5}{6}\|v^{k,i} - v^{k+1}\|$. Then, from

$$\|R_{k,\epsilon^{k,i}}(v^{k,i})\| \le \|R_k(v^{k,i})\| + \|R_k(v^{k,i}) - R_{k,\epsilon^{k,i}}(v^{k,i})\| \le \|R_k(v^{k,i})\| + \frac{H_{\text{tol}}}{2},$$

there exists $i_k \in N$ such that $\|R_{k,\epsilon^{k,i}}(v^{k,i})\| \le H_{\text{tol}}$ holds for $i = i_k$;

(iii) For sufficiently small $H_{\text{tol}} > 0$, when $v^{k,i}$ sufficiently close to v^{k+1}, since

$$\|R_k(v^{k,i}) - R_{k,\epsilon^{k,i}}(v^{k,i})\| \le O(H_{\text{tol}}^2), \tag{36}$$

$$\|DR_k(v^{k,i}) - DR_{k,\epsilon^{k,i}}(v^{k,i})\| \le O(H_{\text{tol}}), \tag{37}$$

$$\|R_{k,\epsilon^{k,i}}(v^{k,i})\| \ge H_{\text{tol}}, \quad \forall\, i < i_k, \tag{38}$$

and

$$\|R_{k,\epsilon^{k,i}}(v^{k,i})\| = O(\|R_k(v^{k,i})\|),$$

which can be induced from (36) and (38), it has

$$\|R_k(v^{k,i}) - R_{k,\epsilon^{k,i}}(v^{k,i})\| \le O(\|R_k(v^{k,i})\|^2) = O(\|v^{k,i} - v^{k+1}\|^2) \tag{39}$$

and

$$\|DR_k(v^{k,i}) - DR_{k,\epsilon^{k,i}}(v^{k,i})\| \le O(\|R_k(v^{k,i})\|) = O(\|v^{k,i} - v^{k+1}\|). \tag{40}$$

Now, from (39) and (40), we have

$$\begin{aligned}
\|v^{k,i+1} - v^{k+1}\| &= \|(DR_{k,\epsilon^{k,i}}(v^{k,i}))^{-1}(R_k(v^{k,i}) - R_k(v^{k+1}) - DR_k(v^{k+1})(v^{k,i} - v^{k+1}) \\
&\quad + R_{k,\epsilon^{k,i}}(v^{k,i}) - R_k(v^{k,i}) + (DR_k(v^{k,i}) - DR_{k,\epsilon^{k,i}}(v^{k,i}))(v^{k,i} - v^{k+1}) \\
&\quad + (DR_k(v^{k+1}) - DR_k(v^{k,i}))(v^{k,i} - v^{k+1}))\| \\
&= O(\|v^{k,i} - v^{k+1}\|^2).
\end{aligned}$$

This completes the proof of the theorem. ■

4. Numerical experiment

In this section, we give some numerical results, comparing Algorithm 2 (TAH) with some other algorithms, such as the SQP method with active set strategy (SQP_AS) implemented by calling matlab function `fmincon`, CHIP method in [5] and ACH method in [25], to show the efficiency of our algorithm.

During the computation, we set parameters $h_0 = 0.1$, $h_{\max} = 0.3$, $\theta = 0.1$ or 0.01, $A_1 = A_2 = 1e2$, $\eta_{k,i} = 1e-1$, $\mu_{k,i} = 1e2$ for all $k, i \in N$. 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.

The numerical results reported below were obtained on discretized versions of three semi-infinite programming test problems and two general nonlinear programming problems. Example 4.4 is taken from [27], Example 4.5 from [22], and others are artificial. In Example 4.2, by setting $s_i = 0.5 + (\pi - 0.5)(i-1)/(q-1)$, the semi-infinite problem is discretized into finite programming. In Example 4.3, s in region $[0,1] \times [0,1]$ is discretized by setting $(s_{1i}, s_{2j}) = ((i-1)d, (j-1)d)$ with d equal to 0.11, 0.03 and 0.01, respectively. In Example 4.4, we discretize s in interval $[0,1]$ by setting $s_i = (i-1)/(q-1)$.

In Tables 1–5, q is the constraints number, x^* denotes the final approximate solution point, f^* is the value of the objective function at x^*, g^* is the constraint value at x^*, time is the CPU time in seconds. In Table 6, Iter and N_{grad} denote the total number of iterations and the average gradient evaluations, respectively.

Example 4.1 Let $x = (x_1, \ldots, x_6)^{\mathrm{T}} \in R^6$,

$$\min \quad \sum_{1 \le k \le 6} (x_k - 1)^2$$

Table 1. The numerical results of Example 4.1, $x^0 = (2, 2, 7, 0, -2, 1)^{\mathrm{T}}$.

q	n	Method	x^*	f^*	g^*	Time
1000	6	CHIP	$(0.999994, \ldots, 1.000000)$	2.079e−09	−0.459	1.798
		ACH	$(0.999999, \ldots, 1.000000)$	5.639e−10	−0.459	0.828
		TAH	$(0.999999, \ldots, 1.000000)$	6.770e−11	−0.459	0.350
		SQP_AS	$(0.999959, \ldots, 0.999803)$	6.471e−8	−0.460	2.594
5000	6	CHIP	$(0.999999, \ldots, 1.000000)$	1.993e−11	−0.459	20.727
		ACH	$(0.999999, \ldots, 1.000000)$	2.847e−10	−0.459	3.339
		TAH	$(0.999999, \ldots, 1.000000)$	5.006e−10	−0.459	0.760
		SQP_AS	$(0.999958, \ldots, 0.999808)$	6.284e−8	−0.460	2.792
10,000	6	CHIP	$(1.000000, \ldots, 1.000000)$	9.548e−12	−0.459	59.566
		ACH	$(0.999999, \ldots, 1.000000)$	1.083e−9	−0.459	7.070
		TAH	$(0.999999, \ldots, 1.000000)$	1.477e−10	−0.459	1.296
		SQP_AS	$(0.999958, \ldots, 0.999802)$	6.486e−8	−0.460	3.411

Table 2. The numerical results of Example 4.2, $x^0 = -(1, \ldots, 1)^{\mathrm{T}}$.

q	n	Method	x^*	f^*	g^*	Time
100	10	CHIP	$(-0.521488, \ldots, -0.521488)$	2.314925	−2.999e−5	6.333
		ACH	$(-0.521475, \ldots, -0.521475)$	2.314887	−7.753e−8	1.441
		TAH	$(-0.521475, \ldots, -0.521475)$	2.314887	−8.170e−8	0.266
		SQP_AS	$(-0.521476, \ldots, -0.521475)$	2.314887	−3.627e−12	1.801
100	30	CHIP	$(-0.348206, \ldots, -0.348206)$	1.817659	−1.001e−4	13.634
		ACH	$(-0.348192, \ldots, -0.348192)$	1.817621	−1.758e−7	2.821
		TAH	$(-0.348192, \ldots, -0.348192)$	1.817621	−1.804e−7	0.354
		SQP_AS	$(-0.521476, \ldots, -0.521475)$	1.817620	−2.007e−7	2.356
100	50	CHIP	$(-0.287732, \ldots, -0.287732)$	1.658255	−5.565e−4	61.496
		ACH	$(-0.287666, \ldots, -0.287666)$	1.658084	−5.827e−7	4.613
		TAH	$(-0.287666, \ldots, -0.287666)$	1.658084	−5.828e−7	0.469
		SQP_AS	$(-0.285098, \ldots, -0.285098)$	1.691913	−2.163e−8	2.481
50	20	CHIP	$(-0.404592, \ldots, -0.404592)$	1.972878	−8.348e−6	3.916
		ACH	$(-0.404590, \ldots, -0.404590)$	1.972874	−4.052e−8	1.230
		TAH	$(-0.404590, \ldots, -0.404590)$	1.972874	−4.129e−8	0.276
		SQP_AS	$(-0.404592, \ldots, -0.404591)$	1.972874	−1.927e−11	1.797
500	20	CHIP	$(-0.404671, \ldots, -0.404671)$	1.973101	−5.055e−4	193.530
		ACH	$(-0.404590, \ldots, -0.404590)$	1.972875	−1.445e−7	9.889
		TAH	$(-0.404590, \ldots, -0.404590)$	1.972874	−1.493e−7	0.451
		SQP_AS	$(-0.404028, \ldots, -0.405195)$	1.972872	−1.105e−9	2.051
1000	20	CHIP	$(-0.419417, \ldots, -0.419417)$	2.014746	−9.571e−2	739.025
		ACH	$(-0.404590, \ldots, -0.404590)$	1.972874	−8.445e−9	17.499
		TAH	$(-0.404590, \ldots, -0.404590)$	1.972874	−1.406e−8	0.538
		SQP_AS	$(-0.399028, \ldots, -0.409021)$	1.972677	−2.131e−7	2.701
1000	500	CHIP	Stop for time >6000			
		ACH	$(-0.118473, \ldots, -0.118473)$	1.250982	−5.661e−6	5.033e+3
		TAH	$(-0.118473, \ldots, -0.118473)$	1.250982	−6.261e−6	51.907
		SQP_AS	$(-0.664474, \ldots, -0.664474)$	2.770472	−7.335e+1	74.431

$$
\begin{aligned}
\text{s.t.}\quad & x_1 \exp(-x_2 s_j) \cos(x_3 s_j + x_4) \\
& \quad + \frac{x_3}{x_2} \exp(-s_j x_1) \sin(s_j x_2) + x_5 \exp(-x_6 s_j) - 2 \le 0, \quad j = 1, \ldots, q,
\end{aligned}
$$

where $s_j = 10q(j-1)/(q-1)$.

Table 3. The numerical results of Example 4.3, $x^0 = (2, 2, 2, 2, 2, 2)^T$.

q	n	Method	x^*	f^*	g^*	Time
100	6	CHIP	$(2.613440, -4.146074, \ldots, 4.240281)$	2.412662	$-1.226e-6$	1.033
		ACH	$(2.613445, -4.146085, \ldots, 4.240283)$	2.412660	$-1.120e-7$	0.261
		TAH	$(2.613445, -4.146085, \ldots, 4.240283)$	2.412660	$-1.032e-7$	0.292
		SQP_AS	$(2.613445, -4.146085, \ldots, 4.240283)$	2.412660	$2.670e-10$	1.913
1156	6	CHIP	$(2.512639, -3.953905, \ldots, 4.149019)$	2.417476	$-1.041e-6$	31.197
		ACH	$(2.512664, -3.953951, \ldots, 4.149036)$	2.417474	$-1.904e-7$	0.548
		TAH	$(2.512663, -3.953948, \ldots, 4.149035)$	2.417474	$-5.117e-7$	0.406
		SQP_AS	$(2.512665, -3.953953, \ldots, 4.149037)$	2.417474	$1.399e-14$	2.040
10,201	6	CHIP	$(2.503407, -3.974211, \ldots, 4.189137)$	2.435911	$-2.781e-5$	1.097e+3
		ACH	$(2.547929, -4.052878, \ldots, 4.223231)$	2.435643	$-1.746e-9$	4.293
		TAH	$(2.547929, -4.052878, \ldots, 4.223231)$	2.435643	$-1.746e-9$	1.400
		SQP_AS	$(2.547927, -4.052879, \ldots, 4.223226)$	2.435643	$8.882e-15$	3.517

Table 4. The numerical results of Example 4.4, $x^0 = (3, 3, 3, 5)^T$.

q	n	Method	x^*	f^*	g^*	Time
100	4	CHIP	$(0.993302, 0.673475, -0.389950, 0.000049)$	$4.897e-5$	$-4.111e-6$	0.916
		ACH	$(0.995705, 0.668208, -0.388173, 0.000019)$	$1.856e-5$	$-1.102e-7$	0.445
		TAH	$(0.995705, 0.668208, -0.388846, 0.000019)$	$1.856e-5$	$-1.104e-7$	0.396
		SQP_AS	$(0.995697, 0.668237, -0.388378, 0.000018)$	$1.842e-5$	$9.154e-8$	2.361
1000	4	CHIP	$(0.991845, 0.676942, -0.387670, 0.000332)$	$3.323e-4$	$-2.658e-4$	15.583
		ACH	$(0.995703, 0.668210, -0.388846, 0.000019)$	$1.856e-5$	$-9.764e-8$	1.140
		TAH	$(0.995703, 0.668210, -0.388847, 0.000019)$	$1.856e-5$	$-1.007e-7$	0.558
		SQP_AS	$(0.995684, 0.668276, -0.388828, 0.000018)$	$1.838e-5$	$2.467e-7$	4.492
10,000	4	CHIP	$(0.991752, 0.677103, -0.387668, 0.000757)$	$7.569e-4$	$-6.889e-4$	8.530e+2
		ACH	$(0.995702, 0.668212, -0.388847, 0.000019)$	$1.856e-5$	$-8.014e-8$	10.051
		TAH	$(0.995702, 0.668212, -0.388847, 0.000019)$	$1.856e-5$	$-8.022e-8$	2.642
		SQP_AS	$(0.995708, 0.668171, -0.388858, 0.000018)$	$1.842e-5$	$9.796e-8$	69.155

Table 5. The numerical results of Example 4.5, $x^0 = (-1.5, 1.5, 2.5, 1.5)^T, h^0 = 9$.

q	n	Method	x^*	f^*	g^*	Time
100	5	CHIP	Stop for time >6000			
		ACH	$(-0.997280, 0.983042, 1.975640, 0.995934)$	0.543863	$-2.437e-8$	0.656
		TAH	$(-0.997280, 0.983042, 1.975640, 0.995934)$	0.543863	$-4.704e-9$	0.428
		SQP_AS	$(-0.997280, 0.983042, 1.975640, 0.995934)$	0.543863	$1.611e-13$	2.104
1000	5	CHIP	Stop for time >6000			
		ACH	$(-0.985203, 0.828095, 1.988116, 1.022360)$	1.075140	$-7.059e-9$	3.898
		TAH	$(-0.985203, 0.828095, 1.988116, 1.022360)$	1.075140	$-4.518e-7$	0.460
		SQP_AS	$(-0.985203, 0.828095, 1.988116, 1.022360)$	1.075140	$1.177e-14$	1.762
10,000	5	CHIP	Stop for time >6000			
		ACH	$(-0.965716, 0.974188, 1.968446, 0.988166)$	1.301831	$-9.662e-9$	49.366
		TAH	$(-0.965716, 0.974188, 1.968446, 0.988166)$	1.301831	$-2.487e-9$	0.764
		SQP_AS	$(-0.965716, 0.974188, 1.968446, 0.988166)$	1.301831	$9.368e-13$	2.980

Table 6. The numerical results of Example 4.1, $x^0 = (2, 2, 7, 0, -2, 1)^T$.

	$q = 1000$		$q = 5000$		$q = 10,000$	
Method	Iter	N_{grad}	Iter	N_{grad}	Iter	N_{grad}
TAH	56	25	57	135	57	281
ACH	56	1000	57	5000	57	10,000

Example 4.2 Let $x = (x_1, \ldots, x_n)^{\mathrm{T}} \in R^n$,

$$\begin{aligned} \min \quad & \sum_{1 \le k \le n} \frac{(x_k - 1)^2}{n} \\ \text{s.t.} \quad & \prod_{1 \le k \le n} \cos(s x_k) + s \sum_{1 \le k \le n} x_k^3 \le 0, \quad \text{for } s \in [0.5, \pi]. \end{aligned}$$

Example 4.3 Let $x = (x_1, x_2, x_3, x_4, x_5, x_6)^{\mathrm{T}} \in R^6$,

$$\begin{aligned} \min \quad & x_1 + \frac{1}{2x_2} + \frac{1}{2x_3} + \frac{1}{3x_4} + \frac{1}{4x_5} + \frac{1}{3x_6} \\ \text{s.t.} \quad & \exp(s_1^2 + s_2^2) - x_1 - x_2 s_1 - x_3 s_2 - x_4 s_1^2 - x_5 s_1 s_2 - x_6 s_2^2 \le 0, \quad \text{for } s \in [0,1] \times [0,1]. \end{aligned}$$

Example 4.4 Let $x = (x_1, x_2, x_3, x_4)^{\mathrm{T}} \in R^4$,

$$\begin{aligned} \min \quad & x_4 \\ \text{s.t.} \quad & \left(\exp(s) - \frac{x_1 + s x_2}{1 + s x_3} \right)^2 - x_4 \le 0, \quad \text{for } s \in [0,1]. \end{aligned} \tag{41}$$

Example 4.5 The problem of fitting enclosing cylinders to data in R^n has been presented by many literatures [1,3,22]. Let data points $z_j \in R^n, j = 1, \ldots, q$, be given and let a d-dimensional linear manifold be sought by l_∞ orthogonal distance regression, so that the Chebyshev norm of the vector of orthogonal distance from the data points to the manifold is minimized. In [22], a special case for $n = 3$ and $d = 1$ was considered. The linear manifold is given by

$$l(x, s) = \begin{bmatrix} x_1 s + x_2 \\ x_3 s + x_4 \\ s \end{bmatrix}, \tag{42}$$

where $x \in R^4$ fixes the line and s is a scalar parameter. For any data point z_j, $s_j(x)$, the parameter which gives the nearest point on the line defined by x, is given by

$$s_j(x) = \frac{(x_1, x_3, 1) z_j - x_1 x_2 - x_3 x_4}{x_1^2 + x_3^2 + 1}, \quad j = 1, \ldots, q,$$

and the orthogonal distance vectors are defined by

$$d_j(x) = z_j - l(x, s_j(x)), \quad j = 1, \ldots, q.$$

Then the problem to be solved is

$$\begin{aligned} \min \quad & h \\ \text{s.t.} \quad & \|d_j(x)\|_2^2 \le h, \quad j = 1, \ldots, q. \end{aligned} \tag{43}$$

The data points $z_j,\ j = 1, \ldots, q$ were generated as follows:

$$r_j = -5 + \frac{10(j-1)}{q-1}, \quad z_j^1 = -r_j + 1 + \sigma_j^1, \quad z_j^2 = 2r_j + 1 + \sigma_j^2, \quad z_j^3 = r_j + \sigma_j^3,$$

where the residuals $\sigma^1, \sigma^2, \sigma^3 \sim N(0, 0.3^2)$.

Table 7. The CPU time of Example 4.1 with changeable A_1, A_2, $\eta_{k,i}$, $\mu_{k,i}$.

A_1	A_2	$\eta_{k,i}$	$\mu_{k,i}$	Time
1e2	1e2	1e−1	1e2	0.760
1e1	1e1	1e−1	1e2	0.652
1e0	1e0	1e−1	1e2	0.616
1e0	1e0	1e0	1e2	0.614
1e0	1e0	1e1	1e2	0.613
1e0	1e0	1e0	1e1	0.614
1e0	1e0	1e0	1e0	0.642

To show the effect of the truncation more clearly, the total number of iterations and the average gradient evaluations during all iterations in TAH and ACH for Example 4.1 are listed in Table 6. It can be seen that the adopting of truncation does not increase the number of iterations, but dramatically reduce the gradient calculations.

In the following, a test on Example 4.1 ($q = 5000$) for different parameter values is given, and the result is listed in Table 7.

Result in Table 7 shows that the performance of our algorithm moderately depends on the values of parameters A_1, A_2, $\eta_{k,i}$ and $\mu_{k,i}$. In our experience, these parameters can generally take values in a rather considerable range. This may be caused by that ϵ is moderately dependent on these parameters because of the `ln` operation (see (13) and (14)).

5. Concluding remarks

We have proposed an efficient implementation of aggregate homotopy method for nonconvex nonlinear programming problems. Instead of aggregating all the constraints, truncated aggregate technique was adopted such that only a small adaptively updated subset of the constraints is aggregated at each iteration, and hence the number of gradient and Hessian calculations is reduced dramatically. Moreover, the truncating criterions, concerning only with computation of function values, can adaptively update the subset to guarantee the locally quadratic convergence of the correction process with as few computation cost as possible. It can be seen from the above test results that the TAH method is very efficient, especially for the problems with large number of constraints and computationally expensive gradients or Hessians. Though some parameters are involved in truncating criterions, our algorithm is comparatively stable for moderate change of parameter value.

Assumptions (C_1), (C_2), (C_3^*) and (C_4) are sufficient for convergence analysis of aggregate homotopy method. We did not verify whether the numerical examples satisfy these assumptions. It should be noted that if assumption (C_3^*) does not hold, some potential ill-condition of coefficient matrix at inner iteration maybe arised. Some discussions on ill-conditioning arising from interior point methods have been given by Wright et al. [23]. It will be one of our on-going work based on existing researches to make more further consideration on potential ill-conditioning caused by degeneracy.

Acknowledgements

The authors like to express thanks to anonymous reviewers, whose precise and substantial remarks led to an improved version of the paper.

This work was supported by the National Natural Science Foundation of China (11171051, 11126172, 91230103 and 11261019) and the Fundamental Research Funds for the Central Universities.

References

[1] P.K. Agarwal, B. Aronov, and M. Sharir, *Line traversals of balls and smallest enclosing cylinders in three dimensions*, Discrete Comput. Geom. 21 (1999), pp. 373–388.

[2] E.L. Allgower and K. Georg, *Numerical path following*, in *Handbook of Numerical Analysis*, P.G. Ciarlet and J.L. Lions, eds., Vol. 5, Elsevier, North-Holland, 1997, pp. 3–207.

[3] R. Brenberg and T. Theobald, *Algebraic methods for computing smallest enclosing and circumscribing cylinders of simplices*, Appl. Algebra Eng. Commun. Comput. 14 (2004), pp. 439–460.

[4] G.C. Feng and B. Yu, *Combined homotopy interior point method for nonlinear programming problems*, in *Advances in Numerical Mathematics*, H. Fujita and M. Yamaguti, eds., *Proceedings of the Second Japan–China Seminar on Numerical Mathematics, Lecture Notes in Numerical and Applied Analysis*, Vol. 14, Kinokuniya, Tokyo, 1995, pp. 9–16.

[5] G.C. Feng, Z.H. Lin, and B. Yu, *Existence of an interior pathway to a Karush–Kuhn–Tucker point of a nonconvex programming problem*, Nonlinear Anal. 32 (1998), pp. 761–768.

[6] K. Georg, *On tracing an implicitly defined curve by quasi-Newton steps and calculating bifurcation by local perturbations*, SIAM J. Sci. Stat. Comput. 2 (1981), pp. 35–50.

[7] H.B. Keller, *Numerical solution of bifurcation and nonlinear eigenvalue problems*, in *Application of Bifurcation Theory*, P.H. Rabinowitz, ed., Academic Press, New York, pp. 359–384.

[8] B.W. Kort and D.P. Bertsekas, *A New Penalty Function Algorithm for Constrained Minimization*, Proceedings of the 1972 IEEE Conference on Decision and Control, New Orleans, LA, 1972.

[9] M. Kubicek, *Dependence of solution of nonlinear systems on a parameter, Algorithm 502*, ACM Trans. Math. Softw. 2 (1976), pp. 98–107.

[10] X.S. Li, *An aggregate function method for nonlinear programming*, Sci. China Ser. A 34 (1991), pp. 1467–1473.

[11] X.S. Li, *An aggregate function constraint method for nonlinear programming*, J. Oper. Res. Soc. 42 (1991), pp. 1003–1010.

[12] X.S. Li, *An entropy-based aggregate method for minmax optimization*, Eng. Optim. 18 (1992), pp. 227–285.

[13] X.S. Li and S.C. Fang, *On the entropic regularization method for solving min–max problems with applications*, Math. Methods Oper. Res. 46 (1997), pp. 119–130.

[14] X.S. Li and S. Pan, *Solving the finite min–max problem via an exponential penalty method*, Comput. Tech. 8 (2003), pp. 3–15.

[15] Z.H. Lin, B. Yu, and G.C. Feng, *A combined homotopy interior point method for convex nonlinear programming*, Appl. Math. Comput. 84 (1997), pp. 193–211.

[16] R. Menzel and H. Schwetlick, *Zur Lösung parameterabhängiger nichtlinearer Gleichungen mit singulären Jacobi-Matrizen*, Numer. Math. 30 (1978), pp. 65–79.

[17] J.M. Peng, *A smoothing function and its applications*, in *Reformulation-Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, M. Fukushima and L. Qi, eds., Kluwer Academic Publisher, Dordrecht, 1998, pp. 293–321.

[18] J.M. Peng and Z.H. Lin, *A non-interior continuation method for generalized linear complementarity problems*, Math. Program. 86 (1999), pp. 533–563.

[19] W.C. Rheinboldt, *Solution fields of nonlinear equations and continuation methods*, SIAM J. Numer. Anal. 17 (1980), pp. 221–237.

[20] J. Sun and L.W. Zhang, *On the log-exponential trajectory of linear programming*, J. Global Optim. 25 (2003), pp. 75–90.

[21] H.W. Tang and L.W. Zhang, *A maximum entropy method for the convex programming*, Chin. Sci. Bull. 39 (1994), pp. 682–684.

[22] G.A. Watson, *Fitting enclosing cylinders to data in R^n*, Numer. Algorithms 43 (2006), pp. 189–196.

[23] M.H. Wright, *Ill-conditioning and computational error in interior methods for nonlinear programming*, SIAM J. Optim. 9 (1998), pp. 84–111.

[24] Y. Xiao and B. Yu, *A truncated aggregate smoothing newton method for minimax problems*, Appl. Math. Comput. 216 (2010), pp. 1868–1879.

[25] B. Yu, G.C. Feng, and S.L. Zhang, *The aggregate constraint homotopy method for nonconvex nonlinear programming*, Nonlinear Anal. TMA 45 (2001), pp. 839–847.

[26] B. Yu, G.X. Liu, and G.C. Feng, *The aggregate homotopy method for constrained sequential max–min problems*, Northeastern Math. J. 4 (2003), pp. 287–290.

[27] J.L. Zhou and A.L. Tits, *An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective function*, SIAM J. Numer. Anal. 6 (1996), pp. 461–487.