Optimization, 2016
Vol. 65, No. 4, 729–749, http://dx.doi.org/10.1080/02331934.2015.1064122

Taylor & Francis
Taylor & Francis Group

A spline smoothing homotopy method for nonconvex nonlinear programming

Li Dong[a b], Bo Yu[a*] and Guohui Zhao[a]

[a] *School of Mathematical Sciences, Dalian University of Technology, Dalian, P.R. China;* [b] *College of Science, Dalian Nationalities University, Dalian, P.R. China*

(*Received 26 February 2014; accepted 5 June 2015*)

Homotopy methods are globally convergent under weak conditions and robust; however, the efficiency of a homotopy method is closely related with the construction of the homotopy map and the path tracing algorithm. Different homotopies may behave very different in performance even though they are all theoretically convergent. In this paper, a spline smoothing homotopy method for nonconvex nonlinear programming is developed using cubic spline to smooth the max function of the constraints of nonlinear programming. Some properties of spline smoothing function are discussed and the global convergence of spline smoothing homotopy under the weak normal cone condition is proven. The spline smoothing technique uses a smooth constraint instead of m constraints and acts also as an active set technique. So the spline smoothing homotopy method is more efficient than previous homotopy methods like combined homotopy interior point method, aggregate constraint homotopy method and other probability one homotopy methods. Numerical tests with the comparisons to some other methods show that the new method is very efficient for nonlinear programming with large number of complicated constraints.

Keywords: nonlinear programming; spline function; homotopy method; interior-point method

AMS Subject Classifications: 90C30; 49M37

1. Introduction

In this paper, we consider the following nonlinear programming problem

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

where $f : R^n \to R$ and $g : R^n \to R^m$. Let $\Omega = \{x \in R^n | g_i(x) \le 0, i = 1, \ldots, m\}$, $\Omega^0 = \{x \in R^n | g_i(x) < 0, i = 1, \ldots, m\}$, $\partial\Omega = \Omega \backslash \Omega^0$ and for any $x \in \Omega$, $B(x) = \{j \in \{1, 2, \ldots, m\} | g_j(x) = 0\}$.

To our knowledge, if x^* is a solution of (1) and the Abadie constraint qualification holds at x^*, then there exists $y^* \in R^m$, such that (x^*, y^*) is a solution of the following KKT system of (1)

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

$$
\begin{aligned}
&\nabla f(x) + \sum_{i=1}^{m} y_i \nabla g_i(x) = 0, \\
&y_i g_i(x) = 0, \quad y_i \geq 0, g_i(x) \leq 0, i = 1, \ldots, m.
\end{aligned} \tag{2}
$$

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

For convex programming, the global convergence of the central path following methods was proved under assumptions that the logarithmic barrier function was strictly convex and the solution set was nonempty and bounded in [1–4]. A homotopy method for nonconvex nonlinear programming, which was called combined homotopy interior point method (CHIP) proposed by Feng et al. [5], Feng and Yu [6], Lin et al. [7]. The global convergence under the normal cone condition was proven. In [8,9], Yu et al. presented two modified CHIP methods. These methods generalized the normal cone condition to so-called quasi normal cone condition and pseudo cone condition, respectively. Based on the CHIP method, in [10], Shang and Yu proposed a constraint shifting combined homotopy method (CSCH), which could choose the initial point outside the feasible region, it could solve some problems that did not satisfy the normal cone condition. The CSCH is easier to be constructed than the modified CHIP.

In [11], Yu et al. constructed an aggregate constraint homotopy (ACH) using the so-called aggregate function which was introduced in [12], the global convergence under the weak normal cone condition was proven. The dimension of the linear systems arising in the process of numerically tracing the homotopy path determined by the CHIP and CSCH methods was $n + m + 1$ (where n is the number of variables, while m is the number of constraints). In the situation of ACH method, the dimension became $n + 2$, this property made this class method very efficient when m was very large.

In this paper, we present a new homotopy method called spline smoothing homotopy (SSH) method for nonlinear programming (1) using cubic spline which was introduced in [13] to smoothly approximate the min (or max) function. The smooth spline approximation of the max function of the constraints involves only few constraints, so it acts also as an active set technique, so it can improve the efficiency of the homotopy method. The approximation properties, the $C^{2,1}$ smoothness of smooth spline and the formulas of computing its gradient and Hessian are given. These properties are necessary to make the smooth spline to keep some conditions on constraints, which are necessary for proving the convergence of the homotopy method. Under the weak normal cone condition, we prove that SSH method determines a smooth interior path from a given interior point to a KKT point of the nonlinear programming (1). For the sake of using a cubic spline function and not a quartic spline, a parameterized version of the Sard's theorem with $C^{r,1}$ smoothness hypothesis given in [14] is used.

The rest of this paper is organized as follows. In Section 2, we give parameterized Sard theorem with $C^{r,1}$ smoothness. In Section 3, we give some properties of cubic spline to smooth max functions and the formulas of computing its gradient and Hessian. In Section 4, we give the spline smoothing homotopy and prove some propositions and main theorem on the existence of a smooth path from a given interior point to a KKT point. In Section 5, a procedure for tracking the homotopy path is listed and several numerical examples with some remarks on the numerical results are given.

2. Parameterized Sard theorem with $C^{r,1}$ smoothness

To develop our main result, we need the following definitions, lemmas and theorems which are from differential topology.

Definition 1 Let $U \subset R^n$ be an open set, and $f : U \to R^p$ be a smooth mapping. We say $y \in R^p$ is a regular value for f, if

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

LEMMA 2.1 (Lemmas 5–27, [15]) *If 0 is a regular value of the mapping f_a, then $f_a^{-1}(0)$ consists of some smooth manifolds.*

LEMMA 2.2 (Theorems 5–30, [15]) *A one-dimensional smooth manifold is diffeomorphic to a unit circle or a unit interval.*

Definition 2 A map $f \in C^r(R^n, R^m)$ is said to belong to $C^{r,1}$ if $D^r f$ is locally Lipschitz on R^n.

The following parameterized Sard theorem is commonly used for proving the regularity of a homotopy.

THEOREM 2.3 (Parameterized Sard Theorem [16]) *Let $U \subset R^m$ and $V \subset R^n$ be two open sets, and $f : U \times V \to R^k$ be an C^r differentiable map with $r > \max\{0, m-k\}$. If $0 \in R^k$ is a regular value of f, then for almost all $a \in V$, 0 is a regular value of $f_a = f(a, \cdot)$.*

In this paper, in order to use a spline function with degree as low as possible, namely S_3^2, in SSH, we use the following parameterized Sard theorem with $C^{r,1}$ smoothness.

THEOREM 2.4 *Let $U \subset R^m$ and $V \subset R^n$ be two open sets, and $f : U \times V \to R^k$ be an $C^{r,1}$ differentiable map with $m > k$ and $r \geq \max\{1, m-k\}$. If $0 \in R^k$ is a regular value of f, then for almost all $a \in V$, 0 is a regular value of $f_a = f(a, \cdot)$.*

Theorem 2.4 can be proven similarly to the parameterized Sard theorem in [16] using the following theorem in [14].

THEOREM 2.5 (Theorem 1, [14]) *Let n, m be positive integers with $n > m$ and $r = n - m$. If $f \in C^{r,1}(R^n, R^m)$, then the set of critical values of f has m-measure zeros.*

3. Cubic spline which uniformly approximates max functions

We formulated Problem (1) equivalently as

$$\begin{aligned} &\min \ f(x), \\ &\text{s.t.} \ \ g_{\max}(x) \leq 0, \end{aligned} \tag{3}$$

with a single but nonsmooth constraint $g_{\max}(x) = \max_{1\le i\le m}\{g_i(x)\}$. We consider to approximate smoothly $g_{\max}(x)$ by the cubic spline introduced in [13] and will use the spline smooth approximation $\hat{g}(x,t)$ to construct a homotopy, called spline smoothing homotopy, to solve (1).

In this section, we will introduce the definition of $\hat{g}(x,t)$, prove its approximation properties, the $C^{2,1}$ smoothness and give the formulas of its gradient and Hessian. In the next section, we will prove that $\hat{g}(x,t)$ can keep some conditions on $g(x)$, which are necessary for proving the convergence of the homotopy method.

Let us first recall the formulation of multivariate spline. Let D be a polyhedral domain of R^n which is partitioned with irreducible algebraic surfaces into cells $\triangle = \{\triangle_i | i = 1,\dots,N\}$. A function $s(x)$ defined on D is called a k-spline function with rth order smoothness, expressed for short as $s(x) \in S_k^r(D,\triangle)$, if $s(x) \in C^r(D)$ and $s(x)|_{\triangle_i} = p_i \in P_k$, where P_k is the set of all polynomial of degree k or less in n variables.

In [13], Zhao et al. constructed the homogenous Morgan-Scott partition of type two and a cubic spline $S_3^2(x) \in C^2$ to approximate $\min\{x_1, x_2, \dots, x_n\}$ uniformly, where $x = (x_1, x_2, \dots, x_n)^T \in R^n$. Now let us introduce it.

Let $M = (-\bar{m}, -\bar{m}, \dots, -\bar{m})^T \in R^n$, $P_i = (0,\dots,0,\bar{m},0,\dots,0)^T \in R^n$, where $\bar{m} > 0$, $1 \le i \le n$. Points $M, P_1, P_2, \dots, P_n$ define a n-simplex, namely V. Let $E_i = (\varepsilon,\dots,\varepsilon,0,\varepsilon,\dots,\varepsilon)^T \in R^n$,$1 \le i \le n$, $0 < \varepsilon < \bar{m}$. Let P_i' be the intersection points of line ME_i with the hyperplane passing $P_1, P_2, \dots, P_n$. Furthermore, join points P_i and P_j' $(j \in \Lambda = \{1,2,\dots,n\})$, P_j' and P_k' $(j,k \in \Lambda$ and $k \ne j)$ with straight lines. These lines intersect at the following points: $P'_{i_1,\dots,i_k} = \sum_{j=1}^{k} P'_{i_j}/k$ where $i_j \in \Lambda$. All these points can define a triangular partition of V, with cells $P_{i_n}\dots P_{i_{k+1}} M P'_{i_1} \dots P'_{i_1\dots i_k}$. It is obvious that V turns into R^n when $\bar{m} \to +\infty$ and the above triangular partition of V turns into a partition of R^n, namely homogenous Morgan-Scott partition of type 2 and is denoted by $\bar{\triangle}^2_{MS}$, with cells $\bar{\triangle}_{i_1\dots i_k}$, the limit of $P_{i_n}\dots P_{i_{k+1}} M P'_{i_1} \dots P'_{i_1\dots i_k}$, which is defined by

$$\begin{cases} x_{i_l} - x_{i_{l+1}} \le 0, & 1 \le l < k, \quad 1 \le k \le n, \\ \sum_{j=1}^{k-1} x_{i_j} - (k-1)x_{i_l} + \varepsilon \ge 0, & l = k, \quad 1 \le k \le n, \\ \sum_{j=1}^{k} x_{i_j} - kx_{i_l} + \varepsilon \le 0, & k+1 \le l \le n, \ 1 \le k \le n. \end{cases}$$

The C^2 cubic spline $S_3^2(x) \in S_3^2(R^n, \bar{\triangle}^2_{MS})$ which approximates uniformly $\min\{x_1, x_2, \dots, x_n\}$ (as $\varepsilon \to +0$) was defined in [13] as

$$S_3^2(x_1, x_2, \dots, x_n) = x_{i_1} + \sum_{l=1}^{k-1} C_l \left(\sum_{j=1}^{l} x_{i_j} - l x_{i_{l+1}} + \varepsilon \right)^3, \text{ for } x \in \bar{\triangle}_{i_1\dots i_k}(\varepsilon),$$

where $C_1 = -1/(6\varepsilon^2)$, $C_k/C_{k+1} = (k+2)/k$, $1 \le k \le n$.

Because $\max\{z_1, z_2, \dots, z_m\} = -\min\{-z_1, -z_2, \dots, -z_m\}$, we can approximate uniformly $\max\{z_1, z_2, \dots, z_m\}$ (as $\varepsilon \to +0$) by the following C^2 cubic spline function $s_3^2(z;\varepsilon) \in S_3^2(R^m, \triangle^2_{MS})$.

$$s_3^2(z_1, z_2, \ldots, z_m; \varepsilon) = z_{i_1} + \sum_{l=1}^{k-1} c_l \left(l z_{i_{l+1}} - \sum_{j=1}^{l} z_{i_j} + \varepsilon \right)^3, \text{ for } z \in \triangle_{i_1 \ldots i_k}(\varepsilon),$$

where $c_1 = 1/(6\varepsilon^2)$, $c_k/c_{k+1} = (k+2)/k$, $1 \le k \le m$. The cell $\triangle_{i_1 \ldots i_k}(\varepsilon)$ is the region defined by the following inequalities

$$\begin{cases} z_{i_l} - z_{i_{l+1}} \ge 0, \text{ when } 1 \le l < k, \\ (k-1) z_{i_k} - \sum\limits_{j=1}^{k-1} z_{i_j} + \varepsilon \ge 0, \\ k z_{i_l} - \sum\limits_{j=1}^{k} z_{i_j} + \varepsilon \le 0, \text{ when } k+1 \le l \le m. \end{cases}$$

The composite function $\hat{g}(x, t)$ approximates uniformly $\max\{g_1(x), g_2(x), \ldots, g_m(x)\}$ as $t \to +0$, where

$$\begin{array}{c} \hat{g}(x, t) = s_3^2(g_1(x), g_2(x), \ldots, g_m(x); t), \text{ for } x \text{ such that} \\ (g_1(x), g_2(x), \ldots, g_m(x)) \in \triangle_{i_1 \ldots i_k}(t). \end{array} \tag{4}$$

PROPOSITION 3.1 *For any $t > 0$, when $g(x) \in \triangle_{i_1 \ldots i_k}(t)$, the following results hold:*

(a) $g_{\max}(x) = g_{i_1}(x)$;

(b) $g_{\max}(x) \le \hat{g}(x, t) \le g_{\max}(x) + \dfrac{t}{3}\left(1 - \dfrac{1}{k}\right)$.

Proof

(a) According to the definition of the cell $\triangle_{i_1 \ldots i_k}(t)$, we can obtain that $g(x) \in \triangle_{i_1 \ldots i_k}(t)$ iff

$$\begin{cases} g_{i_1}(x) - g_{i_2}(x) \ge 0, \\ g_{i_2}(x) - g_{i_3}(x) \ge 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ g_{i_{k-1}}(x) - g_{i_k}(x) \ge 0, \\ (k-1) g_{i_k}(x) - \sum\limits_{j=1}^{k-1} g_{i_j}(x) + t \ge 0, \\ k g_{i_{k+1}}(x) - \sum\limits_{j=1}^{k} g_{i_j}(x) + t \le 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ k g_{i_m}(x) - \sum\limits_{j=1}^{k} g_{i_j}(x) + t \le 0. \end{cases} \tag{5}$$

From the first $(k-1)$ inequalities in (5), we know $g_{i_1}(x) \ge g_{i_j}(x)$ $(1 \le j \le k)$. From the kth inequality in (5), we have

$$\sum_{j=1}^{k-1} g_{i_j}(x) - (k-1) g_{i_k}(x) - t \le 0. \tag{6}$$

Add (6) on the $(k+1)$-th inequality in (5), we can obtain $g_{i_k}(x) \geq g_{i_{k+1}}(x)$. Similarly, we have $g_{i_k}(x) \geq g_{i_j}(x),\ j \in \{k+1, k+2, \ldots, m\}$. So, we have proven $g_{\max}(x) = g_{i_1}(x)$.

(b) Instead of the l-th to the $(k-1)$th inequality in (5), where $1 \leq l < k$, we use another set of inequalities:

$$\begin{cases} l g_{i_l}(x) - l g_{i_k}(x) \geq 0, \\ g_{i_{l+1}}(x) - g_{i_k}(x) \geq 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ g_{i_{k-1}}(x) - g_{i_k}(x) \geq 0. \end{cases} \tag{7}$$

By summing up all inequalities in (7) and the k-th inequality in (5) and from $g_{i_j}(x) \geq g_{i_l}(x),\ j \in \{1, 2, \ldots, (l-1)\}$, we can obtain $0 \leq (l-1)g_{i_l}(x) - \sum_{j=1}^{l-1} g_{i_j}(x) + t \leq t$. Because $c_l \geq 0$, we have $0 \leq c_l((l-1)g_{i_l}(x) - \sum_{j=1}^{l-1} g_{i_j}(x) + t)^3 \leq c_l t^3$. According to the definition of $\hat{g}(x,t)$ and conclusion (a), we have $g_{\max}(x) \leq \hat{g}(x,t) \leq g_{\max}(x) + \sum_{l=1}^{k-1} c_l t^3$. By $c_l/c_{l+1} = (l+2)/l$, we can obtain $c_l = (2/(l(l+1)))c_1 = 1/(3l(l+1)t^2)$. Thus $\sum_{l=1}^{k-1} c_l t^3 = (t/3)(1-(1/k))$. The proof is completed. □

Proposition 3.2 *For any $x \in \partial\Omega$, let $B(x) = \{i_1, \ldots, i_k\}$. Then, when t is sufficiently small, we have $g(x) \in \triangle_{i_1 \ldots i_k}(t)$.*

Proof It is easy to see that when t is sufficiently small, (5) holds and hence $g(x) \in \triangle_{i_1 \ldots i_k}(t)$. □

Proposition 3.3

(a) *The gradient of $\hat{g}$ about x, at (x,t) such that $g(x) \in \triangle_{i_1 \ldots i_k}(t)$, is*

$$\nabla_x \hat{g}(x,t) = \sum_{i=1}^{m} \lambda_i(x,t) \nabla g_i(x) = \sum_{j=1}^{k} \lambda_{i_j}(x,t) \nabla g_{i_j}(x), \tag{8}$$

where

$$\lambda_{i_j}(x,t) = \begin{cases} 1 - 3\sum_{l=1}^{k-1} c_l (h_l(x,t))^2, \ \text{for } j = 1, \\ 3(j-1)c_{j-1}(h_{j-1}(x,t))^2 - 3\sum_{l=j}^{k-1} c_l (h_l(x,t))^2, \text{for } 2 \leq j < k, \\ 3(k-1)c_{k-1}(h_{k-1}(x,t))^2, \ \text{for } j = k, \\ 0, \ \text{for } k < j \leq m, \end{cases} \tag{9}$$

and $h_l(x,t) = l g_{i_{l+1}}(x) - \sum_{r=1}^{l} g_{i_r}(x) + t$;

(b) *For any $x \in \Omega$ and $t > 0$, $\lambda_i(x,t) \geq 0$ $(1 \leq i \leq m)$ and $\sum_{i=1}^{m} \lambda_i(x,t) = 1$.*

Proof

(a) Through some calculations, we may obtain the conclusion.

(b) Suppose without loss of generality that $g(x) \in \triangle_{i_1 \dots i_k}(t)$, from the proof of the Proposition 3.1, we know $0 \le c_l(lg_{i_{l+1}}(x) - \sum_{j=1}^{l} g_{i_j}(x) + t)^2 \le c_l t^2$ and $c_l = 1/(3l(l+1)t^2)$. So we have

$$
\begin{aligned}
\lambda_{i_1}(x,t) &= 1 - 3\sum_{l=1}^{k-1} c_l(h_l(x,t))^2 \ge 1 - 3\sum_{l=1}^{k-1} c_l t^2 = 1 - 3t^2 c_1 \sum_{l=1}^{k-1} \frac{2}{(l+1)l} \\
&= 1 - 6t^2 c_1 \left(1 - \frac{1}{k}\right) = \frac{1}{k} > 0, \\
\lambda_{i_j}(x,t) &= 3(j-1)c_{j-1}(h_{j-1}(x,t))^2 - 3\sum_{l=j}^{k-1} c_l(h_l(x,t))^2 \\
&\ge 3(j-1)c_{j-1}(h_{j-1}(x,t))^2 - 3(h_{j-1}(x,t))^2 \sum_{l=j}^{k-1} c_l \\
&= 3(j-1)c_{j-1}(h_{j-1}(x,t))^2 - 3j(j-1)(h_{j-1}(x,t))^2 c_{j-1} \sum_{l=j}^{k-1} \frac{1}{(l+1)l} \\
&= 3(j-1)c_{j-1}(h_{j-1}(x,t))^2 - 3j(j-1)(h_{j-1}(x,t))^2 c_{j-1} \left(\frac{1}{j} - \frac{1}{k}\right) \\
&= 3j(j-1)c_{j-1}(h_{j-1}(x,t))^2 \frac{1}{k} \ge 0, \text{ for } 2 \le j < k, \\
\lambda_{i_k}(x,t) &= 3(k-1)c_{k-1}(h_{k-1}(x,t))^2 \ge 0.
\end{aligned}
$$

Together with $\lambda_{i_j}(x,t) = 0,$ when $k < j \le m$, we have $\lambda_i(x,t) \ge 0$ for $1 \le i \le m$. From (9), we have

$$
\begin{aligned}
\sum_{i=1}^{m} \lambda_i(x,t) = \sum_{j=1}^{k} \lambda_{i_j}(x,t) = 1 - 3\Bigg(&\sum_{l=1}^{k-1} c_l(h_l(x,t))^2 - c_1(h_1(x,t))^2 \\
&+ \sum_{l=2}^{k-1} c_l(h_l(x,t))^2 - 2c_2(h_2(x,t))^2 + \dots + \sum_{l=k-2}^{k-1} c_l(h_l(x,t))^2 \\
&-(k-2)c_{k-2}(h_{k-2}(x,t))^2 - (k-2)c_{k-1}(h_{k-1}(x,t))^2\Bigg) \\
= 1.&
\end{aligned}
$$

□

PROPOSITION 3.4 *If $g(x) \in C^{2,1}$, then for $g(x) \in \triangle_{i_1 \dots i_k}(t)$*

(a) $\nabla_x^2 \hat{g}(x,t) = \sum_{j=1}^{k} \lambda_{i_j}(x,t) \nabla^2 g_{i_j}(x) + \sum_{j=1}^{k} \left(\sum_{\breve{j}=1}^{k} \xi_{j,\breve{j}}(x,t) \nabla g_{i_{\breve{j}}}(x))(\nabla g_{i_j}(x) \right)^T,$ *where*

$$\xi_{1,\breve{j}}(x,t) = \begin{cases} 6\sum_{l=1}^{k-1} c_l(h_l(x,t)) \text{ when } \breve{j} = 1, \\ -6(\breve{j}-1)c_{\breve{j}-1}(h_{\breve{j}-1}(x,t)) + 6\sum_{l=\breve{j}}^{k-1} c_l(h_l(x,t)) \text{ when } 2 \le \breve{j} < k, \\ -6(k-1)c_{k-1}(h_{k-1}(x,t)) \text{ when } \breve{j} = k. \end{cases}$$

$$\xi_{j,\breve{j}}(x,t) = \begin{cases} -6(j-1)c_{j-1}(h_{j-1}(x,t)) + 6\sum_{l=j}^{k-1} c_l(h_l(x,t)) \text{ when } 1 \le \breve{j} < j, \\ -6(\breve{j}-1)c_{\breve{j}-1}(h_{\breve{j}-1}(x,t)) + 6\sum_{l=\breve{j}}^{k-1} c_l(h_l(x,t)) \text{ when } j < \breve{j} < k, \\ 6(\breve{j}-1)^2 c_{\breve{j}-1}(h_{\breve{j}-1}(x,t)) + 6\sum_{l=\breve{j}}^{k-1} c_l(h_l(x,t)) \text{ when } \breve{j} = j, \\ -6(k-1)c_{k-1}(h_{k-1}(x,t)) \text{ when } \breve{j} = k, \text{ for } 2 \le j < k, \end{cases}$$

$$\xi_{k,\breve{j}}(x,t) = \begin{cases} -6(k-1)c_{k-1}(h_{k-1}(x,t)) \text{ when } 1 \le \breve{j} < k, \\ 6(k-1)^2 c_{k-1}(h_{k-1}(x,t)) \text{ when } \breve{j} = k. \end{cases}$$

(b) *$\hat{g}(x,t)$ is an $C^{2,1}$ function for $t > 0$.*

Proof

(a) Through some calculations, we may obtain the conclusion.
(b) For $s_1(z;\varepsilon) \in S_1(R^n, \triangle^2_{MS})$, we know $s_1(z;\varepsilon)$ is Lipschitz. For the convenience of the reader, we prove it. The cell of $\triangle^2_{MS}$ is denoted by $\widetilde{\triangle}_i(\varepsilon)$ $(1 \le i \le \sum_{k=1}^n C_n^k k!)$ and let $s_{1_i}(z;\varepsilon) = a_{i,0}(\varepsilon) + \sum_{j=1}^n a_{i,j}(\varepsilon) z_j$ for $z \in \widetilde{\triangle}_i(\varepsilon)$. $\forall z', z'' \in R^n$, suppose without loss of generality, let $z' \in \widetilde{\triangle}_{i_1}(\varepsilon)$, $z'' \in \widetilde{\triangle}_{i_{s+1}}(\varepsilon)$, then we take the intersection points $q_{i_1}, q_{i_2}, \ldots, q_{i_s}$ of line $z'z''$ with $Q_{i_1}, Q_{i_2}, \ldots, Q_{i_s}$, where Q_{i_j} $(1 \le i_j < \sum_{k=1}^n C_n^k k!)$ is adjacent plan of two cells $\widetilde{\triangle}_{i_j}(\varepsilon)$ and $\widetilde{\triangle}_{i_{j+1}}(\varepsilon)$. So we have $s_{1_{i_j}}(q_{i_j}) = s_{1_{i_{j+1}}}(q_{i_j})$ $(1 \le j \le s)$ and $\|z' - q_{i_1}\| + \|q_{i_1} - q_{i_2}\| + \cdots + \|q_{i_{s-1}} - q_{i_s}\| + \|q_{i_s} - z''\| = \|z' - z''\|$. Then

$$\begin{aligned} \|s_{1_{i_1}}(z') - s_{1_{i_{s+1}}}(z'')\| &\le \|s_{1_{i_1}}(z') - s_{1_{i_1}}(q_{i_1})\| + \|s_{1_{i_2}}(q_{i_1}) - s_{1_{i_2}}(q_{i_2})\| + \cdots \\ &\quad + \|s_{1_{i_s}}(q_{i_{s-1}}) - s_{1_{i_s}}(q_{i_s})\| + \|s_{1_{i_{s+1}}}(q_{i_s}) - s_{1_{i_{s+1}}}(z'')\| \\ &\le L_{i_1}\|z' - q_{i_1}\| + L_{i_2}\|q_{i_1} - q_{i_2}\| \\ &\quad + \cdots + L_{i_s}\|q_{i_{s-1}} - q_{i_s}\| + L_{i_{s+1}}\|q_{i_s} - z''\| \\ &\le L(\|z' - q_{i_1}\| + \|q_{i_1} - q_{i_2}\| + \cdots \\ &\quad + \|q_{i_{s-1}} - q_{i_s}\| + \|q_{i_s} - z''\|) \\ &= L\|z' - z''\|, \end{aligned}$$

where $L = \max\{L_{i_1}, L_{i_2}, \ldots, L_{i_{s+1}}\}$.

From the definition of $\xi_{j,\breve{j}}(x,t)(1 \le j \le k)$, we know it is the composite function of $s_1(z;\varepsilon)$ and $g(x)$. Due to $g(x) \in C^{2,1}$ and $s_1(z;\varepsilon)$ is Lipschitz, we obtain $\nabla_x^2 \hat{g}(x,t)$ is locally Lipschitz on Ω. Then it is easy to prove $\hat{g}(x,t) \in C^{2,1}$ for $t > 0$. $\square$

4. Spline smoothing homotopy and homotopy path

In this paper, the following assumptions are made:

(A1) $f(x) \in C^{2,1}$ and $g(x) \in C^{2,1}$;
(A2) Ω^0 is nonempty, Ω is bounded;
(A3) For any $x \in \partial\Omega$, $\{\nabla g_i(x) | i \in B(x) = \{i | g_i(x) = 0\}\}$ are positive independent, i.e. $\sum_{i \in B(x)} \lambda_i \nabla g_i(x) = 0, \lambda_i \geq 0 \Rightarrow \lambda_i = 0$;
(A4) (The weak normal cone condition of Ω *w.r.t.* $\hat{\Omega}$.) There exists a closed subset $\hat{\Omega} \subset \Omega^0$ with nonempty interior $\hat{\Omega}^0$, such that for any given $x \in \partial\Omega$,

$$\{x + \sum_{i \in B(x)} \lambda_i \nabla g_i(x) : \lambda_i \geq 0, \sum_{i \in B(x)} \lambda_i > 0\} \cap \hat{\Omega} = \emptyset.$$

Let $\Omega_\theta(t) = \{x \in R^n | \hat{g}_\theta(x,t) \leq 0\}$, $\Omega_\theta(t)^0 = \{x \in R^n | \hat{g}_\theta(x,t) < 0\}$, where $\hat{g}_\theta(x,t) = \hat{g}(x, \theta t)$ defined by (4).

PROPOSITION 4.1 *Under assumptions* (A1) *and* (A2), *we have*

(a) *For any given* $\theta \in (0,1]$ *and* $t \in (0,1]$, $\Omega_\theta(t) \subset \Omega$;
(b) *For any closed subset* $Q \subset \Omega^0$, *there exists a* $\theta \in (0,1]$, *such that* $Q \subset \Omega_\theta(1)^0$.

Proof

(a) For $\forall x \in \Omega_\theta(t)$, by Proposition 3.1 (b)

$$g_{\max}(x) \leq \hat{(g)}_\theta(x,t) \leq 0,$$

this means that $x \in \Omega$, so we have $\Omega_\theta(t) \subset \Omega$.
(b) Because Q is bounded and closed and $g_{\max}(x)$ is continuous, there exists a point $x^{(0)} \in Q$ at which $g_{\max}(x)$ reaches its maximum in Q, and $g_{\max}\left(x^{(0)}\right) < 0$. Let $\widetilde{\theta} = \min\{-3kh_{\max}\left(x^{(0)}\right)/(2(k-1)), 1\}$, then by Proposition 3.1 (b), for any $x \in Q$

$$\begin{aligned}\hat{g}_\theta(x,1) &\leq g_{\max}(x) + \frac{\widetilde{\theta}}{3}\left(1 - \frac{1}{k}\right) \\ &\leq g_{\max}\left(x^{(0)}\right) + \frac{\widetilde{\theta}}{3}\left(1 - \frac{1}{k}\right) \\ &= \frac{1}{2} g_{\max}\left(x^{(0)}\right) \\ &< 0,\end{aligned}$$

which means that $x \in \Omega_\theta(1)^0$. Since $\forall x \in Q$, we have $Q \subset \Omega_\theta(1)^0$. □

PROPOSITION 4.2 *If assumptions* (A1)–(A3) *hold, then there exists a* $\theta \in (0,1]$ *such that the boundary of* $\Omega_\theta(t)$ *is regular for any* $t \in (0,1]$, *i.e.* $\forall x \in \partial\Omega_\theta(t)$, $\nabla_x \hat{g}_\theta(x,t) \neq 0$.

Proof If the conclusion is not true, suppose that there exist sequences $\{x^{(k)}\}_{k=1}^{\infty} \in \Omega$ and $\{t_k\}_{k=1}^{\infty} > 0$, such that $t_k \to 0$, as $k \to \infty$, $\hat{g}_{\theta}(x^{(k)}, t_k) = 0$, and

$$\nabla_x \hat{g}_{\theta}(x^{(k)}, t_k) = \sum_{i=1}^{m} \lambda_i(x^{(k)}, t_k)\nabla g_i(x^{(k)}) = 0.$$

By Proposition 3.3, there exist subsequences of $\{x^{(k)}\}_{k=1}^{\infty}$ and $\{\lambda_i(x^{(k)}, t_k)\}_{k=1}^{\infty}$ (without loss of generality, the sequences themselves) which converges, respectively, to some $\bar{x}$ and $\bar{\lambda}_j$. Moreover, $\bar{\lambda}_j = 0$ for $j \notin B(\bar{x})$.

By taking limits, we get

$$g_{\max}(\bar{x}) = 0, \sum_{i \in B(\bar{x})} \bar{\lambda}_i \nabla g_i(\bar{x}) = 0, \sum_{i \in B(\bar{x})} \bar{\lambda}_i = 1,$$

which contradicts assumption (A3). □

Proposition 4.3 *If assumptions* (A1)–(A4) *hold, then for any closed subset* $N \subset \hat{\Omega}$, *there exists a* $\theta \in (0, 1]$ *such that for any* $t \in (0, 1]$ $\Omega_{\theta}(t)$ *satisfies the weak normal cone condition* $w.r.t.$ N.

Proof If the conclusion is not true, suppose that there exist sequences $\{t_k\}_{k=1}^{\infty} > 0$, $\{\bar{x}^{(k)}\}_{k=1}^{\infty} \in \Omega$, $\{\hat{x}^{(k)}\}_{k=1}^{\infty} \in Q$, and $\{\lambda^{(k)}\}_{k=1}^{\infty} > 0$ such that $t_k \to 0$ as $k \to \infty$, $\hat{g}_{\theta}(\bar{x}^{(k)}, t_k) = 0$, and

$$\begin{aligned} \hat{x}^{(k)} &= \bar{x}^{(k)} - \lambda^{(k)} \nabla_x \hat{g}_{\theta}\left(\bar{x}^{(k)}, t_k\right) \\ &= \bar{x}^{(k)} - \lambda^{(k)} \sum_{i=1}^{m} \lambda_i\left(\bar{x}^{(k)}, t_k\right) \nabla g_i\left(\bar{x}^{(k)}\right), \end{aligned}$$

where $\lambda_i(\bar{x}^{(k)}, t_k) \geq 0$ and $\sum_{i=1}^{m} \lambda_i(\bar{x}^{(k)}, t_k) = 1$.

By taking limits, we get $\hat{x} = \bar{x} - \bar{\lambda} \sum_{i \in B(\bar{x})} \bar{\lambda}_i \nabla g_i(\bar{x})$. It is easy to prove that $\bar{x} \in \partial\Omega$ and $\hat{x} \in Q \subset \hat{\Omega}$, $\hat{x} \neq \bar{x}$. This contradicts to assumption (A4). □

Using the single smooth constraint $\hat{g}_{\theta}(x, t)$, we construct the following SSH

$$H(w^{(0)}, w, t) := \begin{pmatrix} (1-t)(\nabla f(x) + \lambda \nabla_x \hat{g}_{\theta}(x, t)) + t(x - x^{(0)}) \\ \lambda \hat{g}_{\theta}(x, t) - t\lambda^{(0)} \hat{g}_{\theta}(x^{(0)}, 1) \end{pmatrix} = 0, \qquad (10)$$

where $w = (x, \lambda)$, $w^{(0)} = (x^{(0)}, \lambda^{(0)})$, $x^{(0)} \in \hat{\Omega}^0$, $\lambda^{(0)} > 0$.

For a given $w^{(0)} \in \hat{\Omega}^0 \times R_{++}^1$, we rewrite $H(w^{(0)}, w, t)$ in (10) as $H_{w^{(0)}}(w, t)$. Let $H_{w^{(0)}}^{-1}(0) = \{(w, t) \in \Omega \times R_{+}^1 \times (0, 1] | H_{w^{(0)}}(w, t) = 0\}$.

Proposition 4.4 *Suppose that assumptions* (A1) *and* (A3) *hold, then* 0 *is a regular value of* $H_{w^{(0)}}$ *for almost all* $w^{(0)} \in \hat{\Omega}^0 \times R_{++}^1$, *and* $H_{w^{(0)}}^{-1}(0)$ *consists of some smooth curves, one of which starts at* $(w^{(0)}, 1)$.

Proof For any $w^{(0)} \in \hat{\Omega}^0 \times R^1_{++}$, and $t \in (0, 1]$,

$$\frac{\partial H(w^{(0)}, w, t)}{\partial w^{(0)}} = \begin{pmatrix} -tI & 0 \\ -t\lambda^{(0)} \nabla_x \hat{g}_\theta(x^{(0)}, 1) & -t\hat{g}_\theta(x^{(0)}, 1) \end{pmatrix},$$

where I is an identity matrix. By a simple calculation,

$$\left| \frac{\partial H(w^{(0)}, w, t)}{\partial w^{(0)}} \right| = (-1)^{n+1} t^{n+1} \hat{g}_\theta \left(x^{(0)}, 1 \right).$$

From $x^{(0)} \in \hat{\Omega}^0$, we get that $\hat{g}_\theta(x^{(0)}, 1) \neq 0$. Thus

$$\left| \frac{\partial H\left(w^{(0)}, w, t\right)}{\partial w^{(0)}} \right| \neq 0.$$

As a mapping of $(w^{(0)}, w, t)$, the Jacobin matrix of $H(w^{(0)}, w, t)$ is of full row rank, this means that 0 is a regular value of $H(w^{(0)}, w, t)$. By Proposition 3.4, $H(w^{(0)}, w, t)$ is an $C^{2,1}$ map, we get that 0 is a regular value of the mapping $H_{w^{(0)}}(w, t)$ using the Theorem 2.4. By Lemma 2.1, $H^{-1}_{w^{(0)}}(0)$ consists of some smooth manifolds. Because $H_{w^{(0)}}(w^{(0)}, 1) = 0$, there must be a smooth curve $\Gamma_{w^{(0)}}$ starting at $(w^{(0)}, 1)$. $\square$

We give the following main theorem on the existence of the smooth path from any interior point $x^{(0)} \in \hat{\Omega}^0$ to a solution of the KKT system (2) of (1).

THEOREM 4.5 *If assumptions (A1)–(A4) hold, H is defined by (10), then for any $\hat{x} \in \hat{\Omega}^0$, there exists an open neighbourhood $N(\hat{x})$ of $\hat{x}$ such that $N(\hat{x}) \subset \hat{\Omega}^0$, and there exists a $\theta \in (0, 1]$ such that $N(\hat{x}) \subset \Omega_\theta(1)^0$, $\partial\Omega_\theta(t)$ is regular and $\Omega_\theta(t)$ satisfies the weak normal cone condition w.r.t. $N(\hat{x})$ for any $t \in (0, 1]$. Furthermore, for almost all $x^{(0)} \in N(\hat{x})$ and $\lambda^{(0)} > 0$, $H^{-1}_{w^{(0)}}(0)$ contains a bounded smooth cure $\Gamma_{w^{(0)}}$ starting at $(w^{(0)}, 1)$ and terminates in or approaches to the hyperplane $t = 0$. Moreover, let (x^*, λ^*, t^*) be any limit point of $\Gamma_{w^{(0)}}$ on the hyperplane $t = 0$, then x^* is a KKT point of* (1).

Proof By Propositions 4.1–4.3, we can easily see that, for any $\hat{x} \in \hat{\Omega}^0$, there exists a neighbourhood $N(\hat{x})$ of $\hat{x}$ such that $N(\hat{x}) \subset \hat{\Omega}^0$, and there exists a $\theta \in (0, 1]$ such that $N(\hat{x}) \subset \Omega_\theta(1)^0$, $\partial\Omega_\theta(t)$ is regular and $\Omega_\theta(t)$ satisfies the weak normal cone condition $w.r.t.$ $N(\hat{x})$ for any $t \in (0, 1]$. By the Propositions 4.4, for almost all $x^{(0)} \in N(\hat{x})$ and $\lambda^{(0)} > 0$, 0 is a regular value of $H_{w^{(0)}}(w, t) : \Omega \times R^1_+ \times (0, 1] \to R^{n+1}$. For given $w^{(0)} \in \hat{\Omega}^0 \times R^1_{++}$, if 0 is a regular value of $H_{w^{(0)}}(w, t) : \Omega \times R^1_+ \times (0, 1] \to R^{n+1}$, from the fact that $H_{w^{(0)}}(w^{(0)}, 1) = 0$, the nonsingularity of $\frac{\partial H(w^{(0)}, w, 1)}{\partial(w)}|_{w=w^{(0)}}$ and the implicit function theorem, we know that $H^{-1}_{w^{(0)}}(0)$ consists of a smooth curve $\Gamma_{w^{(0)}}$, which starts at $(w^{(0)}, 1)$ and goes into $\Omega_\theta(t)^0 \times R^1_{++} \times (0, 1)$ and terminates in the boundary of $\Omega \times R^1_+ \times (0, 1]$.

Let $(\bar{x}, \bar{\lambda}, \bar{t}) \in \Omega \times R^1_+ \times [0, 1]$ be an ending limit point of $\Gamma_{w^{(0)}}$. Only the following five cases are possible :

(1) $(\bar{x}, \bar{\lambda}, \bar{t}) \in \Omega \times R^1_+ \times \{0\}$ and $\bar{\lambda} < +\infty$;
(2) $(\bar{x}, \bar{\lambda}, \bar{t}) \in \Omega_\theta(1) \times R^1_+ \times \{1\}$ and $\bar{\lambda} < +\infty$;

(3) $(\bar{x}, \bar{\lambda}, \bar{t}) \in \Omega_\theta(\bar{t}) \times \{+\infty\} \times [0, 1]$;
(4) $(\bar{x}, \bar{\lambda}, \bar{t}) \in \partial\Omega_\theta(\bar{t}) \times R^1_{++} \times (0, 1)$ and $\bar{\lambda} < +\infty$;
(5) $(\bar{x}, \bar{\lambda}, \bar{t}) \in \Omega_\theta(\bar{t}) \times \{0\} \times (0, 1)$.

Because $H_{w^{(0)}}(w, 1) = 0$ has only one solution $(x^{(0)}, \lambda^{(0)})$ in $\Omega \times R^1_+$, case (2) is impossible. By the continuity of Γ and the second equality of (10), we know that cases (4) and (5) are impossible.

Using Propositions 4.2 and 4.3, similar to the proof of Theorem 2.9 of Yu et al. [11], we can prove case (3) is impossible.

As a conclusion, case (1) is the only possible case. That is $\Gamma_{w^{(0)}}$ must be bounded and approach to the hyperplane at $t = 0$.

By the boundedness of x, λ for $H_{w^{(0)}}(x, \lambda, t) = 0$ and $t \in (0, 1]$ and $0 \le \lambda_i(x, t) \le 1$ in Proposition 3.2(b), we know $(x, \lambda, \lambda_1(x, t), \dots, \lambda_m(x, t))$ has at least one accumulation point as $t \to 0_+$. Let $(x^*, \lambda^*, \lambda_1^*, \dots, \lambda_m^*)$ be an accumulation point of $(x, \lambda, \lambda_1(x, t), \dots, \lambda_m(x, t))$ and $y_i^* = \lambda^* \lambda_i^*$, by (1) we have

$$\nabla f(x^*) + \sum_{i=1}^{m} y_i^* \nabla_x \hat{g}_i(x^*) = 0.$$

From $g_i(x) \le \hat{g}_\theta(x, t) \le 0$, $\lambda > 0$ and $\lambda_i(x, t) \ge 0$ for $t \in (0, 1]$ and $H_{w^{(0)}}(x, \lambda, t) = 0$, we have $g_i(x^*) \le 0$ and $\lambda_i^* \ge 0$. If $x^* \in \Omega^0$, then $\lim_{x \to x^*, t \to 0_+} \hat{g}_\theta(x, t) = g_{\max}(x^*) < 0$. and then by the second equality of (10) $\lambda^* = 0$, else $x^* \in \partial\Omega$, $\lim_{x \to x^*, t \to 0_+} \hat{g}_\theta(x, t) = g_{\max}(x^*) = 0$ by the Proposition 3.3 (b), $\lambda_i^* = 0$ for $i \notin \bar{B}(x^*)$. Thus, we have that $y_i^* g_i^*(x^*) = 0$ for $1 \le i \le m$.

We get that $(x^*, y_1^*, \dots, y_m^*)$ is a solution of (2), which means that x^* is a KKT point of (1) and $y_1^*, \dots, y_m^*$ are corresponding Lagrangian multipliers. □

5. The SSH-S-N procedure and numerical experiments

5.1. *The SSH-S-N procedure*

In this section, we give a predictor-corrector algorithm – SSH-S-N procedure to trace the path generated by the spline smoothing homotopy, in which secant predictor and Newton corrector steps are used.

The first predictor step is tangent predictor, other predictor steps are secant predictor. Step length is adjusted according to the angle between current and previous predictor directions and the times of iteration of previous corrector step. The corrector step is Newton corrector along with the direction that vertical to the predictor director. Particularly, in order to improve the efficiency of SSH-S-N, the algorithm includes an end game strategy, in which we use the standard Newton's method to solve

$$F_{(\theta, t_c)}(x, \lambda) = \begin{pmatrix} \nabla f(x) + \lambda \nabla_x \hat{g}_\theta(x, t_c) \\ \lambda \hat{g}_\theta(x, t_c) \end{pmatrix} = 0,$$

where t_c is a small positive constant.

Algorithm 5.1 (the **SSH-S-N** procedure)

Step 0: Give $\theta \in (0, 1]$, t_{end} and t_c, starting point $w^{(0)} \in \hat{\Omega}^0 \times R^1_{++}$, initial step length h_1, step contraction factors $B_{\min}$, step expansion factors $B_{\max}$, tracking tolerances H_{tol} and H_{final} for correction.

Step 1: Let $t_0 = 1$, enter Subroutine 5.1, compute $H'_{w^{(0)}}(w^{(0)}, t_0)$. Let $d^{(0)} = (0, \ldots, 0, -1) \in R^{n+2}$, compute the predictor direction d by solving the following system of equation:

$$\begin{pmatrix} H'_{w^{(0)}}(w^{(0)}, t_0) \\ d^{(0)T} \end{pmatrix} d = -d^{(0)},$$

set $d^{(1)} = \frac{d}{\|d\|}$, $k = 1$, $N_{good} = 2$, goto Step 4.

Step 2: **(predictor step)** Compute the predictor direction $d^{(k)} = \frac{(w^{(k-1)}, t_{k-1}) - (w^{(k-2)}, t_{k-2})}{\|(w^{(k-1)}, t_{k-1}) - (w^{(k-2)}, t_{k-2})\|}$, the angle between the current and the last predictor directions $\beta^{(k)} = \arccos((d^{(k)})^T d^{(k-1)})$.

Step 3: If corrector step fail or $\beta^k > \pi/4$, set $h_k = B_{\min}(1)h_{k-1}$, $N_{good} = 0$.
If $i \geq 5$, set $h_k = B_{\min}(2)h_{k-1}$, $N_{good} = 0$.
If $i = 4$, set $h_k = h_{k-1}$, $N_{good} = N_{good} + 1$.
If $i = 3$, set $N_{good} = N_{good} + 1$.
If $N_{good} > 2$, set $h_k = \min(1, B_{\max}(2)h_{k-1})$.
If $i \leq 2$, set $N_{good} = N_{good} + 1$.
If $N_{good} > 2$, set $h_k = \min(1, B_{\max}(1)h_{k-1})$.

Step 4: If $h_k < 10^{-10}$, stop the algorithm with an error flag,
Else compute the predictor point $(w^{(k,0)}, t_{k,0}) = (w^{(k-1)}, t_{k-1}) + h_k d^{(k)}$, set $i = 0$.

Step 5: If $t_{k,0} \leq t_{end}$, adjust step length h_k, compute a new predictor point $(w^{(k,0)}, 0)$. If $w^{(k,0)}$ is feasible, set $l = 0$, goto the end game.

Step 6: **(corrector step)** If $i = 5$, $w^{(k)} = w^{(k-1)}$, $t_k = t_{k-1}$, $d^{(k+1)} = d^{(k)}$, replace k by $k + 1$, goto Step 3,
Else, enter Subroutine 5.1, compute $H'_{w^0}(w^{(k,i)}, t_{k,i})$ and then compute the corrector direction $d^{(k,i+1)}$ by

$$\begin{pmatrix} H'_{w^{(0)}}(w^{(k,i)}, t_{k,i}) \\ d^{kT} \end{pmatrix} d^{(k,i+1)} = \begin{pmatrix} -H_{w^{(0)}}(w^{(k,i)}, t_{k,i}) \\ 0 \end{pmatrix},$$

compute the corrector point

$$(w^{(k,i+1)}, t_{k,i+1}) = (w^{(k,i)}, t_{k,i}) + d^{(k,i+1)},$$

replace i by $i + 1$.

Step 7: If $t_{k,i} \geq 1$ or $w^{(k,i)}$ is infeasible, $w^{(k)} = w^{(k-1)}$, $t_k = t_{k-1}$, $d^{(k+1)} = d^{(k)}$, replace k by $k + 1$, the corrector step fails, goto Step 3.
If $t_{k,i} < 0$, adjust the step length, compute a new predictor point $(w^{(k,0)}, 0)$.
If $w^{(k,0)}$ is feasible, set $l = 0$, goto the end game,
Else set $w^{(k)} = w^{(k-1)}$, $t_k = t_{k-1}$, $d^{(k+1)} = d^{(k)}$, replace k by $k + 1$, the corrector step fails, goto Step 3.

Step 8: If $\|d^{(k,i)}\| > H_{tol}$ and $\|H_{w^{(0)}}(w^{(k,i)}, t_{k,i})\|_{\inf} > H_{tol}$, goto Step 6,

Else $w^{(k)} = w^{(k,i)}$, $t_k = t_{k,i}$, if $t_k < t_c$, return with $x^* = x^{(k-1)}$, stop the algorithm,
Else set $H_{tol} = \min\{H_{tol}, t_k\}$, replace k by $k+1$, goto Step 2.

Step 9: **(the end game)** Enter Subroutine 5.1, compute $F_{(\theta,t_c)}(w^{(k,l)})$, $F'_{(\theta,t_c)}(w^{(k,l)})$ and then computer $d^{(k,l+1)} = -(F'_{(\theta,t_c)}(w^{(k,l)}))^{-1}F_{(\theta,t_c)}(w^{(k,l)})$, the corrector point $w^{(k,l+1)} = w^{(k,l)} + d^{(k,l+1)}$, replace l by $l+1$.

Step 10: If $\|F_{(\theta,t_c)}(w^{(k,l)})\|_{inf} \le H_{final}$, $\|d^{(k,l)}\| \le H_{final}$, return with $x^* = x^{(k,l)}$, stop the algorithm.

Step 11: If $l = 5$ or $\|d^{(k,l)}\| > \|d^{(k,l-1)}\|$, set $t_{end} = 0.3 \times t_{end}$, $w^{(k)} = w^{(k-1)}$, $t_k = t_{k-1}$, $d^{(k+1)} = d^{(k)}$, replace k by $k+1$, goto Step 3,
Else goto step 9.

Subroutine 5.1 (Search the cell).

substep 1: Let $\bar{I} = \{j|g_{\max}(x^{(k,i)}) - g_j(x^{(k,i)}) < \theta t_{k,i}\}$, $\bar{k}$ be the cardinality of $\bar{I}$, and $\bar{I} = \{i_1, i_2, \ldots, i_{\bar{k}}\}$. Range $\{g_{i_j}(x^{(k,i)})\}_{j=1}^{\bar{k}}$ according to $g_{i_1}(x^{(k,i)}) \ge g_{i_2}(x^{(k,i)}) \ge \ldots \ge g_{i_{\bar{k}}}(x^{(k,i)})$.

substep 2: If $\bar{k} = 1$, the cell is $\triangle_{i_1}(\theta t_{k,i})$,
Else, if $(\tilde{k}-1)g_{i_{\tilde{k}}}(x^{(k,i)}) - \sum_{j=1}^{\tilde{k}-1} g_{i_j}(x^{(k,i)}) + \theta t_{k,i} \ge 0$ for every $\tilde{k} \in \{\bar{k}, \bar{k}-1, \ldots, 2\}$, we have $\tilde{k} \in I \subseteq \{\bar{k}, \bar{k}-1, \ldots, 2\}$.

substep 3: Set $\hat{k}$ is the maximum element of I, then the cell is $\triangle_{i_1 \ldots i_{\hat{k}}}(\theta t_{k,i})$, and exists Subroutine 5.1.

5.2. *Numerical experiment*

We have implemented the SSH-S-N algorithm using the MATLAB. In order to show the efficiency of the algorithm, we have also implemented CHIP, ACH methods using similar procedures. We compare these algorithms with KNITRO which is a solver for large nonlinear optimization, where KNITRO provides three state-of-art algorithms for solving problems and active set algorithm is suitable for solving nonlinear programming problem with many constraints. We choose it and its parameters as default values. We choose problem 5.1 from the CUTEr test set [17], problem 5.2 in [18], two problems 5.3 and 5.4 in [19], problems 5.5 in [20] and give problem 5.6, respectively.

The test results were obtained by running MATLAB R2011a on a desktop with Windows XP Professional operation system, Intel(R) Core(TM) i3-370 2.40 GHz processor and 2.92 GB of memory. The default parameters are chosen as follows:

Table 1. Test results for Example 5.1.

Method	$f(x^*)$	$g_{\max}(x^*)$	Time
CHIP	−0.999887528714237	−8.7260e-005	22.7158
ACH	−0.999999968425505	−3.1574e-008	1.0125
KNITRO	−0.999999982183160	2.4583e-011	0.6060
SSH	−1.0000	0.0000	0.2589

Table 2. Test results for Example 5.2.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	Time
	CHIP	1.000000000000000	−2.2204e-016	60.6359
100^2	ACH	1.000000000138630	−1.3863e-010	3.1371
	KNITRO	1.000000080009031	8.8818e-16	0.7430
	SSH	1.000000000025000	−2.5000e-011	0.6278
	CHIP	1.000003049566824	−3.0492e-006	883.4582
200^2	ACH	1.000000000138629	−1.3863e-010	15.2567
	KNITRO	1.000000080094647	8.8818e-16	12.448
	SSH	1.000000000025000	−2.5000e-011	1.0463
	CHIP	–	–	fail[3]
300^2	ACH	1.000000000138629	−1.3863e-010	36.7205
	KNITRO	1.000000080248269	8.8818e-16	26.105
	SSH	1.000000000025000	−2.5000e-011	2.2792
	CHIP	–	–	fail[3]
500^2	ACH	1.000000000138630	−1.3863e-010	105.8642
	KNITRO	1.000000080687340	8.8818e-16	69.258
	SSH	1.000000000025000	−2.5000e-011	5.9249
	CHIP	–	–	fail[3]
1000^2	ACH	1.000000000138630	−1.3863e-010	407.0833
	KNITRO	–	–	fail[2]
	SSH	1.000000000025000	−2.5000e-011	18.3479

- Parameter $\theta = 0.0001$;
- Parameters in end game section $t_c = 10^{-6}$, $t_{end} = 0.1$, $m_{\max} = 5$;
- Step size parameters $h_0 = 0.1$, $B_{\min} = [0.5, 0.75]$, $B_{\max} = [3, 1.5]$;
- Tracking tolerances $H_{tol} = 10^{-3}$, $H_{final} = 10^{-12}$;
- Initial Lagrangian multipliers $(1, \ldots, 1) \in R^m$ for CHIP method, 1 for ACH and SSH methods.

For different problems, we list the objective function $f(x^*)$, the max function $g_{\max}(x^*)$ of constraints at x^* and CPU time in seconds, where x^* is approximate solution computed by the corresponding algorithm. For the problems that were not solved by the conservative setting, we also give the reason for failure. The notation 'fail[1]' indicates the step length in predictor step is smaller than 10^{-10} before $t = 0$. This 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 Newton iterations or 3600 s.

Example 5.1 [17]

$$
\begin{aligned}
f(x) &= x_2,\\
g_i(x) &= -x_1\cos(2\pi i/10,000) - x_2\sin(2\pi i/10,000) - 1,\\
&\quad i = 1, \ldots, 10,000.\\
x^{(0)} &= (0.8, 0.5) \in R^2.
\end{aligned}
$$

Table 3. Test results for Example 5.3.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	Time
	CHIP	–	–	fail[3]
10^5	ACH	97.158852437685624	0.000	5.4201
	KNITRO	97.158852529936652	−4.4627e-06	3.542
	SSH	97.158852437685624	0.000	0.9941
	CHIP	–	–	fail[3]
5×10^5	ACH	97.158852437685624	0.000	29.0840
	KNITRO	97.158852609014517	−4.4627e-06	37.838
	SSH	97.158852437685624	0.000	3.9175
	CHIP	–	–	fail[3]
10^6	ACH	97.158852437685624	0.000	55.8894
	KNITRO	–	–	fail[2]
	SSH	97.158852437685624	0.000	7.2850
	CHIP	–	–	fail[3]
5×10^6	ACH	97.158852437685624	0.000	271.3043
	KNITRO	–	–	fail[2]
	SSH	97.158852437685624	0.000	34.4681
	CHIP	–	–	fail[3]
10^7	ACH	–	–	fail[2]
	KNITRO	–	–	fail[2]
	SSH	97.158852437685624	0.000	66.6486

Example 5.2 [18]

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

Example 5.3 [19]

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

Table 4. Test results for Example 5.4.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	Time
	CHIP	–	–	fail[1]
10^4	ACH	2.430533988749895	0.000	1.4062
	KNITRO	2.430533988749895	2.2204e-16	0.8710
	SSH	2.430533988749895	0.000	0.8121
	CHIP	–	–	fail[1]
2×10^4	ACH	2.430533988749907	−8.6597e-015	2.4540
	KNITRO	2.430533988749907	2.2204e-16	2.9780
	SSH	2.430533988749895	0.000	0.8360
	CHIP	–	–	fail[1]
4×10^4	ACH	2.430533988763167	−9.1704e-012	5.0861
	KNITRO	2.454433988763167	2.2204e-16	7.4170
	SSH	2.430533988749895	0.000	0.9432
	CHIP	–	–	fail[3]
6×10^4	ACH	2.430533988795290	−3.1367e-011	7.5444
	KNITRO	2.430633988795290	2.2204e-16	13.5780
	SSH	2.430533988749895	0.000	1.4406
	CHIP	–	–	fail[3]
8×10^4	ACH	2.430533988823046	−5.0546e-011	10.6630
	KNITRO	2.430633988823046	2.2204e-16	16.5860
	SSH	2.430533988751606	−1.1826e-012	1.8757

Table 5. Test results for Example 5.5 with $n = 30$.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	Time
CHIP	0.031453202660645	−5.4549e-003	9.1668	
10^3	ACH	0.031022760952790	0.0000	1.7579
	KNITRO	0.031022772908412	−2.4737e-005	2.969
	SSH	0.031022760952790	0.0000	0.1557
	CHIP	0.041433274661343	−0.1214	267.8435
10^4	ACH	0.031022791079590	−2.2204e-016	4.8858
	KNITRO	0.031022814962909	2.4948e-005	49.313
	SSH	0.031022791079590	0.0000	3.0114
	CHIP	–	–	fail[3]
10^5	ACH	0.031022793893199	−1.3544e-014	46.4658
	KNITRO	–	–	fail[2]
	SSH	0.031022793893198	2.2204e-016	23.4122
	CHIP	–	–	fail[2]
10^6	ACH	–	–	fail[2]
	KNITRO	–	–	fail[2]
	SSH	0.031023608781846	−1.0776e-005	344.3138

Table 6. Test results for Example 5.5 with $m = 100$.

n	Method	$f(x^*)$	$g_{\max}(x^*)$	Time
	CHIP	0.029399883417009	3.3306e-016	6.1666
200	ACH	0.029399883417024	−1.9417e-013	1.8751
	KNITRO	0.029399893453975	2.8306e-005	11.531
	SSH	0.029399883417017	−1.0147e-013	0.9165
	CHIP	0.029488614569146	−2.4349e-004	9.1659
400	ACH	0.029399883380900	−1.8607e-013	6.0711
	KNITRO	0.029399893397862	2.8306e-005	45.266
	SSH	0.029399883380893	−8.3377e-014	3.4166
	CHIP	0.029399883380886	2.2204e-016	16.7669
600	ACH	0.029399883380896	−1.2589e-013	14.3287
	KNITRO	0.029399893747722	2.8306e-005	97.219
	SSH	0.029399883380898	−1.5487e-013	7.8640
	CHIP	0.029567449317801	−1.8695e-004	24.6168
800	ACH	0.029399883380901	−1.9850e-013	29.9600
	KNITRO	0.029399893739437	2.8306e-005	191.484
	SSH	0.029399883380894	−9.9698e-014	15.7286
	CHIP	0.029773104560561	−3.0575e-004	32.0488
1000	ACH	0.029399883380894	−1.0869e-013	52.7604
	KNITRO	0.029399894202721	2.8306e-005	308.438
	SSH	0.029399883380910	−3.0819e-013	27.4092

Table 7. Test results for Example 5.6 with $n = 800$.

m	Method	$f(x^*)$	$g_{\max}(x^*)$	Time
	CHIP	20.210635574436939	−2.1538e-014	193.0102
100	ACH	20.210635574439873	−5.7731e-015	253.3545
	KNITRO	20.2106356156303	2.0772e-004	226.125
	SSH	20.210636266333168	−2.6657e-008	164.1312
	CHIP	40.424305960672918	−5.6119e-008	572.7819
200	ACH	40.424303616164572	−1.0956e-008	857.4890
	KNITRO	40.4243031284032	5.0772e-004	570.047
	SSH	40.424303047400919	1.3100e-014	496.7975
	CHIP	60.637300729793594	−5.0914e-008	1079.8707
300	ACH	60.637297549465295	−1.0072e-008	2037.5643
	KNITRO	60.6372968808955	-0.0013	3062.406
	SSH	60.637297566921731	−1.029e-008	984.8547
	CHIP	80.850125666102116	−3.4885e-008	1956.3840
400	ACH	80.850123007681901	−9.2807e-009	3167.2833
	KNITRO	–	–	fail[2]
	SSH	80.850123015835678	−9.3593e-009	1730.3105
	CHIP	101.0628837497786	−2.9297e-008	3471.5560
500	ACH	101.0628808199577	−6.7224e-009	4577.8916
	KNITRO	–	–	fail[2]
	SSH	101.0628808093738	−6.6411e-009	2364.7964

Table 8. Test results for Example 5.6 with $m = 100$.

n	Method	$f(x^*)$	$g_{\max}(x^*)$	Time
	CHIP	20.210635574437084	2.2648e-014	11.8135
100	ACH	20.210635574438012	3.2862e-014	8.8959
	KNITRO	20.2106356158759	−9.2275e-005	50.906
	SSH	20.210635574438182	2.4424e-014	5.1139
	CHIP	20.210635574438850	−3.1752e-014	356.1564
1000	ACH	20.210635574436136	3.1086e-015	323.3035
	KNITRO	20.2106356157321	0.0010	312.234
	SSH	20.210635574438907	3.5971e-014	178.2559
	CHIP	20.210638676823059	-1.1953e-007	1544.8107
2000	ACH	20.210635574434164	2.1760e-014	1596.5091
	KNITRO	20.2106356574943	−9.2275e-005	1075.922
	SSH	20.210636182608813	−2.3432e-008	814.3553
	CHIP	20.210635574439568	2.6867e-014	3618.9849
3000	ACH	20.210635574442843	7.1054e-015	3698.8322
	KNITRO	20.2106395893774	−9.2275e-005	3691.094
	SSH	20.210635574442843	7.1054e-015	3129.8278
	CHIP	–	–	fail^2
4000	ACH	20.210635574441049	8.2156e-015	7527.6500
	KNITRO	–	–	fail^2
	SSH	20.210635574441049	8.2156e-015	6397.9948

Table 9. The time of searching the cell for Examples 5.1–5.4.

	Example 5.1	Example 5.2	Example 5.3	Example 5.4
m	–	1000^2	10^7	8×10^4
Time1	0	7.2896	14.0595	0.2968
Time2	0.2589	18.3479	66.6489	1.8757

Note: Time1 is the time of searching the cell and Time2 is the CPU time.

Example 5.4 [19]

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

Example 5.5 [20]

$$
\begin{aligned}
f(x) &= x^T x/2,\\
g_i(x) &= 3 + 4.5\sin(4.7\pi(t_i - 1.23)/8) - \sum_{k=1}^{n} x_k t_i^{k-1},
\end{aligned}
$$

$$
\begin{aligned}
t_i &= i/m,\ i = 1, \dots, m. \\
x^{(0)} &= (1, \dots, 1) \in R^n.
\end{aligned}
$$

Example 5.6

$$
\begin{aligned}
f(x) &= \sum_{i=1}^{m} \sum_{j=1}^{n} (a_{i,j} x_j)^2, \\
g_i(x) &= \tan(t_i) - \sum_{k=1}^{n} x_k t_i^{k-1}, \\
a_{i,j} &= i/m + j/(100n), \\
t_i &= i/m, \\
i &= 1, \dots, m,\ \ j = 1, \dots, n. \\
x^{(0)} &= (1, 5, 0, \dots, 0) \in R^n.
\end{aligned}
$$

5.3. *Remarks*

Now, we give some remarks on numerical results.

(1) From Tables 1–5 and 7, we see when n is fixed, the advantage of SSH method is very obvious as m increases. For nonlinear programming with large number of complicated constraints, the SSH method can save much computation of the gradient and the Hessian of constraint functions and hence is more efficient than the CHIP and ACH method. Even compare with very successful optimization softwares like KNITRO, the SSH method is very encouraging by above preliminary numerical tests (Tables 1–8).
(2) From Table 9, we know the efficiency of the SSH method is influenced by the method of searching the cell. If it can be improved, the SSH method will get better performance.
(3) Algorithm 5.1 is a simple implementation of the SSH method. It needs to do much work to improve implementation of the SSH method on all processes of numerical path tracing, say, schemes of predictor and corrector, step length updating, linear system solving and end game.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The research was supported by the National Natural Science Foundation of China [grant number 11171051], [grant number 91230103]; the Fundamental Research Funds for the Central Universities [DC201502050408]; the Doctoral Starting up Foundation of Dalian Nationalities University, China [0701110100].

References

[1] Kortanek KO, Potra F, Ye Y. On some efficient interior point algorithms for nonlinear convex programming. Linear Algebra Appl. 1991;152:169–189.

[2] Monteiro RDC, Adler I. An extension of Karmarkar type algorithm to a class of convex separable programming problems with global linear rate of convergence. Math. Oper. Res. 1990;15:408–422.

[3] Wang Y, Feng GC, Liu TZ. Interior point algorithm for convex nonlinear programming problems. Numer. Math. J. Chinese Univ. 1992;1:1–8.

[4] Zhu J. A path following algorithm for a class of convex programming problems. ZOR-Methods Models Oper. Res. 1992;36:359–377.

[5] Feng GC, Lin ZH, Yu B. Existence of an interior pathway to a Karush–Kuhn–Tucker point of a nonconvex programming problem. Nonlinear Anal. 1998;32:761–768.

[6] Feng GC, Yu B. Combined homotopy interior point method for nonlinear programming problems. In: Advances in numerical mathematics; Proceedings of the Second Japan–China Seminar on Numerical Mathematics (Tokyo, 1994). Vol. 14, Lecture notes in numerical and applied analysis. Tokyo: Kinokuniya; 1995. p. 9–16.

[7] Lin ZH, Yu B, Feng GC. A combined homotopy interior point method for convex nonlinear programming. Appl. Math. Comput. 1997;84:193–211.

[8] Liu QH, Yu B, Feng GC. An interior point path following method for nonconvex nonlinear programming problem with quasi normal cone condition. Adv. Math. 2000;19:281–282.

[9] Yu B, Liu QH, Feng GC. A combined homotopy interior point method for nonconvex programming with pseudo cone condition, Northeast. Math. J. 2000;16:383–386.

[10] Shang YF, Yu B. Boundary moving combined homotopy method for nonconvex nonlinear programming and its convergence. J. Jilin Univ. Sci. 2006;44:357–361.

[11] Yu B, Feng GC, Zhang SL. The aggregate constraint homotopy method for nonconvex nonlinear programming. Nonlinear. Anal. 2001;45:839–847.

[12] Li XS. An aggregate function method for nonlinear programming. Sci. China (Ser. A). 1991;34:1467–1473.

[13] Zhao GH, Wang ZR, Mou HN. Uniform approximation of min–max functions by smooth splines. J. Comput. Appl. Math. 2011;236:699–703.

[14] Bates SM. Toward a precise smoothness hypothesis in Sard's theorem. Proc. Amer. Math. Soc. 1993;117:279–283.

[15] Naber GL. Topological methods in Euclidean spaces. Mineola (NY): Dover Publications Inc.; 2000. Reprint of the original (1980).

[16] Abraham R, Robbin J. Transversal mappings and flows. New York (NY): Benjamin; 1967.

[17] Gould NIM, Orban D, Toint PL. CUTEr (and sifdec), a constrained and unconstrained testing environment, revisted. Technical report TR/PA/01/04. Toulouse, France: CERFACS; 2001.

[18] Zhou ZY, Yu B. The flattened aggregate constraint homotopy method for nonlinear programming problems with many nonlinear constraints. Abstr. Appl. Anal. 2014;2014:1–14.

[19] Li DH, Qi LQ, Tam J, Wu SY. A smoothing Newton method for semi-infinite programming. J. Global Optim. 2004;30:169–194.

[20] Ni Q, Ling C, Qi LQ, Teo KL. A truncated projected Newton-type algorithm for large-scale semi-infinite programming. SIAM J. Optim. 2006;16:1137–1154.