Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 430932, 14 pages
http://dx.doi.org/10.1155/2014/430932

Research Article

The Flattened Aggregate Constraint Homotopy Method for Nonlinear Programming Problems with Many Nonlinear Constraints

Zhengyong Zhou[1] and Bo Yu[2]

[1] School of Mathematics and Computer Sciences, Shanxi Normal University, Linfen, Shanxi 041004, China
[2] School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning 116024, China

Correspondence should be addressed to Zhengyong Zhou; zzy198300@163.com

Received 8 February 2014; Accepted 27 April 2014; Published 29 May 2014

Academic Editor: Victor Kovtunenko

The aggregate constraint homotopy method uses a single smoothing constraint instead of m-constraints to reduce the dimension of its homotopy map, and hence it is expected to be more efficient than the combined homotopy interior point method when the number of constraints is very large. However, the gradient and Hessian of the aggregate constraint function are complicated combinations of gradients and Hessians of all constraint functions, and hence they are expensive to calculate when the number of constraint functions is very large. In order to improve the performance of the aggregate constraint homotopy method for solving nonlinear programming problems, with few variables and many nonlinear constraints, a flattened aggregate constraint homotopy method, that can save much computation of gradients and Hessians of constraint functions, is presented. Under some similar conditions for other homotopy methods, existence and convergence of a smooth homotopy path are proven. A numerical procedure is given to implement the proposed homotopy method, preliminary computational results show its performance, and it is also competitive with the state-of-the-art solver KNITRO for solving large-scale nonlinear optimization.

1. Introduction

In this paper, we consider the following nonlinear programming problem:

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & g(x) \le 0, \end{aligned} \tag{1}$$

where $x \in R^n$ is the variable, $g(x) = (g_1(x), \ldots, g_m(x))^T$, $f : R^n \to R$, and $g_i(x) : R^n \to R$, $i = 1, \ldots, m$, are three times continuously differentiable, and m is very large, but n is moderate. It has wide applications, and a typical situation is the discretized semi-infinite programming problem.

From the mid-1980s, much attention has been paid to interior point methods for mathematical programming, and many results on theory, algorithms, and applications on linear programming, convex programming, complementarity problems, semidefinite programming, and linear cone programming were obtained (see monographs [1–6] and references therein). For nonlinear programming, the typical algorithms used were the Newton-type methods to the perturbed first-order necessary conditions combined with line search or trust region methods with a proper merit function (e.g., [7–9]). The general conditions of global convergence for these methods required that the feasible set be bounded and that the Jacobian matrix be uniformly nonsingular. Another typical class of globally convergent methods for nonlinear programming was probability-one homotopy methods (e.g., [10–12]), whose global convergence can be established under some weaker conditions than the ones for Newton-type methods. The excellent feature is that, unlike line search or trust region methods, they do not depend on the descent of a merit function and so are insensitive to the local minimum of the merit function, in which any search direction is not a descent direction of the merit function.

In [10, 11], Feng et al. proposed a homotopy method for nonlinear programming (1), which was called the combined homotopy interior point (abbreviated by CHIP) method;

its global convergence was proven under the normal cone condition (see below for its definition) for the feasible set as well as some common conditions. On the basis of the CHIP method, some modified CHIP methods were presented in [13, 14]; the global convergence was established under the quasinormal cone and pseudocone condition for the feasible set, respectively. In [12], Watson described some probability-one homotopy methods for the unconstrained and inequality constrained optimization, whose global convergence was established under some weaker assumptions. Recently, Yu and Shang proposed a constraint shifting combined homotopy method in [15, 16], in which not only the objective function but also the constraint functions were regularly deformed. The global convergence was proven under the condition that the initial feasible set, which approaches the feasible set of (1) as the homotopy parameter changes from 1 to 0, not necessarily the feasible set of (1), satisfies the normal cone condition.

Let $g_{\max}(x) = \max_{1\le i\le m}\{g_i(x)\}$; then (1) is equivalent to

$$\begin{aligned} &\min \quad f(x), \\ &\text{s.t.} \quad g_{\max}(x) \le 0, \end{aligned} \tag{2}$$

which has only one, but nonsmooth, constraint. In [17], the following aggregate function was introduced, which is a smooth approximation of $g_{\max}(x)$ with a smoothing parameter $t > 0$ and induced from the max-entropy theory:

$$\widehat{g}(x,t) = t \ln \sum_{i=1}^{m} \exp\left(\frac{g_i(x)}{t}\right). \tag{3}$$

It is also known as exponential penalty function (see [18]). By using it for all constraint functions of problem (1), an aggregate constraint homotopy (abbreviated by ACH) method was presented by Yu et al. in [19], whose global convergence was obtained under the condition that the feasible set satisfies the weak normal cone condition. Although the ACH method has only one smoothing constraint, the gradient and Hessian of the aggregate constraint function

$$\begin{aligned} \nabla_x \widehat{g}(x,t) &= \sum_{i=1}^{m} c_i(x,t) \nabla g_i(x), \\ \frac{\partial \widehat{g}(x,t)}{\partial t} &= \frac{1}{t}\left(\widehat{g}(x,t) - \sum_{i=1}^{m} c_i(x,t) g_i(x)\right), \\ \nabla_{xx}^2 \widehat{g}(x,t) &= \sum_{i=1}^{m} c_i(x,t) \nabla^2 g_i(x) \\ &\quad + \frac{1}{t}\sum_{i=1}^{m} c_i(x,t) \nabla g_i(x) \left(\nabla g_i(x)\right)^T \\ &\quad - \frac{1}{t}\sum_{i=1}^{m} c_i(x,t) \nabla g_i(x) \sum_{i=1}^{m} c_i(x,t) \left(\nabla g_i(x)\right)^T, \\ \nabla_{xt}^2 \widehat{g}(x,t) &= \frac{1}{t^2}\left(\sum_{i=1}^{m} c_i(x,t) \nabla g_i(x) \sum_{i=1}^{m} c_i(x,t) g_i(x) \right. \\ &\quad \left. - \sum_{i=1}^{m} c_i(x,t) g_i(x) \nabla g_i(x)\right), \end{aligned} \tag{4}$$

where

$$c_i(x,t) = \frac{\exp\left(g_i(x)/t\right)}{\sum_{i=1}^{m} \exp\left(g_i(x)/t\right)} \in (0,1), \qquad \sum_{i=1}^{m} c_i(x,t) = 1, \tag{5}$$

are complicated combinations of gradients and Hessians of all constraint functions and, hence, are expensive to calculate when m is very large.

Throughout this paper, we assume that the nonlinear programming problem (1) possesses a very large number of nonlinear constraints, but a small number of variables, and the objective and constraint functions are not sparse. For such a problem, the number of constraint functions can be so large that the computation of gradients and Hessians of all constraint functions is very expensive and cannot be stored in memory and, hence, the general numerical methods for solving nonlinear programming are not efficient. Although active set methods only need to calculate gradients and Hessians of a part of constraint functions, require lower storage, and have faster numerical solution, the working set is difficult to estimate without knowing the internal structure of the problem.

In this paper, we present a new homotopy method called the flattened aggregate constraint homotopy (abbreviated by FACH) method for nonlinear programming (1) by using a new smoothing technique, in which only a part of constraint functions is aggregated. Under the normal cone condition for the feasible set and some other general assumptions, we prove that the FACH method can determine a smooth homotopy path from a given interior point of the feasible set to a KKT point of (1), and preliminary numerical results demonstrate its efficiency.

The rest of this paper is organized as follows. We conclude this section with some notations, definitions, and a lemma. The flattened aggregate constraint function with some properties is given in Section 2. The homotopy map, existence and convergence of a smooth homotopy path with proof are given in Section 3. A numerical procedure for tracking the smooth homotopy path, and numerical test results with some remarks are given in Section 4. Finally, we conclude the paper with some remarks in Section 5.

When discussing scalars and scalar-valued functions, subscripts refer to iteration step so that superscripts can be used for exponentiation. In contrast, for vectors and vector-valued functions, subscripts are used to indicate components, whereas superscripts are used to indicate the iteration step. The identity matrix is represented by I.

Unless otherwise specified, $\|\cdot\|$ denotes the Euclidean norm. $\Omega = \{x \in R^n \mid g(x) \le 0\}$ is the feasible set of (1), whereas $\Omega^0 = \{x \in R^n \mid g(x) < 0\}$ is the interior of Ω and $\partial\Omega = \Omega \setminus \Omega^0$ is the boundary of Ω. The symbols R_+^m and R_{++}^m denote the nonnegative and positive quadrants of R^m, respectively. The active index set is denoted by $I(x) = \{i \in \{1,\ldots,m\} \mid g_i(x) = 0\}$ at $x \in \Omega$. For a function $F : R^n \to R^m$, $\nabla F(x)$ is the $n \times m$ matrix whose (i, j)th element is $\partial F_j(x)/\partial x_i$; $F^{-1}(Y) = \{x \mid F(x) \in Y\}$ is the set-valued inverse for the set $Y \subseteq R^m$.

Definition 1 (see [10]). The set Ω satisfies the normal cone condition if and only if the normal cone of Ω at any $x \in \partial\Omega$ does not meet Ω^0; that is,

$$\left\{x + \sum_{i\in I(x)} y_i \nabla g_i(x) \mid y_i \geq 0\right\} \cap \Omega^0 = \emptyset. \tag{6}$$

Definition 2. If and only if there exists $\lambda^* \in R_+^m$ such that (x^*, λ^*) satisfies

$$\begin{gathered} \nabla f(x^*) + \nabla g(x^*)\lambda^* = 0, \\ \lambda_i^* g_i(x^*) = 0, \quad \lambda_i^* \geq 0, \ g_i(x^*) \leq 0, \ i = 1,\ldots,m, \end{gathered} \tag{7}$$

then x^* is called a KKT point of (1) and λ^* is the corresponding Lagrangian multiplier.

Definition 3. Let $U \subset R^n$ be an open set and let $F : U \to R^m$ be a differentiable map. We say $y \in R^m$ is a regular value of F if and only if

$$\text{Range}\left[\frac{\partial F(x)}{\partial x}\right] = R^m, \quad \forall x \in F^{-1}(y). \tag{8}$$

Lemma 4 (parameterized Sard theorem [20]). *Let $U \subset R^n$ and $V \subset R^k$ be two open sets and let $F : U \times V \to R^m$ be a C^r differentiable map with $r > \max\{0, n-m\}$. If $0 \in R^m$ is a regular value of F, then, for almost all $a \in V$, 0 is a regular value of $F_a = F(\cdot, a)$.*

2. The Flattened Aggregate Constraint Function

In this section, we suppose that the following assumptions hold.

Assumption 5. Ω^0 is nonempty and Ω is bounded.

Assumption 6. For any $x \in \partial\Omega$, $\{\nabla g_i(x) \mid i \in I(x)\}$ are positive independent; that is,

$$\sum_{i\in I(x)} y_i \nabla g_i(x) = 0, \quad y_i \geq 0, \ i \in I(x) \Longrightarrow y_i = 0, \ i \in I(x). \tag{9}$$

Assumption 7. Ω satisfies the normal cone condition (see Definition 1).

The first assumption is the Slater's condition and the boundedness of the feasible set, which are two basic conditions. The second assumption provides the regularity of constraints, which is weaker than the linear independence constraint qualification. The last assumption, the normal cone condition of the feasible set, is a generalization of the convex condition for the feasible set. Indeed, if Ω is a convex set, then it satisfies the normal cone condition. Some simple nonconvex sets, satisfying the normal cone condition, are shown in [10].

In this paper, we construct a flattened aggregate constraint function as follows:

$$\begin{aligned} &\widehat{g}_{(\theta,c_1,c_2,\alpha)}(x,t) \\ &\quad = \theta t \ln\left(\sum_{i=1}^m \varphi(g_i(x),t)\exp\left(\frac{g_i(x)}{\theta t}\right) + \exp\left(-\frac{\varepsilon(t)}{\theta t}\right)\right), \end{aligned} \tag{10}$$

where

$$\varepsilon(t) = c_1 t + c_2, \tag{11}$$

$c_1 > 0$, $c_2 > 0$, $0 < \theta \leq 1$ are some adjusting parameters, $\varphi(z,t) : R \times [0,1] \to [0,1]$ is a C^3 bivariate function satisfying $\varphi(z,t) = 0$ for $z \leq -\alpha\varepsilon(t)$ with $\alpha > 1$, and $\varphi(z,t) = 1$ for $z \geq -\varepsilon(t)$. For simplicity of discussion, we write $\widehat{g}_{(\theta,c_1,c_2,\alpha)}(x,t)$ as $\widehat{g}(x,t)$. By its definition, $\varphi(z,t)$ can be expressed as

$$\varphi(z,t) = \begin{cases} 0, & \text{for } z \leq -\alpha\varepsilon(t) \ \text{ with } \alpha > 1, \\ 1, & \text{for } z \geq -\varepsilon(t), \\ \overline{\varphi}(z,t), & \text{otherwise,} \end{cases} \tag{12}$$

where, for any $t \in [0,1]$, $\overline{\varphi}(z,t)$ satisfies

$$\begin{gathered} \overline{\varphi}(-\alpha\varepsilon(t),t) = 0, \qquad \frac{\partial\overline{\varphi}(-\alpha\varepsilon(t),t)}{\partial z} = 0, \\ \frac{\partial^2\overline{\varphi}(-\alpha\varepsilon(t),t)}{\partial z^2} = 0, \qquad \frac{\partial^3\overline{\varphi}(-\alpha\varepsilon(t),t)}{\partial z^3} = 0, \\ \overline{\varphi}(-\varepsilon(t),t) = 1, \qquad \frac{\partial\overline{\varphi}(-\varepsilon(t),t)}{\partial z} = 0, \\ \frac{\partial^2\overline{\varphi}(-\varepsilon(t),t)}{\partial z^2} = 0, \qquad \frac{\partial^3\overline{\varphi}(-\varepsilon(t),t)}{\partial z^3} = 0. \end{gathered} \tag{13}$$

There exist many kinds of functions $\overline{\varphi}(z,t)$, such as polynomial functions and spline functions. Since polynomial functions have simple mathematical expressions and many excellent properties, throughout this paper, $\overline{\varphi}(z,t)$ is chosen as

$$\begin{aligned} \overline{\varphi}(z,t) = {} & \frac{-20(z+\varepsilon(t))^7}{(\alpha\varepsilon(t)-\varepsilon(t))^7} + \frac{-70(z+\varepsilon(t))^6}{(\alpha\varepsilon(t)-\varepsilon(t))^6} \\ & + \frac{-84(z+\varepsilon(t))^5}{(\alpha\varepsilon(t)-\varepsilon(t))^5} + \frac{-35(z+\varepsilon(t))^4}{(\alpha\varepsilon(t)-\varepsilon(t))^4} + 1. \end{aligned} \tag{14}$$

Let $I_\varepsilon(x,t) = \{i \mid g_i(x) > -\alpha\varepsilon(t)\}$; by the definition of $\varphi(z,t)$, $\widehat{g}(x,t)$ in (10) can be rewritten as

$$\widehat{g}(x,t) = \theta t \ln\left(\sum_{i\in I_\varepsilon(x,t)} \varphi(g_i(x),t)\exp\left(\frac{g_i(x)}{\theta t}\right) + \exp\left(-\frac{\varepsilon(t)}{\theta t}\right)\right). \tag{15}$$

The gradient of $\widehat{g}(x,t)$ with respect to x is

$$\nabla_x \widehat{g}(x,t) = \sum_{i\in I_\varepsilon(x,t)} (\lambda_i(x,t) + \xi_i(x,t))\nabla g_i(x), \tag{16}$$

where

$$\lambda_i(x,t) = \frac{\varphi(g_i(x),t)\exp(g_i(x)/\theta t)}{\sum_{j\in I_\varepsilon(x,t)}\varphi(g_j(x),t)\exp(g_j(x)/\theta t) + \exp(-\varepsilon(t)/\theta t)}, \tag{17}$$

$$\xi_i(x,t) = \frac{\theta t\partial\varphi(g_i(x),t)/\partial z\ \exp(g_i(x)/\theta t)}{\sum_{j\in I_\varepsilon(x,t)}\varphi(g_j(x),t)\exp(g_j(x)/\theta t) + \exp(-\varepsilon(t)/\theta t)}. \tag{18}$$

Then the gradient and Hessian of $\widehat{g}(x,t)$ only relate to a part of constraint functions; that is, $i \in I_\varepsilon(x,t)$.

Proposition 8. *For any $\theta > 0$, $t \in (0,1]$, $x \in \Omega$ with $\theta t \downarrow 0$, $t \to \bar{t} \in [0,1]$, $x \to \bar{x} \in \partial\Omega$,*

(1) *$\lambda_i(x,t) \to 0$ for $i \notin I(\bar{x})$;*

(2) *$\sum_{i\in I(\bar{x})}\lambda_i(x,t) \to 1$;*

(3) *$\xi(x,t) \to 0$.*

Proof. (1) Because $\bar{x} \in \partial\Omega$, there exists $i^* \in I(\bar{x})$ such that $g_{i^*}(\bar{x}) = 0$. By the continuity of $\varphi(z,t)$ and $g(x)$, we know that $\lim_{x\to\bar{x},t\to\bar{t}}\varphi(g_{i^*}(x),t) = \varphi(g_{i^*}(\bar{x}),\bar{t}) = 1$ and $\lim_{x\to\bar{x}}(g_i(x) - g_{i^*}(x)) = g_i(\bar{x}) < 0$ for $i \notin I(\bar{x})$; hence

$$\begin{aligned} 0 &\le \lambda_i(x,t) \\ &\le \frac{\exp(g_i(x)/\theta t)}{\varphi(g_{i^*}(x),t)\exp(g_{i^*}(x)/\theta t)} \\ &= \frac{\exp((g_i(x) - g_{i^*}(x))/\theta t)}{\varphi(g_{i^*}(x),t)} \\ &\longrightarrow 0. \end{aligned} \tag{19}$$

Hence, item 1 is satisfied for $\theta t \downarrow 0$, $t \to \bar{t}$, and $x \to \bar{x}$.

(2) By $\lim_{x\to\bar{x},t\to\bar{t}}(-\varepsilon(t) - g_{i^*}(x)) = -\varepsilon(\bar{t}) < 0$, we have

$$\begin{aligned} 1 &\ge \sum_{i\in I_\varepsilon(x,t)}\lambda_i(x,t) \\ &= \frac{\sum_{i\in I_\varepsilon(x,t)}\varphi(g_i(x),t)\exp(g_i(x)/\theta t)}{\sum_{j\in I_\varepsilon(x,t)}\varphi(g_j(x),t)\exp(g_j(x)/\theta t) + \exp(-\varepsilon(t)/\theta t)} \\ &\ge \frac{\varphi(g_{i^*}(x),t)\exp(g_{i^*}(x)/\theta t)}{\varphi(g_{i^*}(x),t)\exp(g_{i^*}(x)/\theta t) + \exp(-\varepsilon(t)/\theta t)} \\ &\ge \frac{\varphi(g_{i^*}(x),t)}{1 + \exp((-\varepsilon(t) - g_{i^*}(x))/\theta t)} \\ &\longrightarrow 1. \end{aligned} \tag{20}$$

Hence, $\sum_{i\in I_\varepsilon(x,t)}\lambda_i(x,t) \to 1$. Together with item 1, we have $\sum_{i\in I(\bar{x})}\lambda_i(x,t) \to 1$ for $\theta t \downarrow 0$, $t \to \bar{t}$, and $x \to \bar{x}$.

(3) If $g_i(x) \ge -\varepsilon(t)$, then $\partial\varphi(g_i(x),t)/\partial z = 0$ by its definition, and hence

$$\xi_i(x,t) = 0; \tag{21}$$

else

$$|\xi_i(x,t)| \le \frac{\theta t\,|\partial\varphi(g_i(x),t)/\partial z|\exp(g_i(x)/\theta t)}{\exp(-\varepsilon(t)/\theta t)} \le \theta t M, \tag{22}$$

where $M = \max_{(z,t)\in R\times[0,1]}|\partial\varphi(z,t)/\partial z|$. Then, we have $\xi(x,t) \to 0$ for $\theta t \downarrow 0$, $t \to \bar{t}$, and $x \to \bar{x}$. □

Proposition 9. *For any given $0 < \theta \le 1$,*

(1) *$-\varepsilon(t) \le \widehat{g}(x,t) \le \max\{g_{\max}(x), -\varepsilon(t)\} + \theta t\ln(m_{I_\varepsilon} + 1)$,*

(2) *$g_{\max}(x) \le \widehat{g}(x,t) \le g_{\max}(x) + \theta t\ln(m_{I_\varepsilon} + 1)$ when $g_{\max}(x) \ge -\varepsilon(t)$,*

(3) *$\widehat{g}(x,t) = -\varepsilon(t)$ when $g_{\max}(x) \le -\alpha\varepsilon(t)$ with $\alpha > 1$,*

where m_{I_ε} denotes the cardinality of $I_\varepsilon(x,t)$.

Proof. (1) By $\varphi(z,t) \ge 0$, we have

$$\begin{aligned} \widehat{g}(x,t) &= \theta t\ \ln\left(\sum_{i\in I_\varepsilon(x,t)}\varphi(g_i(x),t)\exp\left(\frac{g_i(x)}{\theta t}\right) + \exp\left(-\frac{\varepsilon(t)}{\theta t}\right)\right) \\ &\ge \theta t\ \ln\left(\exp\left(-\frac{\varepsilon(t)}{\theta t}\right)\right) \\ &= -\varepsilon(t). \end{aligned} \tag{23}$$

Then the left inequality can be obtained. By $\varphi(z,t) \le 1$, we have

$$\begin{aligned}\widehat{g}(x,t) &= \theta t \ln\left(\sum_{i\in I_\varepsilon(x,t)} \varphi(g_i(x),t)\exp\left(\frac{g_i(x)}{\theta t}\right)\right. \\ &\qquad \left. + \exp\left(-\frac{\varepsilon(t)}{\theta t}\right)\right) \\ &\le \theta t \ln\left(\sum_{i\in I_\varepsilon(x,t)} \exp\left(\frac{g_i(x)}{\theta t}\right)+\exp\left(-\frac{\varepsilon(t)}{\theta t}\right)\right).\end{aligned} \tag{24}$$

Then the right inequality can be obtained by Proposition 2.3 in [19].

(2) For any x, t with $g_{\max}(x) \ge -\varepsilon(t)$, let $g_{i^*}(x) = g_{\max}(x)$; then $\varphi(g_{i^*}(x),t) = 1$ by its definition, and hence

$$\begin{aligned}\widehat{g}(x,t) &= \theta t \ln\left(\sum_{i\in I_\varepsilon(x,t)} \varphi(g_i(x),t)\exp\left(\frac{g_i(x)}{\theta t}\right)\right. \\ &\qquad \left. + \exp\left(-\frac{\varepsilon(t)}{\theta t}\right)\right) \\ &\ge \theta t \ln\left(\varphi(g_{i^*}(x),t)\exp\left(\frac{g_{i^*}(x)}{\theta t}\right)\right) \\ &= \theta t \ln\left(\exp\left(\frac{g_{i^*}(x)}{\theta t}\right)\right) \\ &= g_{i^*}(x) \\ &= g_{\max}(x);\end{aligned} \tag{25}$$

then the left inequality is true. The right inequality can be obtained by item 1.

(3) It is trivial by the definition of $\widehat{g}(x,t)$. □

Proposition 10. *Let* $\Omega_\theta(t) = \{x \mid \widehat{g}(x,t) \le 0\}$ *and* $\Omega_\theta(t)^0 = \{x \mid \widehat{g}(x,t) < 0\}$; *one has*

(1) $\Omega_\theta(t) \subset \Omega$ *for* $t \in (0,1]$;

(2) *for any bounded and closed set* $\widehat{\Omega} \subset \Omega^0$, *there exists a* $\theta \in (0,1]$, *such that* $\widehat{\Omega} \subset \Omega_\theta(1)^0$;

(3) $\Omega_\theta(t) \to \Omega$ *as* $t \downarrow 0$.

Proof. (1) If $x \notin \Omega$, which is equivalent to $g_{\max}(x) > 0$, by Proposition 9(2),

$$0 < g_{\max}(x) \le \widehat{g}(x,t), \tag{26}$$

which means $x \notin \Omega_\theta(t)$; then we have $\Omega_\theta(t) \subset \Omega$.

(2) By the continuity of $g(x)$ and the fact that $\widehat{\Omega}$ is a bounded closed set, there exists a point $x^* \in \widehat{\Omega}$ such that $g_{\max}(x)$ reaches its maximum in $\widehat{\Omega}$, and $g_{\max}(x^*) < 0$. For any $\theta \in (0,1]$ satisfying $\theta \le -\max\{g_{\max}(x^*), -\varepsilon(1)\}/(2\ln(m+1))$, by Proposition 9(1), for any $x \in \widehat{\Omega}$,

$$\begin{aligned}\widehat{g}(x,1) &\le \max\{g_{\max}(x), -\varepsilon(1)\} + \theta\ln(m+1) \\ &\le \max\{g_{\max}(x^*), -\varepsilon(1)\} + \theta\ln(m+1) \\ &= \frac{1}{2}\max\{g_{\max}(x^*), -\varepsilon(1)\} \\ &< 0,\end{aligned} \tag{27}$$

which means $\widehat{\Omega} \subset \Omega_\theta(1)^0$.

(3) For any $x \in \Omega$, that is, $g_{\max}(x) \le 0$, by the right inequality of Proposition 9(1),

$$\begin{aligned}\lim_{t\downarrow 0}\widehat{g}(x,t) &\le \lim_{t\downarrow 0}\max\{g_{\max}(x), -\varepsilon(t)\} \\ &\le \max\{g_{\max}(x), -c_2\} \le 0,\end{aligned} \tag{28}$$

where the second inequality comes from $-\varepsilon(t) = -c_1 t - c_2 \le -c_2 < 0$ for $t \in [0,1]$, which means $\Omega \subset \lim_{t\downarrow 0}\Omega_\theta(t)$. Then, together with item 1,

$$\lim_{t\downarrow 0}\Omega_\theta(t) = \Omega. \tag{29}$$

□

Proposition 11. *There exists a* $\theta \in (0,1]$ *such that, for any* $t \in (0,1]$, $x \in \partial\Omega_\theta(t)$; *then* $\nabla_x\widehat{g}(x,t) \neq 0$.

Proof. If not, for any $\theta \in (0,1]$, there exist corresponding $t_\theta \in (0,1]$ and $x^\theta \in \partial\Omega_\theta(t_\theta)$ such that $\nabla_x\widehat{g}(x^\theta, t_\theta) = 0$, which means that there must exist three sequences $\{\theta_k\}_{k=1}^\infty \subset (0,1]$, $\{t_k\}_{k=1}^\infty \subset (0,1]$, and $\{x^k\}_{k=1}^\infty$ with $x^k \in \partial\Omega_{\theta_k}(t_k)$, such that $\theta_k \downarrow 0$, $t_k \to \bar{t}$, $x^k \to \bar{x} \in \partial\Omega$, and

$$\begin{aligned}&\nabla_x\widehat{g}(x^k, t_k) \\ &\quad = \sum_{i\in I_\varepsilon(x^k,t_k)} \left(\lambda_i(x^k,t_k) + \xi_i(x^k,t_k)\right)\nabla g_i(x^k) \longrightarrow 0\end{aligned} \tag{30}$$

as $k \to \infty$. Then, by Proposition 8, we know that $\lambda(x^k,t_k) \to \bar{\lambda}$, $\xi(x^k,t_k) \to 0$ as $k \to \infty$, and

$$\sum_{i\in\lim_{k\to\infty}I_\varepsilon(x^k,t_k)} \bar{\lambda}_i = \sum_{i\in I(\bar{x})}\bar{\lambda}_i = 1, \tag{31}$$

where the first equality comes from that $\bar{\lambda}_i = 0$ for $i \notin I(\bar{x})$. Therefore, we have $\sum_{i\in I(\bar{x})}\bar{\lambda}_i\nabla g_i(\bar{x}) = 0$ by taking limits on (30); this is a contradiction to Assumption 6. □

Proposition 12. *For any bounded closed set* $\widehat{\Omega} \subset \Omega^0$, *there exists a* $\theta \in (0,1]$ *such that*

$$\{x + \lambda\nabla_x\widehat{g}(x,1) \mid \lambda > 0\} \cap \widehat{\Omega} = \emptyset, \tag{32}$$

for any $x \in \partial\Omega_\theta(1)$.

Proof. If not, there exist a bounded closed set $\widehat{\Omega}$ and four sequences $\{\theta_k\}_{k=1}^{\infty} \subset (0,1]$, $\{x^k\}_{k=1}^{\infty}$ with $x^k \in \partial\Omega_{\theta_k}(1)$, $\{\widehat{x}^k\}_{k=1}^{\infty} \subset \widehat{\Omega}$, and $\{\lambda^k\}_{k=1}^{\infty} > 0$ such that $\theta_k \downarrow 0$, $x^k \to \overline{x} \in \partial\Omega$, $\widehat{x}^k \to \widehat{x} \in \widehat{\Omega}$, $\lambda_i(x^k,1) \to \lambda_i^* \geq 0$, $\xi_i(x^k,1) \to 0$ as $k \to \infty$, and

$$\widehat{x}^k = x^k + \lambda^k \sum_{i \in I_\varepsilon(x^k,1)} \left(\lambda_i\left(x^k,1\right) + \xi_i\left(x^k,1\right)\right) \nabla g_i\left(x^k\right). \tag{33}$$

By Proposition 8, we have

$$\sum_{i \in \lim_{k\to\infty} I_\varepsilon(x^k,1)} \lambda_i^* = \sum_{i \in I(\overline{x})} \lambda_i^* = 1, \tag{34}$$

where the first equality comes from that $\lambda_i^* = 0$ for $i \notin I(\overline{x})$. Then, we have

$$\sum_{i \in I_\varepsilon(x^k,1)} \left(\lambda_i\left(x^k,1\right) + \xi_i\left(x^k,1\right)\right) \nabla g_i\left(x^k\right) \longrightarrow \sum_{i \in I(\overline{x})} \lambda_i^* \nabla g_i(\overline{x}). \tag{35}$$

By Assumption 6 and using (34),

$$\sum_{i \in I(\overline{x})} \lambda_i^* \nabla g_i(\overline{x}) \neq 0. \tag{36}$$

Hence $\{\lambda^k\}_{k=1}^{\infty}$ is a bounded sequence, and there must exist a subsequence converging to $\overline{\lambda} > 0$; then we have by (33)

$$\widehat{x} = \overline{x} + \overline{\lambda} \sum_{i \in I(\overline{x})} \lambda_i^* \nabla g_i(\overline{x}), \tag{37}$$

which contradicts Assumption 7. □

3. The Flattened Aggregate Constraint Homotopy Method

In [19], the following aggregate constraint homotopy was introduced:

$$H(x,\lambda,t) = \begin{pmatrix} (1-t)\left(\nabla f(x) + \lambda \nabla_x \widetilde{g}_\theta(x,t)\right) + t\left(x - x^0\right) \\ \lambda \widetilde{g}_\theta(x,t) - t\lambda^0 \widetilde{g}_\theta\left(x^0,1\right) \end{pmatrix} = 0, \tag{38}$$

where $\widetilde{g}_\theta(x,t) = \theta t \ln \sum_{i=1}^{m} \exp(g_i(x)/\theta t)$ with $\theta \in (0,1]$, λ is the Lagrangian multiplier of the aggregate constraint function $\widetilde{g}_\theta(x,t)$, and (x^0,λ^0) is the starting point in $\widehat{\Omega} \times R_{++}$. Under the weak normal cone condition, which is similar to the normal cone condition, it was proved that the ACH determines a smooth interior path from a given interior point to a KKT point. Then, the predictor-corrector procedure can be applied to trace the homotopy path from $(x^0,\lambda^0,1)$ to $(x^*,\lambda^*,0)$, in which (x^*,λ^*) is a KKT point of (1).

3.1. The Flattened Aggregate Constraint Homotopy. Using the flattened aggregate constraint function $\widehat{g}(x,t)$ in (10), we construct the following flattened aggregate constraint homotopy:

$$H(x,\lambda,t) = \begin{pmatrix} (1-t)\left(\nabla f(x) + \lambda \nabla_x \widehat{g}(x,t)\right) + t\left(x - x^0\right) \\ \lambda \widehat{g}(x,t) - t\lambda^0 \widehat{g}\left(x^0,1\right) \end{pmatrix} = 0, \tag{39}$$

where λ is the Lagrangian multiplier of the flattened aggregate constraint and (x^0,λ^0) is the starting point and can be randomly chosen from $\widehat{\Omega} \times R_{++}$.

We give the main theorem on the existence and convergence of a smooth path from $(x^0,\lambda^0,1)$ to $(x^*,\lambda^*,0)$, in which (x^*,λ^*) is a solution of the KKT system of (1), and hence the global convergence of our proposal, namely, the FACH method, can be proven.

Theorem 13. *Suppose that Assumptions 5–7 hold and $\theta \in (0,1]$ satisfies Propositions 10–12 for the bounded closed set $\widehat{\Omega} \subset \Omega_\theta(1)^0 \subset \Omega^0$; then for almost all starting points $w^0 = (x^0,\lambda^0) \in \widehat{\Omega} \times R_{++}$, the zero-points set $H^{-1}(0) = \{(x,\lambda,t) \mid H(x,\lambda,t) = 0\}$ defines a smooth path Γ_{w^0}, which starts at $(x^0,\lambda^0,1)$ and approaches the hyperplane $t = 0$. Furthermore, let $(x^*,\lambda^*,0)$ be any limit point of Γ_{w^0}, and $y^* = \lambda^* \lambda(x^*,0)$ ($\lambda(x^*,0)$ is a limit of $\lambda(x,t)$ as $x \to x^*, t \downarrow 0$), then x^* is a KKT point of* (1) *and y^* is the corresponding Lagrangian multiplier.*

Proof. Consider H as a map of the variable (w^0,x,λ,t), for any $(w^0,t) \in \widehat{\Omega} \times R_{++} \times (0,1]$, which means $x^0 \in \Omega_\theta(1)^0$; hence $\widehat{g}(x^0,1) < 0$, because

$$\frac{\partial H\left(w^0,x,\lambda,t\right)}{\partial w^0} = \begin{pmatrix} -tI & 0 \\ -t\lambda^0\left(\nabla_x \widehat{g}\left(x^0,1\right)\right)^T & -t\widehat{g}\left(x^0,1\right) \end{pmatrix} \tag{40}$$

is nonsingular and the Jacobi matrix of $H(w^0,x,\lambda,t)$ is of full row rank. Using the parameterized Sard theorem, Lemma 4, for almost all $w^0 \in \widehat{\Omega} \times R_{++}$, 0 is a regular value of $H(x,\lambda,t)$. From

$$H(x,\lambda,1) = \begin{pmatrix} x - x^0 \\ \lambda \widehat{g}(x,1) - \lambda^0 \widehat{g}\left(x^0,1\right) \end{pmatrix}, \tag{41}$$

we know that (x^0,λ^0) is the unique and simple solution of $H(x,\lambda,1) = 0$. Then, by the implicit function theorem, there exists a smooth curve $\Gamma_{w^0} \subset \Omega_\theta(t)^0 \times R_{++} \times (0,1]$ starting from $(x^0,\lambda^0,1)$ and being transversal to the hyperplane $t = 1$.

Since Γ_{w^0} can be extended in $\Omega_\theta(t)^0 \times R_{++} \times (0,1)$ until it converges to the boundary of $\Omega_\theta(t)^0 \times R_{++} \times (0,1)$, there must exist an extreme point $(\overline{x},\overline{\lambda},\overline{t}) \in \partial(\Omega_\theta(\overline{t}) \times R_+ \times [0,1])$. Let $(\overline{x},\overline{\lambda},\overline{t})$ be any extreme point of Γ_{w^0} other than $(x^0,\lambda^0,1)$; then only the following five cases are possible:

(1) $(\overline{x},\overline{\lambda},\overline{t}) \in \Omega \times R_+ \times \{0\}$, $\overline{\lambda} < +\infty$;

(2) $(\bar{x},\bar{\lambda},\bar{t}) \in \Omega_\theta(1) \times R_+ \times \{1\}, \bar{\lambda} < +\infty$;

(3) $(\bar{x},\bar{\lambda},\bar{t}) \in \partial\Omega_\theta(\bar{t}) \times R_{++} \times (0,1), \bar{\lambda} < +\infty$;

(4) $(\bar{x},\bar{\lambda},\bar{t}) \in \Omega_\theta(\bar{t}) \times \{0\} \times (0,1)$;

(5) $(\bar{x},\bar{\lambda},\bar{t}) \in \Omega_\theta(\bar{t}) \times \{+\infty\} \times [0,1]$.

Case (2) means that $H(x,\lambda,1) = 0$ has another solution $(\bar{x},\bar{\lambda})$ except (x^0,λ^0), or (x^0,λ^0) is a double solution of $H(x,\lambda,1) = 0$, which contradicts the fact that (x^0,λ^0) is the unique and simple solution of $H(x,\lambda,1) = 0$. Case (3) means $\widehat{g}(\bar{x},\bar{t}) = 0, 0 < \bar{\lambda} < \infty$, and $0 < \bar{t} < 1$; hence

$$\bar{\lambda}\widehat{g}(\bar{x},\bar{t}) - \bar{t}\lambda^0\widehat{g}(x^0,1) = -\bar{t}\lambda^0\widehat{g}(x^0,1) < 0, \tag{42}$$

which contradicts the last equation of (39). Case (4) means $\widehat{g}(\bar{x},\bar{t}) \le 0, \bar{\lambda} = 0$, and $0 < \bar{t} < 1$; hence

$$\bar{\lambda}\widehat{g}(\bar{x},\bar{t}) - \bar{t}\lambda^0\widehat{g}(x^0,1) = -\bar{t}\lambda^0\widehat{g}(x^0,1) < 0, \tag{43}$$

which contradicts the last equation of (39). Then, Cases (2), (3), and (4) are impossible.

Because $H(x,\lambda,1) = 0$ has only one and also unique solution (x^0,λ^0), Case (2) is impossible. By the continuity and the last equation of (39), we know that Cases (3) and (4) are impossible.

If Case (5) holds, there must exist a sequence $\{(x^k,\lambda^k,t_k)\}_{k=1}^{\infty}$ on Γ_{w^0} such that $x^k \to \bar{x}$, $\lambda^k \to \infty$, and $t_k \to \bar{t}$. By the last equation of (39), we have

$$\widehat{g}(x^k,t_k) = \frac{t_k\lambda^0\widehat{g}(x^0,1)}{\lambda^k} \longrightarrow 0 \tag{44}$$

as $k \to \infty$, which means $\bar{x} \in \partial\Omega_\theta(\bar{t})$. The following two cases may be possible.

(1) If $\bar{t} = 1$, let $\bar{\alpha}$ be any accumulation point of $\{(1-t_k)\lambda^k\}$, which is known to exist from (39) and $\lim_{k\to\infty}\nabla_x\widehat{g}(x^k,t_k) \ne 0$; then we have

$$x^0 = \bar{x} + \bar{\alpha}\nabla_x\widehat{g}(\bar{x},1). \tag{45}$$

If $\bar{\alpha} = 0$, then $\bar{x} = x^0$ contradicts $x^0 \in \Omega_\theta(1)^0$; else contradicts Proposition 12.

(2) If $\bar{t} < 1$, by (39), we have

$$(1-\bar{t})\lim_{k\longrightarrow\infty}\lambda^k\nabla_x\widehat{g}(x^k,t_k) = -(1-\bar{t})\nabla f(\bar{x}) - \bar{t}(\bar{x}-x^0), \tag{46}$$

the right-hand is finite; however, by $\lim_{k\to\infty}\nabla_x\widehat{g}(x^k,t_k) \ne 0$, we know that the left-hand is infinite; this is a contradiction.

As a conclusion, Case (1) is the only possible case. This implies that Γ_{w^0} must approach the hyperplane $t = 0$. Because $\{\lambda(x^k,t_k)\}_{k=1}^{\infty}$(defined in (17)), $\{x^k\}_{k=1}^{\infty}$, and $\{\lambda^k\}_{k=1}^{\infty}$ are bounded sequences, we know that $\{(x^k,\lambda^k,\lambda(x^k,t_k))\}_{k=1}^{\infty}$ has at least one accumulation point as $k \to \infty$. Let $(x^*,\lambda^*,\lambda(x^*,0))$ be any accumulation point; $y_i^* = \lambda^*\lambda_i(x^*,0)$. By (39) and the fact that $\xi(x,t) \to 0$ as $t \downarrow 0$, we have

$$\nabla f(x^*) + \sum_{i=1}^{m} y_i^*\nabla g_i(x^*) = 0. \tag{47}$$

By the fact that $\lim_{t\downarrow 0}\Omega_\theta(t) = \Omega$, we have $x^* \in \Omega$. If $x^* \in \Omega^0$, we know that

$$\begin{aligned}
&\lim_{x\longrightarrow x^*,t\downarrow 0}\widehat{g}(x,t)\\
&\quad\le \lim_{x\longrightarrow x^*,t\downarrow 0}\left(\max\{g_{\max}(x),-\varepsilon(t)\} + \theta t\ln\left(m_{I_\varepsilon}+1\right)\right)\\
&\quad= \max\{g_{\max}(x^*),\varepsilon(0)\}\\
&\quad= \max\{g_{\max}(x^*),-\alpha c_2\}\\
&\quad< 0,
\end{aligned} \tag{48}$$

where the first inequality comes from Proposition 9(1), the second equality comes from the continuity, and the third equality comes from the definition of $\varepsilon(t)$ in (11); hence $\lambda^* = 0$ by the last equation of (39); else $x^* \in \partial\Omega$, $\lim_{x\to x^*,t\downarrow 0}\widehat{g}(x,t) = g_{\max}(x^*) = 0$ by Proposition 9(2) and $\lambda_i(x^*,0) = 0$ for $i \notin I(x^*)$ by Proposition 8. Thus we have

$$y_i^* g_i(x^*) = 0, \quad i = 1,\ldots,m. \tag{49}$$

Summing up, (x^*,y^*) is a solution of the KKT system of (1), which means that x^* is a KKT point of (1) and y^* is the corresponding Lagrangian multiplier. □

3.2. The Modified Flattened Aggregate Constraint Homotopy. From Proposition 8(3), we know that $\xi(x,t) \to 0$ as $t \downarrow 0$ for any $x \in \Omega$ and, hence, we can use the following modified flattened aggregate constraint homotopy (MFACH) instead of the FACH:

$$\begin{aligned}
H(x,\lambda,t) &= \begin{pmatrix} (1-t)(\nabla f(x) + \lambda h(x,t)) + t(x-x^0) \\ \lambda\widehat{g}(x,t) - t\lambda^0\widehat{g}(x^0,1) \end{pmatrix}\\
&= 0,
\end{aligned} \tag{50}$$

where

$$h(x,t) = \sum_{i\in I_\varepsilon(x,t)} \lambda_i(x,t)\nabla g_i(x). \tag{51}$$

Remarks. (i) Since $\xi(x,t)$ is dropped in (51), the expressions of the homotopy map (50) and its Jacobian and hence the corresponding code become simpler. Moreover, the computation of H in (50) and its Jacobi matrix DH is a little cheaper than that for the FACH method.

(ii) To guarantee that $H(x,\lambda,t)$ in (39) be a C^2 map, $\varphi(z,t)$ must be C^3; hence, if $\bar{\varphi}(z,t)$ is chosen as a polynomial, the total degree should be seven; in contrast, for the homotopy map in (50), $\varphi(z,t)$ is only needed to be C^2. Then $\overline{\varphi}(z,t)$ should satisfy

$$\begin{aligned}
&\overline{\varphi}(-\alpha\varepsilon(t),t)=0, \qquad \frac{\partial\overline{\varphi}(-\alpha\varepsilon(t),t)}{\partial z}=0,\\
&\frac{\partial^2\overline{\varphi}(-\alpha\varepsilon(t),t)}{\partial z^2}=0,\\
&\overline{\varphi}(-\varepsilon(t),t)=1, \qquad \frac{\partial\overline{\varphi}(-\varepsilon(t),t)}{\partial z}=0,\\
&\frac{\partial^2\overline{\varphi}(-\varepsilon(t),t)}{\partial z^2}=0
\end{aligned} \tag{52}$$

and $\overline{\varphi}(z,t)$ can be chosen as

$$\begin{aligned}
\overline{\varphi}(z,t)=&\frac{6(z+\varepsilon(t))^5}{(\alpha\varepsilon(t)-\varepsilon(t))^5}+\frac{15(z+\varepsilon(t))^4}{(\alpha\varepsilon(t)-\varepsilon(t))^4}\\
&+\frac{10(z+\varepsilon(t))^3}{(\alpha\varepsilon(t)-\varepsilon(t))^3}+1.
\end{aligned} \tag{53}$$

(iii) The existence and convergence of the homotopy path defined by the MFACH can be proven in a similar way with that of the FACH.

4. The FACH-S-N Procedure and Numerical Results

4.1. The FACH-S-N Procedure. In this section, we give a numerical procedure, FACH-S-N procedure, to trace the flattened aggregate constraint homotopy path by secant predictor and Newton corrector steps. It consists of three main steps: the predictor step, the corrector step, and the end game.

The predictor step is an approximate step along the homotopy path: it uses a predictor direction d^k and a steplength h_k to get a predictor point. The first predictor direction uses the tangent direction, and others use the secant direction. The steplength is determined by several parameters. It is set to no more than 1, which can ensure that the predictor point is close to the homotopy path and hence stays in the convergence domain of Newton's method in the corrector step. If the angle of the predictor direction and the previous one β^k is greater than $\pi/4$, the steplength will be decreased to avoid using the opposite direction as the predictor direction. If the corrector criteria are satisfied with no more than four Newton iterations for three times in succession, the steplength will be increased or kept invariable. Otherwise, the steplength will be decreased.

Once a predictor point $(w^{(k,0)},t_{(k,0)})$ is calculated, one or more Newton iterations are used to bring the predictor point back to the homotopy path in the corrector step. The corrector points $(w^{(k,i)},t_{(k,i)})$, $i=1,\dots,5$, are calculated by $(w^{(k,i+1)},t_{(k,i+1)})=(w^{(k,i)},t_{(k,i)})+d^{(k,i+1)}$, $i=0,\dots,4$, where the step $d^{(k,i+1)}$ is the solution of an augmented system defined by the homotopy equation and the direction perpendicular to the predictor direction. The corrector step terminates when $d^{(k,i)}$ and $H(w^{(k,i)},t_{(k,i)})$ satisfy the tolerance criterions.

At each predictor step and corrector step, the feasibility of the predictor point (w^k,t_k) and the corrector point $(w^{(k,i)},t_{(k,i)})$ needs to be checked. If $t_k<t_c$ in the predictor step or $t_k<0$ in the corrector step, a damping step is used to get a new point $(w^{(k,0)},0)$. Then, if $w^{(k,0)}$ is feasible, the end game, a more efficient strategy than predictor-corrector steps when the homotopy parameter is close to 0, is invoked. Starting with $w^{(k,0)}$, Newton's method is used to solve

$$F_{t_c}(x,\lambda)=\begin{pmatrix}\nabla f(x)+\lambda h(x,t_c)\\ \lambda\widehat{g}(x,t_c)\end{pmatrix}=0, \tag{54}$$

where t_c is a small positive constant. For other situations, the steplength will be decreased to make new predictor-corrector steps (see Algorithm 1).

4.2. The Numerical Experiment. Although there exist so many test problems, such as the CUTEr test set [21], we cannot find a large collection of test problems with moderate variables and very many complicated nonlinear constraints. In this paper, six test problems are chosen to test the algorithm. Problem 4.1 is chosen from the CUTEr test set and it is used to illustrate a special situation; others are derived from the discretized semi-infinite programming problems. We also give two artificial test problems 4.2 and 4.3 and use three problems 4.4–4.6 in [22]. The numbers of variables in problem 4.2 and the number of constrains in problems 4.2–4.6 can be arbitrary. For each test problem, the gradients and Hessians of the objective and constraint functions are evaluated analytically.

The FACH-S-N procedure was implemented in MATLAB. To illustrate its efficiency, we also implemented the ACH method in [19] using a similar procedure. In addition, we downloaded the state-of-the-art solver KNITRO, which provides three algorithms for solving large-scale nonlinear programming problems, and we used the interior-point direct method with default parameters to compare with the FACH and MFACH methods. For any iterate point x^k, if $g_{\max}(x^k)<10^{-6}$, it was treated as a feasible point. The test results were obtained by running MATLAB R2008a on a desktop with Windows XP Professional operation system, Intel(R) Core(TM) i5-750 2.66 GHz processor, and 8 GB of memory. The default parameters were chosen as follows.

(i) Parameters for the flattened aggregate constraint function: $\theta=0.01$, $c_1=0.05$, $c_2=0.5*10^{-5}$, and $\alpha=2$.

(ii) Parameters in the end game section: $t_c=10^{-6}$ and $t_{\text{end}}=0.1$.

(iii) Step size parameters: $h_0=0.1$, $B_{\min}=[0.5,0.75]$, and $B_{\max}=[3,1.5]$.

(iv) Tracking tolerances: $H_{\text{tol}}=10^{-5}$ and $H_{\text{final}}=10^{-12}$.

(v) Initial Lagrangian multipliers: 1 for ACH, FACH, and MFACH methods.

```
Input Give $\theta \in (0,1]$, $c_1 > 0$, $c_2 > 0$, $\alpha > 1$, $t_{\text{end}}$ and $t_c$; starting point $w^0 \in \widehat{\Omega} \times R_{++}$; initial steplength $h_1$, steplength
    contraction factor $B_{\min}$, steplength expansion factor $B_{\max}$; tracking tolerances $H_{\text{tol}}$ and $H_{\text{final}}$ for correction. Set
    $t_0 = 1$, $k = 1$, IT $= 0$, $N_{\text{good}} = 2$, $\beta^1 = 0$.
Step 1. The predictor step.
    Step 1.1. Compute the predictor direction.
        If $k = 1$, compute $d$ by solving
            $$\begin{pmatrix} H'\left(x^0,\lambda^0,1\right) \\ \left(d^0\right)^T \end{pmatrix} d = -d^0,$$
        where $d^0 = (0,\ldots,0,-1) \in R^{n+2}$, set the predictor direction $d^1 = d/\|d\|$;
        else, compute the predictor direction $d^k$, the angle $\beta^k$ between $d^k$ and $d^{k-1}$:
            $$d^k = \frac{\left(w^{k-1},t_{k-1}\right) - \left(w^{k-2},t_{k-2}\right)}{\left\|\left(w^{k-1},t_{k-1}\right) - \left(w^{k-2},t_{k-2}\right)\right\|}, \qquad \beta^k = \arccos\left(\left(d^k\right)^T d^{k-1}\right).$$
    Step 1.2. Adjust the steplength.
        Adjust the steplength $h_k$ as follows:
            if the predictor point is infeasible, set $h_k = B_{\min}(1)h_k$, $N_{\text{good}} = 0$;
            else if the corrector step fails or $\beta^k > \pi/4$, set $h_k = B_{\min}(1)h_{k-1}$, $N_{\text{good}} = 0$;
            else if $i = 5$, set $h_k = B_{\min}(2)h_{k-1}$, $N_{\text{good}} = 0$;
            else if $i = 4$, set $h_k = h_{k-1}$, $N_{\text{good}} = N_{\text{good}} + 1$;
            else if $i = 3$, set $N_{\text{good}} = N_{\text{good}} + 1$,
                if $N_{\text{good}} > 2$, set $h_k = \min\{1, B_{\max}(2)h_{k-1}\}$, else, set $h_k = h_{k-1}$;
            else, set $N_{\text{good}} = N_{\text{good}} + 1$,
                if $N_{\text{good}} > 2$, set $h_k = \min\{1, B_{\max}(1)h_{k-1}\}$, else, set $h_k = h_{k-1}$.
        If $h_k < 10^{-10}$, stop the algorithm with an error flag.
    Step 1.3. Compute the predictor point and check its feasibility.
        Compute the predictor point
            $$\left(w^{(k,0)}, t_{(k,0)}\right) = \left(w^{k-1}, t_{k-1}\right) + h_k d^k.$$
        If $w^{(k,0)} \notin \Omega_\theta(t_{(k,0)})^0 \times R_{++}$ or $t_{(k,0)} \ge 1$,
            the predictor point is infeasible, goto Step 1.2;
        else if $t_{(k,0)} \le t_{\text{end}}$,
            adjust the steplength $h_k$, compute the point $(w^{(k,0)}, 0)$, if $w^{(k,0)} \in \Omega \times R_+$, goto Step 3, else, the predictor
            point is infeasible, goto Step 1.2;
        else, goto Step 2.
Step 2. The corrector step.
    Set $i = 0$.
    Repeat
        If $i = 5$, set IT $=$ IT $+ 5$, $w^k = w^{k-1}$, $t_k = t_{k-1}$, $d^{k+1} = d^k$, $k = k + 1$, the corrector step fails, goto Step 1.2.
        Compute the Newton step $d^{(k,i+1)}$ by solving
            $$\begin{pmatrix} H'\left(w^{(k,i)}, t_{(k,i)}\right) \\ \left(d^k\right)^T \end{pmatrix} d^{(k,i+1)} = \begin{pmatrix} -H\left(w^{(k,i)}, t_{(k,i)}\right) \\ 0 \end{pmatrix},$$
        the corrector point
            $$\left(w^{(k,i+1)}, t_{(k,i+1)}\right) = \left(w^{(k,i)}, t_{(k,i)}\right) + d^{(k,i+1)};$$
        set $i = i + 1$.
        If $w^{(k,i)} \notin \Omega_\theta(t_{(k,i)})^0 \times R_{++}$ or $t_{(k,i)} \ge 1$,
            set IT $=$ IT $+ i$, $w^k = w^{k-1}$, $t_k = t_{k-1}$, $d^{k+1} = d^k$, $k = k + 1$, the corrector step fails, goto Step 1.2;
        else if $t_{(k,i)} < 0$,
            compute the point $(w^{(k,0)}, 0)$ by a damping Newton step, set IT $=$ IT $+ i$, if $w^{(k,0)} \in \Omega \times R_+$, goto Step 3,
            else, set $w^k = w^{k-1}$, $t_k = t_{k-1}$, $d^{k+1} = d^k$, $k = k + 1$, the corrector step fails, goto Step 1.2.
    Until $\|H(w^{(k,i)}, t_{(k,i)})\|_{\inf} \le H_{\text{tol}}$ and $\|d^{(k,i)}\| \le H_{\text{tol}}$.
    If $t_k < t_c$, return with $x^* = x^k$, stop the algorithm; else, set $H_{\text{tol}} = \min\{H_{\text{tol}}, t_k\}$, $k = k + 1$, goto Step 1.
Step 3. The end game.
    Set $l = 0$.
    Repeat
        If $l = 5$ or $\|d^{(k,l)}\| > \|d^{(k,l-1)}\|$ for $l > 1$, or $w^{(k,l)} \notin \Omega \times R_+$, set $t_{\text{end}} = 0.3 * t_{\text{end}}$, $w^k = w^{k-1}$, $t_k = t_{k-1}$,
        $d^{k+1} = d^k$, IT $=$ IT $+ l$, $k = k + 1$, the corrector step fails, goto Step 1.2.
        Compute the Newton step $d^{(k,l+1)}$ by solving $F'_{t_c}(w^{(k,l)})d^{(k,l+1)} = -F_{t_c}(w^{(k,l)})$, the corrector point
            $$w^{(k,l+1)} = w^{(k,l)} + d^{(k,l+1)};$$
```

Algorithm 1: Continued.

```
        set $l = l + 1$.
    Until $\|F_{t_c}(w^{(k,l)})\|_{\inf} \le H_{\text{final}}$ and $\|d^{(k,l)}\| \le H_{\text{final}}$.
    Set IT = IT + $l$, return with $x^* = x^{(k,l)}$, stop the algorithm.
```

Algorithm 1: The FACH-S-N procedure.

For each problem with different parameters, we list the value of objective function $f(x)$ and the maximal function of constraints $g_{\max}(x)$ at x^*, the number of iterations IT, the number of evaluations of gradients of individual constraint functions in the FACH and MFACH methods N_s (in contrast, N_s is $m * IT$ in the ACH method), and the running time in seconds $Time$. For problems that were not solved by the conservative setting, we also give the reason for failure. The notation "fail^1" indicates that the steplength in predictor step is smaller than 10^{-10}; it is generally due to poor conditioned Jacobian matrix. The notation "fail^2" means out of memory. The notation "fail^3" means no result in 5000 iterations or 3600 seconds.

Example 14 (see [21]). Consider

$$
\begin{aligned}
f(x) &= \sin(x_1 - 1 + 1.5\pi) \\
&\quad + \sum_{2\le i\le 1000} 100 \sin(-x_i + 1.5\pi + x_{i-1}^2), \\
g_i(x) &= \begin{cases} x_1 - \pi & i = 1; \\ x_{i-1}^2 - x_i - \pi & i = 2, \dots, 1000; \\ -x_1 - \pi & i = 1001; \\ -x_{i-1}^2 + x_i - \pi & i = 1002, \dots, 2000, \end{cases} \\
x^0 &= (1, \dots, 1) \in R^{1000}.
\end{aligned}
\tag{55}
$$

Remark 15. In this example, the starting point happens to be an unconstrained minimizer of the objective function and we found that the x-components of iterative points generated by the FACH and MFACH methods (not other homotopy methods) remain invariant. This is not an occasional phenomenon. In fact, if $(x^0, 0) \in \Omega^0 \times R_+^m$ is a solution of the KKT system, that is, x^0 is a stationary point of the objective function $f(x)$, when the parameters α, c_1, and c_2 in FACH and MFACH methods satisfy $g_{\max}(x^0) \le -\alpha(c_1 + c_2)$, the x-components of points on the homotopy path Γ_{w^0} will remain invariant, which can be derived from the fact that $(x^0, t\lambda^0\varepsilon(1)/\varepsilon(t), t)$ is the solution of (39) and (50) for $t \in (0, 1]$. Moreover, since

$$
H'(x^0, \lambda^0, 1) = \begin{pmatrix} I & 0 & 0 \\ 0 & \widehat{g}(x^0, 1) & * \end{pmatrix},
\tag{56}
$$

the x-component of d^1 is 0. By a similar discussion, the x-component of $d^{(1,i)}$ is 0, and hence $x^1 = x^{(1,i)} = x^0$. In this analogy, we know that $x^k = x^{(k,i)} = x^0$ for any k and i in algorithm FACH-S-N.

Table 1: Test results for Example 14.

Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
ACH	−99901.0000	−2.1416	35	—	85.5496
KNITRO	−99901.0000	−2.1416	0	—	2.8733
FACH	−99901.0000	−2.1416	4	0	8.3912
MFACH	−99901.0000	−2.1416	4	0	8.4150

Example 16. Consider

$$
\begin{aligned}
f(x) &= x_3^2 + x_4^2, \\
g_{i,j}(x) &= \frac{(t_i - x_1)^2}{x_3^2} + \frac{(t'_j - x_2)^2}{x_4^2} - 1, \\
t_i &= \frac{i}{(\sqrt{m} - 1)}, \quad i = 0, \dots, \sqrt{m} - 1, \\
t'_j &= \frac{j}{(\sqrt{m} - 1)}, \quad j = 0, \dots, \sqrt{m} - 1, \\
x^0 &= (0, 0, 100, 100) \in R^4.
\end{aligned}
\tag{57}
$$

Example 17. Consider

$$
\begin{aligned}
f(x) &= \frac{1}{n} \sum_{1\le k\le n} (x_k - 1)^2, \\
g_i(x) &= \prod_{1\le k\le n} \cos(t_i x_k) + t_i \sum_{1\le k\le n} x_k^3, \\
t_i &= \frac{0.5 + \pi i}{(m - 1)}, \quad i = 0, \dots, m - 1, \\
x^0 &= (-2, \dots, -2) \in R^n.
\end{aligned}
\tag{58}
$$

Example 18 (see [22]). Consider

$$
\begin{aligned}
f(x) &= x_1^2 + x_2^2 + x_3^2, \\
g_i(x) &= x_1 + x_2 \exp(x_3 t_i) + \exp(2t_i) - 2\sin(4t_i), \\
t_i &= \frac{i}{(m - 1)}, \quad i = 0, \dots, m - 1, \\
x^0 &= (-200, -200, 200) \in R^3.
\end{aligned}
\tag{59}
$$

TABLE 2: Test results for Example 16.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^2	ACH	1.0000	$-1.3863e-08$	394	—	0.2423
	KNITRO	1.0000	$-7.1233e-10$	18	—	0.2288
	FACH	1.0000	$-1.3863e-08$	541	721	0.3270
	MFACH	1.0000	$-1.3863e-08$	567	823	0.2692
10^4	ACH	1.0000	$-1.3863e-08$	397	—	1.9730
	KNITRO	1.0000	$-8.0005e-10$	20	—	0.7227
	FACH	1.0000	$-1.3863e-08$	541	721	0.4359
	MFACH	1.0000	$-1.3863e-08$	567	823	0.4676
10^6	ACH	1.0000	$-1.3863e-08$	404	—	249.3395
	KNITRO	1.0000	$-7.8830e-10$	13	—	2170.4301
	FACH	1.0000	$-1.3863e-08$	541	805	10.9486
	MFACH	1.0000	$-1.3863e-08$	567	907	11.2815

TABLE 3: (a) Test results for Example 17 with $n = 100$. (b) Test results for Example 17 with $n = 500$. (c) Test results for Example 17 with $n = 1000$. (d) Test results for Example 17 with $n = 2000$.

(a)

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^2	ACH	1.4918	$-1.5282e-03$	243	—	36.9898
	KNITRO	0.2603	$-1.9708e-06$	32	—	4.1262
	FACH	1.4915	$-6.6613e-16$	156	96	0.4067
	MFACH	1.4915	$6.1062e-15$	144	85	0.3186
10^3	ACH	1.4918	$-1.5282e-03$	243	—	325.6083
	KNITRO	0.2603	$-2.7042e-05$	19	—	25.1720
	FACH	1.4915	$-6.6613e-16$	156	96	0.6852
	MFACH	1.4915	$6.1062e-15$	144	85	0.5342
10^4	ACH	1.4918	$-1.5282e-03$	243	—	3218.5422
	KNITRO	0.2603	$-2.5434e-04$	18	—	271.1651
	FACH	1.4915	$-6.6614e-16$	156	96	5.1941
	MFACH	1.4915	$6.1062e-15$	144	86	4.0120

(b)

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^2	ACH	1.2524	$-1.0693e-02$	405	—	1371.0198
	KNITRO	0.1417	$-2.0927e-05$	31	—	95.1782
	FACH	1.2510	$1.2490e-14$	159	45	3.9951
	MFACH	1.2510	$1.2490e-14$	139	25	2.9676
10^3	ACH	—	—	—	—	fail[3]
	KNITRO	—	—	—	—	fail[3]
	FACH	1.2510	$1.2490e-14$	159	45	6.5173
	MFACH	1.2510	$1.2490e-14$	139	25	4.6898
10^4	ACH	—	—	—	—	fail[2]
	KNITRO	—	—	—	—	fail[2]
	FACH	1.2510	$1.2490e-14$	159	45	31.7107
	MFACH	1.2510	$1.2490e-14$	139	25	25.1842

(c)

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^2	ACH	—	—	—	—	fail[3]
	KNITRO	0.1099	$-5.6905e-05$	27	—	359.4507
	FACH	1.1880	$2.3370e-14$	197	46	20.8919
	MFACH	1.1880	$2.4258e-14$	197	46	21.2641

(c) Continued.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^3	ACH	—	—	—	—	fail[2]
	KNITRO	—	—	—	—	fail[2]
	FACH	1.1880	$2.3370e-14$	197	46	28.8674
	MFACH	1.1880	$2.4258e-14$	197	46	27.5140
10^4	ACH	—	—	—	—	fail[2]
	KNITRO	—	—	—	—	fail[2]
	FACH	1.1880	$2.3370e-14$	197	46	101.3199
	MFACH	1.1880	$2.4258e-14$	197	46	99.1354

(d)

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^2	ACH	—	—	—	—	fail[3]
	KNITRO	—	—	—	—	fail[3]
	FACH	1.1406	$-6.2839e-14$	260	47	126.6290
	MFACH	1.1406	$4.0967e-14$	258	46	125.7973
10^3	ACH	—	—	—	—	fail[2]
	KNITRO	—	—	—	—	fail[2]
	FACH	1.1406	$-6.2839e-14$	260	47	146.6079
	MFACH	1.1406	$4.0967e-14$	258	46	145.7293
10^4	ACH	—	—	—	—	fail[2]
	KNITRO	—	—	—	—	fail[2]
	FACH	1.1406	$-6.2839e-14$	260	47	368.5331
	MFACH	1.1406	$4.0967e-14$	258	46	369.3701

Table 4: Test results for Example 18.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^2	ACH	5.3347	$-4.4409e-16$	1605	—	0.7546
	KNITRO	$1.2000e+05$	$-3.9900e+02$	2	—	0.0141
	FACH	5.3347	$1.3323e-15$	815	288	0.2977
	MFACH	5.3347	$1.3323e-15$	820	293	0.3004
10^4	ACH	5.3347	$-4.4409e-16$	1605	—	7.0581
	KNITRO	$1.2000e+05$	$-3.9900e+02$	2	—	0.3163
	FACH	5.3347	$1.3323e-15$	815	288	1.3127
	MFACH	5.3347	$1.3323e-15$	820	294	1.3185
10^6	ACH	5.3347	$1.3323e-15$	1606	—	575.4663
	KNITRO	$1.2000e+05$	$-3.9900e+02$	2	—	3243.2201
	FACH	5.3347	$-9.2480e-08$	960	6444	42.7664
	MFACH	5.3347	$-1.1278e-07$	931	7094	41.7933

Example 19 (see [22]). Consider

$$\begin{aligned}
f(x) &= \left(x_1 - 2x_2 + 5x_2^2 - x_2^3 - 13\right)^2 \\
&\quad + \left(x_1 - 14x_2 + x_2^2 + x_2^3 - 29\right)^2, \\
g_i(x) &= x_1^2 + 2x_2 t_i^2 + \exp\left(x_1 + x_2\right) - \exp\left(t_i\right), \\
t_i &= \frac{i}{(m-1)}, \quad i = 0, \ldots, m-1, \\
x^0 &= (0, -45) \in R^2.
\end{aligned} \tag{60}$$

Example 20 (see [22]). Consider

$$\begin{aligned}
f(x) &= \frac{x_1^2}{3} + \frac{x_1}{2} + x_2^2, \\
g_i(x) &= \left(1 - x_1^2 t_i^2\right)^2 - x_1 t_i^2 - x_2^2 + x_2, \\
t_i &= \frac{i}{(m-1)}, \quad i = 0, \ldots, m-1, \\
x^0 &= (-1, 100) \in R^2.
\end{aligned} \tag{61}$$

To explain the numerical efficiency, we make the following remarks by numerical results in Tables 1, 2, 3, 4, 5 and 6.

Table 5: Test results for Example 19.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^2	ACH	97.1589	0.0000	213	—	0.0964
	KNITRO	97.1592	−7.5266 − 05	68	—	0.5162
	FACH	97.1589	0.0000	170	191	0.0795
	MFACH	97.1589	0.0000	160	183	0.0608
10^4	ACH	97.1589	0.0000	227	—	0.5310
	KNITRO	102.2095	−6.7465 − 01	263	—	15.4868
	FACH	97.1589	0.0000	249	10926	0.2846
	MFACH	97.1589	0.0000	251	11321	0.2783
10^6	ACH	97.1589	0.0000	229	—	50.8763
	KNITRO	188.9496	−8.3841e − 01	17	—	2206.4660
	FACH	97.1589	0.0000	334	427520	9.5724
	MFACH	97.1589	0.0000	253	486506	6.8910

Table 6: Test results for Example 20.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	IT	N_s	Time
10^2	ACH	2.4305	0.0000	342	—	0.1749
	KNITRO	2.4305	−1.3822e − 10	21	—	0.0291
	FACH	2.4305	2.2204e − 16	348	387	0.1368
	MFACH	2.4305	2.2204e − 16	341	374	0.1276
10^4	ACH	2.4305	−6.6632e − 09	344	—	0.9804
	KNITRO	2.4305	−1.6180e − 10	38	—	1.0329
	FACH	2.4305	−6.6632e − 09	296	14710	0.2610
	MFACH	2.4305	−6.6632e − 09	358	23559	0.2802
10^6	ACH	2.4305	−4.9783e − 08	344	—	94.3943
	KNITRO	2.6182	−1.2609e − 04	61	—	2372.2093
	FACH	—	—	—	—	fail[1]
	MFACH	2.4305	−4.9021e − 08	516	2231308	16.3756

(i) If $g_{\max}(x) \leq -\alpha\varepsilon(t)$ for any $(x, \lambda, t) \in \Gamma_{w^0}$, the FACH and MFACH methods do not need to calculate the gradient and Hessian of any constraint functions.

(ii) For problems whose gradients and Hessians of constraint functions are expensive to evaluate, the performance of the FACH and MFACH methods is much better than the ACH method based on similar numerical tracing procedures.

(iii) Compared to the interior-point direct algorithm of the state-of-the-art solver KNITRO, the test results show that the FACH and MFACH methods perform worse when m is small but much better when m is large for most problems. In addition, we can see that their time cost increases more slowly than KNITRO solver as m increases.

(iv) The function $\varphi(z, t)$ is important for the flattened aggregate constraint function. Theoretically, the parameters c_1, c_2, and α in the FACH and MFACH methods can be chosen freely. However, they do matter in the practical efficiency of the FACH and MFACH methods and should be suitably chosen. If these parameters are too large, then too many gradients and Hessians of individual constraint functions need to be evaluated and, hence, cause low efficiency. On the other hand, if they are too small, the Hessian of the flattened aggregated constraint function (10) may become ill-conditioned. In our numerical tests, we fixed $c_1 = 0.05$, $c_2 = 0.5 * 10^{-5}$, and $\alpha = 2$. In addition, the function $\overline{\varphi}(z, t)$ can be defined in many ways, and preliminary numerical experiments show that the algorithms with different functions $\overline{\varphi}(z, t)$ have similar efficiencies.

(v) In algorithm FACH-S-N, we gave only a simple implementation of the ACH, FACH, and MFACH methods. To improve implementation of the FACH and MFACH methods, a lot of work needs to be done on all processes of numerical path tracing, say, schemes of predictor and corrector, steplength updating, linear system solving, and end game. Other practical strategies in the literature for large-scale nonlinear programming problems (e.g., [9, 23–26]) are also very important for improving the efficiency.

5. Conclusions

By introducing a flattened aggregate constraint function, a flattened aggregate constraint homotopy method is proposed for nonlinear programming problems with few variables and many nonlinear constraints, and its global convergence is proven. By greatly reducing the computation of gradients and Hessians of constraint functions in each iteration, the proposed method is very competitive for nonlinear programming problems with a large number of complicated constraint functions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The research was supported by the National Natural Science Foundation of China (11171051, 91230103).

References

[1] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2001.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[3] C. B. Garcia and W. I. Zangwill, *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice Hall, Englewood Cliffs, NJ, USA, 1981.

[4] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, vol. 13 of *SIAM Studies in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 1994.

[5] T. Terlaky, *Interior Point Methods of Mathematical Programming*, vol. 5 of *Applied Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[6] Y. Y. Ye, *Interior Point Algorithms*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, NY, USA, 1997.

[7] A. Forsgren and P. E. Gill, "Primal-dual interior methods for nonconvex nonlinear programming," *SIAM Journal on Optimization*, vol. 8, no. 4, pp. 1132–1152, 1998.

[8] D. M. Gay, M. L. Overton, and M. H. Wright, "A primal-dual interior method for nonconvex nonlinear programming," in *Advances in Nonlinear Programming*, vol. 14 of *Applied Optimization*, pp. 31–56, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[9] R. J. Vanderbei and D. F. Shanno, "An interior-point algorithm for nonconvex nonlinear programming," *Computational Optimization and Applications*, vol. 13, no. 1–3, pp. 231–252, 1999.

[10] G. Feng, Z. Lin, and B. Yu, "Existence of an interior pathway to a Karush-Kuhn-Tucker point of a nonconvex programming problem," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 32, no. 6, pp. 761–768, 1998.

[11] G. C. Feng and B. Yu, "Combined homotopy interior point method for nonlinear programming problems," in *Advances in Numerical Mathematics*, vol. 14 of *Lecture Notes in Numerical and Applied Analysis*, pp. 9–16, 1995.

[12] L. T. Watson, "Theory of globally convergent probability-one homotopies for nonlinear programming," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 761–780, 2000/01.

[13] Q. H. Liu, B. Yu, and G. C. Feng, "An interior point path-following method for nonconvex programming with quasi normal cone condition," *Advances in Mathematics*, vol. 19, no. 4, pp. 281–282, 2000.

[14] B. Yu, Q. H. Liu, and G. C. Feng, "A combined homotopy interior point method for nonconvex programming with pseudo cone condition," *Northeastern Mathematical Journal*, vol. 16, no. 4, pp. 383–386, 2000.

[15] Y. F. Shang, *Constraint shifting combined homotopy method for nonlinear programming, equilibrium programming and variational inequalities [Ph.D. thesis]*, Jilin University, 2006.

[16] B. Yu and Y. F. Shang, "Boundary moving combined homotopy method for nonconvex nonlinear programming," *Journal of Mathematical Research and Exposition*, vol. 26, no. 4, pp. 831–834, 2006.

[17] X. S. Li, "An aggregate function method for nonlinear programming," *Science in China A: Mathematics, Physics, Astronomy*, vol. 34, no. 12, pp. 1467–1473, 1991.

[18] B. W. Kort and D. P. Bertsekas, "A new penalty function algorithm for constrained minimization," in *Proceedings of the 1972 IEEE Conference on Decision and Control*, New Orleans, La, USA, 1972.

[19] B. Yu, G. C. Feng, and S. L. Zhang, "The aggregate constraint homotopy method for nonconvex nonlinear programming," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 45, no. 7, pp. 839–847, 2001.

[20] E. L. Allgower and K. Georg, *Introduction to Numerical Continuation Methods*, vol. 45 of *Classics in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2003.

[21] N. I. M. Gould, D. Orban, and P. L. Toint, "CUTEr (and sifdec), a constrained and unconstrained testing environment, revisited," Tech. Rep. TR/PA/01/04, CERFACS, Toulouse, France, 2001.

[22] D. H. Li, L. Qi, J. Tam, and S. Y. Wu, "A smoothing Newton method for semi-infinite programming," *Journal of Global Optimization*, vol. 30, no. 2-3, pp. 169–194, 2004.

[23] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[24] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, vol. 17 of *Springer Series in Computational Mathematics*, Springer, New York, NY, USA, 1992.

[25] R. Fletcher and S. Leyffer, "Nonlinear programming without a penalty function," *Mathematical Programming A*, vol. 91, no. 2, pp. 239–269, 2002.

[26] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: an SQP algorithm for large-scale constrained optimization," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 979–1006, 2002.