Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL
Pacific Journal of Optimization
Vol. 10, No. 1, pp. 37–51, January 2014

A CONSTRAINT SHIFTING SPLINE SMOOTHING HOMOTOPY METHOD FOR GENERAL NONLINEAR PROGRAMMING*

Li Dong, Bo Yu† and Zhengyong Zhou

Abstract: In this paper, a constraint shifting spline smoothing homotopy method (CSSSH) is proposed for solving nonlinear programming with both equality and inequality constraints. The global convergence of CSSSH under fairly weak conditions is proven. All problems that can be solved by previous homotopy methods can also be solved by the proposed method. The CSSSH-S-N procedure is given. The spline smoothing technique uses a smooth inequality constraint instead of m inequality constraints. At each iteration, the smooth spline approximation of the max-function of the inequality constraints involves only few inequality constraints, hence the number of gradient and Hessian calculations is reduced dramatically, so the CSSSH is more efficient than previous homotopy methods like, combined homotopy interior point method(CHIP), combined homotopy infeasible interior point method (CHIIP) and constraint shifting homotopy method (CSH). And the starting point in the proposed method is not necessarily a feasible point or an interior point, so it is more convenient to be implemented than CHIP and CHIIP. Numerical tests with the comparisons to some other methods show that the new method is very efficient for nonlinear programming with both equality and large number of complicated inequality constraints.

Key words: *spline function, homotopy method, nonlinear programming, global convergence*

Mathematics Subject Classification: *90C30, 49M37*

1 Introduction

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

$$\begin{array}{ll} \min & f(x), \\ \text{s.t.} & g(x) \leq 0, \\ & h(x) = 0, \end{array} \tag{1.1}$$

where $f : R^n \to R$, $g : R^n \to R^m$, and $h : R^n \to R^p$. Let $\Omega = \{x \in R^n | g(x) \leq 0, h(x) = 0\}$, $\Omega^{(0)} = \{x \in R^n | g(x) < 0, h(x) = 0\}$, $\partial\Omega = \Omega \backslash \Omega^{(0)}$. To our knowledge, a point x^* that satisfies

$$\begin{array}{l} \nabla f(x) + \sum\limits_{i=1}^{m} y_i \nabla g_i(x) + \sum\limits_{i=1}^{p} z_i \nabla h_i(x) = 0, \\ y_i g_i(x) = 0, \quad y_i \geq 0, g_i(x) \leq 0, i = 1, \ldots, m, \\ h_j(x) = 0, j = 1, \ldots, p, \end{array} \tag{1.2}$$

*This work was supported by the National Natural Science Foundation of China (11171051, 91230103), the Fundamental Research Funds for the Central Universities (DC13010214).

†Corresponding author.

where $y \in R^m_+$ and $z \in R^p$, is called a KKT point of (1.1). There are many numerical methods for solving the nonlinear programming (1.1), including sequential quadratic programming methods, augmented Lagrangian methods, interior point methods, homotopy methods and so on. For solving nonconvex programming, the homotopy method is a good methodology since its global convergence to a KKT point can be proven under fairly weak conditions. In the last three decades, some results on the homotopy methods for nonlinear programming have been given, see, [5,6,10–12,17,18]. In [12], Shang and Yu proposed a constraint shifting combined homotopy method (abbreviated by CSCH) for nonlinear programming with only inequality constraints, which could choose the initial point outside the feasible region, it could solve some problems that did not satisfy the normal cone condition. In [9], Lin, Li and Yu proposed a combined homotopy interior point method(CHIP) for solving nonconvex programming with both equality and inequality constraints, as in (1.1). The homotopy equation was defined as

$$H(x,y,z,t) = \begin{pmatrix} (1-t)(\nabla f(x) + \nabla g(x)y) + \nabla h(x)z + t(x - x^{(0)}) \\ Yg(x) - tY^{(0)}g(x^{(0)}) \\ h(x) \end{pmatrix} = 0, \tag{1.3}$$

where $(y,z) \in R^m_+ \times R^p$, $\Omega^{(0)} = \{g(x) < 0,\, h(x) = 0\}$, $(x^{(0)}, y^{(0)}) \in \Omega^{(0)} \times R^m_{++}$, $t \in [0,1]$, $Y = \mathrm{diag}(y_1, y_2, \ldots, y_m)$, $Y^{(0)} = \mathrm{diag}(y_1^{(0)}, y_2^{(0)}, \ldots, y_m^{(0)})$. Under the nonemptiness and boundedness of $\Omega^{(0)}$, linear independence constraint qualification (LICQ) and a normal cone condition on both inequality and equality constraints, existence and global convergence to a KKT point of a smooth homotopy path were proven. Due to the last component, $h(x) = 0$, the homotopy equation (1.3) needs an interior starting point $x^{(0)}$ which satisfies also equality constraints. Because finding a point satisfying both equality and strict inequality constraints may be as difficult as the solving original problem, the CHIP is not convenient to use. In [15], Yang, Yu and Xu proposed a combined homotopy infeasible interior point method(CHIIP) for (1.1). It required only the starting point to be an interior point and not to be a feasible point, so it was more practical than CHIP for (1.1). Under a normal cone condition concerning only with the inequality constraints, as well as a stronger positive linear independence assumption, existence and global convergence of the homotopy path with probability one to a solution of the KKT system were proven. In [16], Yang, Yu and Xu proposed a constraint shifting homotopy method(CSH) for (1.1). Under some conditions including the nonemptiness and boundedness of parameterized feasible sets, positive-linear independence and a normal cone condition about parameterized constraints, existence and global convergence of the homotopy path were proven.

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, using cubic spline which was introduced in [19] to smoothly approximate the min (or max) function, a constraint shifting spline smoothing homotopy (abbreviated by CSSSH) is constructed and an interior path following method—constraint shifting spline smoothing homotopy method for general nonlinear programming is proposed. The spline smoothing technique uses a smooth inequality constraint instead of m inequality parameterized constraints and acts also as an active set technique, so it can improve the efficiency of the homotopy method. Under the condition that the nonemptiness and boundedness of parameterized feasible sets, positive-linear independence and the initial feasible set, not necessarily the original feasible set, satisfies the normal cone condition, existence and global convergence of the homotopy path are proven. In addition, CSSSH does not require starting from an interior point, so it is convenient to be implemented.

The rest of this paper is organized as follows. In section 2, we give the constraint shifting spline smoothing homotopy and prove some propositions and main theorem on the existence and global convergence of the homotopy path. In section 3, a procedure for tracking the homotopy path is listed and several numerical examples are given.

2 The Homotopy and Homotopy Path

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

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

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 Parameterized Sard theorem in [1] are commonly used for proving the regularity of a homotopy. In this paper, in order to use a spline function with degree as low as possible, namely S_3^2, in CSSSH, we use the following parameterized Sard theorem with $C^{r,1}$ smoothness.

Theorem 2.1 ([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 $r > \max\{0, m-k\}$. If $0 \in R^k$ is a regular value of f, then for almost all $a \in V$, 0 is a regular value of $f_a = f(a, \cdot)$.*

Let $\tilde{g}(x,t) = g(x) - t^\alpha c$, where $\alpha \geq 1$, $c \in R^m_+$. For given $x^{(0)}$, let $\Omega(t) = \{x \in R^n | \tilde{g}(x,t) \leq 0, h(x) - th(x^{(0)}) = 0\}$, $\Omega^{(0)}(t) = \{x \in R^n | \tilde{g}(x,t) < 0, h(x) - th(x^{(0)}) = 0\}$, $\partial\Omega(t) = \Omega(t) \backslash \Omega^{(0)}(t)$, $\tilde{\Omega} = \bigcup_{t \in [0,1]} \Omega(t)$, $I(x,t) = \{j \in \{1, 2, \ldots, m\} | \tilde{g_j}(x,t) = 0\}$. It is obvious that $\Omega(0) = \Omega$.

The following hypotheses will be used in this paper. There exist a vector $c \in R^m_+$ and an open subset V of the set $\{x | \tilde{g}(x,t) < 0\}$ satisfying following assumptions:

(A1) For any $t \in [0,1]$, $\Omega^{(0)}(t)$ is nonempty, and $\tilde{\Omega}$ bounded;

(A2) For any $x \in V$, the matrix $\nabla h(x)$ has full column rank. For any $t \in [0,1]$, and $x \in \Omega(t)$, $\sum_{i \in I(x,t)} \alpha_i \nabla \tilde{g}_i(x,t) + \sum_{j=1}^{p} \beta_j \nabla h_j(x) = 0$ for $\alpha_i \in R_+$ and $\beta_j \in R$ implies $\alpha_i = \beta_j = 0$, $i \in I(x,t)$, $j = 1, \ldots, p$;

(A3) $\forall x \in \Omega(1)$,

$$\{x + \nabla h(x)z + \sum_{i \in I(x,1)} y_i \nabla \tilde{g}_i(x,1) : z \in R^p, y_i \geq 0, i \in I(x,1)\} \cap \Omega(1) = \{x\}.$$

We know $\tilde{g}(x,t) \leq 0$ is equivalent to $\tilde{g}_{\max}(x,t) \leq 0$, where $\tilde{g}_{\max}(x,t) = \max_{1 \leq i \leq m}\{\tilde{g}_i(x,t)\}$, then the number of inequation is reduced to one, but $\tilde{g}_{\max}(x,t)$ is nonsmooth. We consider to uniformly approximate $\tilde{g}_{\max}(x,t)$ by the smooth spline introduced in [19].

Let us first recall the formulation of multivariate spline. Let D be a polyhedral domain of R^m which is partitioned with irreducible algebraic surfaces into cells $\triangle = \{\triangle_i | i = 1, \ldots, N\}$. A function $s(z)$ defined on D is called a k-spline function with r-th order smoothness, expressed for short as $s(z) \in S_k^r(D, \triangle)$, if $s(z) \in C^r(D)$ and $s(z)|_{\triangle_i} = p_i \in P_k$, where P_k is the set of all polynomial of degree k or less in m variables. Similar to the smooth spline which uniformly approximate $\min\{z_1, z_2, \ldots, z_m\}$ given in [19], we can construct a spline function $s_3^2(z;\varepsilon) \in S_3^2(R^m)$ to uniformly approximates $\max\{z_1, z_2, \ldots, z_m\}$ (as $\varepsilon \to +0$) as follows

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

where $c_1 = 1/(6\varepsilon^2), c_k/c_{k+1} = (k+2)/k$, $1 \le k \le m$, and 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 following composite function uniformly approximates $\max\{\tilde{g}_1(x,t), \tilde{g}_2(x,t), \ldots, \tilde{g}_m(x,t)\}$ as $t \to +0$.

$$\hat{g}(x,t) = s_3^2(\tilde{g}_1(x,t), \tilde{g}_2(x,t), \ldots, \tilde{g}_m(x,t); t). \tag{2.1}$$

For convenience, denote by $\nabla\tilde{g}(x,t)$ the gradient of $\tilde{g}(x,t)$ with respect to the variable x.

To solve the KKT system (1.2), for a randomly chosen $(x^{(0)}, \xi) \in V \times R^n$ and any given $\eta \in R_{++}$, we construct the following constraint shifting spline smoothing homotopy

$$H(w,t) = \begin{pmatrix} (1-t)(\nabla f(x) + \lambda \nabla \hat{g}(x,t)) + \nabla h(x) z + t(x - x^{(0)}) + t(1-t)\xi \\ \lambda \hat{g}(x,t) + t\eta \\ h(x) - t h(x^{(0)}) \end{pmatrix}, \tag{2.2}$$

where $w = (x, \lambda, z) \in W = \tilde{\Omega} \times R_+ \times R^p$, $t \in (0,1]$.

For a given $w^{(0)} \in V \times R_{++} \times R^p$, we rewrite $H(w,t)$ in (2.2) as $H_{w^{(0)}}(w,t)$. Let $H_{w^{(0)}}^{-1}(0) = \{(w,t) \in W \times (0,1] | H_{w^{(0)}}(w,t) = 0\}$.

Proposition 2.2. *Suppose that assumptions (A1), (A2) and (A3) hold, then for $w \in W$, the homotopy equation $H(w,1) = 0$ has only one simple solution $(x, \lambda, z) = w^{(0)} = (x^{(0)}, \lambda^{(0)}, z^{(0)}) = (x^{(0)}, -\hat{g}(x,1)^{-1}\eta, 0)$.*

Proof. By the condition (A1), suppose without loss of generality that $(\hat{x}, \hat{\lambda}, \hat{z})$ solves the equation $H(w,1) = 0$. It follows from $H(\hat{x}, \hat{\lambda}, \hat{z}, 1) = 0$, $\eta \in R_{++}$ and $\lambda \ge 0$ that $\hat{g}(x,1) < 0$. By proposition 3.1 in [4], we know $\tilde{g}(x,1) < 0$, then we have $\hat{x} \in \Omega^{(0)}(1)$. Hence $I(\hat{x},1) = \varnothing$. Let us proceed by contradiction and assume that $\hat{x} \ne x^{(0)}$, which implies that $\hat{z} \ne 0$. By the first equality in (2.2), we have that $x^{(0)} = \hat{x} + \nabla h(\hat{x})\hat{z}$, which contradicts the condition (A3). Hence, $\hat{x} = x^{(0)}$ and $\nabla h(\hat{x})\hat{z} = \nabla h(x^{(0)})\hat{z} = 0$. From the condition (A2), we know that $\hat{z} = 0$. From the second equality in (2.2), we have that $\hat{\lambda} = -\hat{g}(\hat{x},1)^{-1}\eta$. Hence, the solution of $H(w,1) = 0$ is unique. It follows from the nonsingularity of

$$\frac{\partial H(w^{(0)},1)}{\partial w} = \begin{pmatrix} I & 0 & \nabla h(x^{(0)}) \\ \lambda^{(0)} \nabla \hat{g}(x^{(0)},1) & \hat{g}(x^{(0)},1) & 0 \\ (\nabla h(x^{(0)}))^T & 0 & 0 \end{pmatrix}$$

that the solution of $H(w,1) = 0$ is simple. This completes the proof. □

Proposition 2.3. *Suppose that $f(x) \in C^{2,1}$, $g(x) \in C^{2,1}$ and $h(x) \in C^{2,1}$. Let the assumptions (A1) and (A2) hold and H be defined as (2.2), then for almost all $(x^{(0)}, \xi) \in V \times R^n$, 0 is a regular value of $H : W \times (0,1) \to R^{n+1+p}$.*

Proof. Let $\tilde{H}(w, x^{(0)}, \xi, t) : R^n \times R_{++} \times R^p \times V \times R^n \times (0,1) \to R^{n+1+p}$ be the same mapping with $H(w,t)$ but taking also $x^{(0)}$ and ξ as variates. Consider the following submatrix of the Jacobian $D\tilde{H}(w, x^{(0)}, \xi, t)$ of $\tilde{H}(w, x^{(0)}, \xi, t)$:

$$\frac{\partial \tilde{H}(w, x^{(0)}, \xi, t)}{\partial(x^{(0)}, \lambda, \xi)} = \begin{pmatrix} -tI & (1-t)\nabla \hat{g}(x,t) & t(1-t)I \\ 0 & \hat{g}(x,t) & 0 \\ -t(\nabla h(x^{(0)}))^T & 0 & 0 \end{pmatrix},$$

From $t \in (0,1)$, $\eta > 0$ and $\lambda \hat{g}(x,t) + t\eta = 0$, we know that $\hat{g}(x,t) \neq 0$. It follows from (A2) that $(\nabla h(x^{(0)}))^T$ is the matrix of full row rank. These imply that $\frac{\partial \tilde{H}(w, x^{(0)}, \xi, t)}{\partial(x^{(0)}, \lambda, \xi)}$ and $D\tilde{H}(w, x^{(0)}, \xi, t)$ are the matrices of full row rank for any solution of $\tilde{H}(w, x^{(0)}, \xi, t) = 0$ in $R^n \times R_{++} \times R^p \times V \times R^n \times (0,1)$. Hence, 0 is a regular value of $\tilde{H}(w, x^{(0)}, \xi, t)$. From Proposition 3.4 in [4] and the Theorem 2.1, we know that for almost all $(x^{(0)}, \xi) \in V \times R^n$, 0 is a regular value of $H : W \times (0,1) \to R^{n+1+p}$. This completes the proof. □

Theorem 2.4. *Suppose that $f(x) \in C^{2,1}$, $g(x) \in C^{2,1}$ and $h(x) \in C^{2,1}$. Let the assumptions (A1) and (A2) hold and H be defined as (2.2), then for almost all $(x^{(0)}, \xi) \in V \times R^n$, 0 is a regular value of $H : W \times (0,1) \to R^{n+1+p}$. For given $(x^{(0)}, \xi) \in V \times R^n$, if 0 is a regular value of $H : W \times (0,1) \to R^{n+1+p}$, then there must be a smooth curve $\Gamma_{x^{(0)}}$ in $H^{-1}(0)$, starting from $(w^{(0)}, 1)$. In addition, if the assumption (A3) holds, then $\Gamma_{x^{(0)}}$ terminates in or approaches to the hyperplane $t = 0$. If $(x^*, \lambda^*, z^*, 0)$ is a limit point of $\Gamma_{x^{(0)}}$, then x^* is a KKT point of the problem (1.1).*

Proof. From the assumptions (A1), (A2) and Propositions 2.3, we know that for almost all $(x^{(0)}, \xi) \in V \times R^n$, 0 is a regular value of $H : W \times (0,1) \to R^{n+1+p}$. For given $(x^{(0)}, \xi) \in V \times R^n$, if 0 is a regular value of $H : W \times (0,1) \to R^{n+1+p}$, from the fact that $H(w,1) = 0$, the nonsingularity of $\frac{\partial H(w^{(0)},1)}{\partial(w)}$ and the implicit function theorem, we know that $H^{-1}(0)$ contains a smooth curve $\Gamma_{x^{(0)}}$, which starts from $(w^{(0)}, 1)$ and goes into $\Omega^{(0)}(1) \times R_{++} \times R^p \times (0,1)$ and terminates in the boundary of $\Omega(t) \times R_+ \times R^p \times [0,1]$.

Let $(w^*, t_*) = (x^*, \lambda^*, z^*, t_*)$ be an ending limit point of $\Gamma_{x^{(0)}}$. Only the following five cases are possible:

(1) $w^* \in \Omega(1) \times R_+ \times R^p$, $t_* = 1$, $\| (\lambda^*, z^*) \| < \infty$;

(2) $w^* \in \Omega(t_*) \times R_+ \times R^p$, $t_* \in [0,1]$, $\| (\lambda^*, z^*) \| = \infty$;

(3) $w^* \in \Omega(t_*) \times \partial R_+ \times R^p$, $t_* \in (0,1)$, $\| (\lambda^*, z^*) \| < \infty$;

(4) $w^* \in \partial\Omega(t_*) \times R_{++} \times R^p$, $t_* \in (0,1)$, $\| (\lambda^*, z^*) \| < \infty$;

(5) $w^* \in \Omega \times R_+ \times R^p$, $t_* = 0$, $\| (\lambda^*, z^*) \| < \infty$.

It follows from (A1), (A2), (A3) and Proposition 2.2 that $w^{(0)}$ is the unique and simple solution of $H(w,1) = 0$, which implies that case (1) is impossible.

If the case (2) happens, then there exists a sequence of points $\{x^{(k)}, \lambda^{(k)}, z^{(k)}, t_k\}$ on $\Gamma_{x^{(0)}}$ such that $t_k \to t_* \in [0,1]$, $x_k \to x_* \in \Omega(t_*)$, $\| \lambda^{(k)}, z^{(k)} \| \to \infty$, as $k \to \infty$. By the fist equality in the homotopy equation $H(x^{(k)}, \lambda^{(k)}, z^{(k)}, t_{(k)}) = 0$, we have that

$$\begin{aligned} (1-t_k)(\nabla f(x^{(k)}) + \lambda^{(k)} \nabla \hat{g}(x^{(k)}, t_k)) + \nabla h(x^{(k)}) z^{(k)} \\ + t_k(x^{(k)} - x^{(0)}) + t_k(1-t_k)\xi = 0 \end{aligned} \tag{2.3}$$

And only the following two subcases are possible: (a) $t_* = 1$; (b) $t_* \in [0, 1)$.

(a) $t_* = 1$.

If $\| ((1-t_k)\lambda^{(k)}, z^{(k)}) \| < \infty$, suppose without loss of generality that $((1-t_k)\lambda^{(k)}, z^{(k)}) \to (\overline{\lambda}, \overline{z})$, then $\overline{\lambda} > 0$ from the equality $\lambda^{(k)}\hat{g}(x^{(k)}, t_k) + t_k\eta = 0$. By taking the limit in (2.3) we get

$$\begin{aligned} x^{(0)} &= x^{(*)} + \lim_{k\to\infty}[(1-t_k)(\nabla f(x^{(k)}) + \lambda^{(k)}\nabla\hat{g}(x^{(k)}, t_k)) + \nabla h(x^{(k)})z^{(k)}] \\ &= x^{(*)} + \nabla h(x^{(*)})\overline{z} + \lim_{k\to\infty}(1-t_k)\lambda^{(k)}\nabla\hat{g}(x^{(k)}, t_k) \end{aligned}$$

From the proposition 3.3 in [4], we have $\nabla\hat{g}(x^{(k)}, t_k)) = \sum_{i=1}^{m} y_i(x^{(k)}, t_k)\nabla\tilde{g}_i(x^{(k)}, t_k)$ and $y_i(x^{(*)}, 1) = 0$ for $i \notin I(x^{(*)}, 1)$. We have

$$x^{(0)} = x^* + \nabla h(x^*)\overline{z} + \overline{\lambda}\textstyle\sum_{i\in I(x^*,1)} y_i(x^*, 1)\nabla\tilde{g}_i(x^*, 1)$$

which contradicts with the assumption (A3).

For the case $\| ((1-t_k)\lambda^{(k)}, z^{(k)}) \| \to \infty$, the discussion is the same the case (b), which will be done below.

(b) $t_* \in [0, 1)$.

Suppose without loss of generality that $((1-t_k)\lambda^{(k)}, z^{(k)}) / \| ((1-t_k)\lambda^{(k)}, z^{(k)}) \| \to (\overline{\alpha}, \overline{\beta})$ with $\| (\overline{\alpha}, \overline{\beta}) \| = 1$. Divide the both sides of (2.3) by $\| ((1-t_k)\lambda^{(k)}, z^{(k)}) \|$ and take the limit, we have that

$$\overline{\alpha}\nabla\hat{g}(x^*, t_*) + \sum_{i=1}^{l} \overline{\beta}_i \nabla h_i(x^*) = 0$$

From the proposition 3.3 in [4], we have $\nabla\hat{g}(x^*, t_*) = \sum_{i=1}^{m} y_i(x^*, t_*)\nabla\tilde{g}_i(x^*, t_*)$, $y_i(x^*, t_*) = 0$ for $i \notin I(x^*, t_*)$, $y_i(x^*, t_*) \geq 0, i \in I(x^*, t_*)$ and $\sum_{i\in I(x^*,t_*)} y_i(x^*, t_*) = 1$. We have

$$\overline{\alpha} \sum_{i\in I(x^*,t_*)} y_i(x^*, t_*)\nabla\tilde{g}_i(x^*, t_*) + \sum_{i=1}^{l} \overline{\beta}_i \nabla h_i(x^*) = 0$$

which contradicts with the assumption (A2).

From (a) and (b), we conclude that the case (2) is impossible.

From $\lambda^*\hat{g}(x^*, t_*) + t_*\eta = 0$, we know that $t_* > 0$ and $\lambda^* \in \partial R_+$ cannot happen simultaneously, which implies that the case (3) is impossible. If $\lambda^* > 0$ and $t_* > 0$, from $\lambda^*\hat{g}(x^*, t_*) + t_*\eta = 0$, we have $\hat{g}(x^*, t_*) < 0$. By proposition 3.1 in [4], we know $\tilde{g}(x^*, t_*) < 0$, which implies that the case (4) is impossible. As a result, the case (5) is the only possible case. Hence, (x^*, λ^*, z^*) is a solution of the KKT system (1.2). This completes the proof. □

3 The CSSSH-S-N Procedure and Numerical Experiments

3.1 The CSSSH-S-N Procedure

In this section, we give a predictor-corrector algorithm—CSSSH-S-N procedure to trace the path generated by the Constraint Shifting 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 corrctor along with the direction that vertical to the predictor director. At each predictor step and corrector step, we need to check if the computed point is in $\Omega(t)$ or not. If not, we take some damping step. Particularly, in order to improve the efficiency of CSSSH-S-N, the algorithm includes an end game strategy, in which we use the standard Newton's method to solve

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

where t_c is a small positive constant.

For convenience, $\hat{\Omega}_t$ denotes $\Omega(t)\times R_+\times R^p$. From the expression of $\hat{g}(x,t)$, we know when t is smaller, then the number of gradient and Hessian calculations of the inequality constrains is less. In order to improve the efficiency of the CSSSH method, for a given $\theta\in(0,1]$, we use θt instead of t of $\hat{g}(x,t)$. Now we are ready to give a formal statement of our algorithm.

Algorithm 3.1 (the **CSSSH-S-N** procedure)**.**

Step 0: Give a random vector ξ, a positive vector η, $\theta\in(0,1]$, $\theta_1\in(0,1)$, t_{end} and t_c, the starting point $(w^{(0)},1)$, initial step length h_1, step contraction factors $B_{\min}$, step expansion factors $B_{\max}$; tracking tolerances H_{tol} and H_{final} for correction. Let $w^{(k,i)}=w^{(0)},t_{k,i}=1$, goto Step 1 and compute $H'_{w^{(0)}}(w^{(0)},\theta t_0)$. Let $d^{(0)}=(0,\ldots,0,-1)\in R^{n+1+p+1}$, compute the predictor direction d by solving the following system of equation:

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

set $d^{(1)}=\frac{d}{\|d\|}$, $t_0=1$, $k=1$, $N_{good}=2$, goto Step 4.

Step 1 (search the cell): Let $\bar{I}=\{j|\tilde{g}_{\max}(x^{(k,i)},t_{k,i})-\tilde{g}_j(x^{(k,i)},t_{k,i})<\theta t_{k,i}\}$, $\bar{k}$ be the cardinality of $\bar{I}$, and $\bar{I}=\{i_1,i_2,\ldots,i_{\bar{k}}\}$. Range $\{\tilde{g}_{i_j}(x^{(k,i)},t_{k,i})\}_{j=1}^{\bar{k}}$ according to $\tilde{g}_{i_1}(x^{(k,i)},t_{k,i})\geq\tilde{g}_{i_2}(x^{(k,i)},t_{k,i})\geq\cdots\geq\tilde{g}_{i_{\bar{k}}}(x^{(k,i)},t_{k,i})$. If $\bar{k}=1$, the cell is $\triangle_{i_1}(\theta t_{k,i})$. Else, for every $\tilde{k}\in\{\bar{k},\bar{k}-1,\ldots,2\}$, if $(\tilde{k}-1)\tilde{g}_{i_{\tilde{k}}}(x^{(k,i)},t_{k,i})-\sum_{j=1}^{\tilde{k}-1}\tilde{g}_{i_j}(x^{(k,i)},t_{k,i})+\theta t_{k,i}\geq0$, we have $\tilde{k}\in I\subseteq\{\bar{k},\bar{k}-1,\ldots,2\}$. Let $\hat{k}$ be the maximum element of I, then the cell is $\triangle_{i_1\ldots i_{\hat{k}}}(\theta t_{k,i})$.

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 $\beta^{(k)}=\arccos((d^{(k)})^Td^{(k-1)})$.

Step 3: If corrector step fail or $\beta^{(k)}>\pi/4$, set $h_k=B_{\min}(1)h_{k-1}$, $N_{good}=0$; If $i\geq5$, 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 $i\leq2$, set $N_{good}=N_{good}+1$, if $N_{good}>2$, set $h_k=\min(1,B_{\max}(1)h_{k-1})$.

Step 4: If $h_k<10^{-20}$, stop the algorithm with an error flag; else determine the smallest nonnegative integer j such that $(w^{(k,0)},t_{k,0})=(w^{(k-1)},t_{k-1})+\theta_1^jh_kd^{(k)}\in\hat{\Omega}_{t_{k,0}}\times(0,1]$, set $h_k=\theta_1^jh_k$, $i=0$.

Step 5: If $t_{k,0} \le t_{end}$, 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 goto Step 1, compute $H'_{w^{(0)}}(w^{(k,i)}, \theta t_{k,i})$ and then compute the corrector direction $d^{(k,i+1)}$ by

$$\begin{pmatrix} H'_{w^{(0)}}(w^{(k,i)}, \theta t_{k,i}) \\ {d^k}^T \end{pmatrix} d^{(k,i+1)} = \begin{pmatrix} -H_{w^{(0)}}(w^{(k,i)}, \theta t_{k,i}) \\ 0 \end{pmatrix},$$

determine the smallest nonnegative integer j such that

$$(w^{(k,i+1)}, t_{k,i+1}) = (w^{(k,i)}, t_{k,i}) + \theta_1^j d^{(k,i+1)} \in \hat{\Omega}_{t_{k,i+1}} \times (0,1),$$

replace i by $i+1$.

Step 7: If $t_{k,i} \ge 1$ 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 $\|H_{w^{(0)}}(w^{(k,i)}, \theta t_{k,i})\|_{\inf} > H_{tol}$, 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): Let $w^{(k,i)} = w^{(k,l)}, t_{k,i} = t_c$, goto Step 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 $t_{end} = 0.3 * t_{end}$, $w^{(k)} = w^{(k-1)}$, $t_k = t_{k-1}$, $d^{(k+1)} = d^{(k)}$, replace k by $k+1$, goto Step 3; else goto step 9.

3.2 Numerical Experiment

We have implemented the CSSSH-S-N algorithm using the MATLAB. In order to show the efficiency of the algorithm, we also have implemented CHIP and CSH methods using similar procedures. We compare these algorithms with KNITRO [14] 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 it's parameters as default values.

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:

- Set $\tilde{g}(x,t) = g(x) - t^2\beta e$, $\beta = 0$ when $g(x^{(0)}) < 0$, $\beta = 10 + \max_i g_i(x^{(0)})$ when $g(x^{(0)}) \nless 0$, $\eta = 10e$;
- Parameter $\theta = 0.0001$, $\theta_1 = 0.9$;

· Parameters in end game section $t_c = 10^{-6}$, $t_{end} = 0.1$;

· Step size parameters $h_0 = 0.1$, $B_{\min} = [0.5, 0.75]$, $B_{\max} = [3, 1.5]$;

· Tracking tolerances $H_{tol} = 10^{-3}$, $H_{final} = 10^{-12}$;

For different problems, we list the final approximate solution x^*, the objective function $f(x^*)$ and CPU time in seconds. 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^{-20} before $t = 0$, this is generally due to poor conditioned Jacobian matrix. The notation "fail^2" means $x^{(0)}$ is not feasible interior point for CHIP method. The notation "fail^3" means out of memory. The notation "fail^4" means no result in 5000 Newton iterations or 7200 seconds.

Example 3.1.

$$
\begin{aligned}
f(x) &= \sin(x_1 - 1 + 1.5\pi) \\
&\quad + \sum_{2 \le i \le 100} 100 \sin(-x_i + 1.5\pi + x_{i-1}^2), \\
g_i(x) &= \begin{cases} x_1 - \pi & i = 1; \\ x_{i-1}^2 - x_i - \pi & i = 2, \dots, 100; \\ -x_1 - \pi & i = 101; \\ -x_{i-1}^2 + x_i - \pi & i = 102, \dots, 200, \end{cases} \\
h_i(x) &= x_i - x_{i+1} \quad i = 1, \dots, 99.
\end{aligned}
$$

TABLE 1: Test result for Example 3.1.

x^0	Method	$f(x^*)$	x^*	Time
(0.6,0.6,...,0.6)	CSH	-	-	fail^1
	CHIP	-9901	$(1.0, \cdots, 1.0)$	4.4899
	KNITRO	-9901	$(1.0, \cdots, 1.0)$	0.3440
	CSSSH	-9901	$(1.0, \cdots, 1.0)$	3.5607
$x_1 = x_{10} = x_{20} = x_{30} = 0.9$	CSH	-9901	$(1.0, \cdots, 1.0)$	11.5612
$x_i = 1, i \in \{2, \dots, 9, 11, \dots,$	CHIP	-	-	fail^2
$19, 21, \dots, 29, 31, \dots, 100\}$	KNITRO	-9901	$(1.0, \cdots, 1.0)$	1.250
	CSSSH	-9901	$(1.0, \cdots, 1.0)$	4.8499

Example 3.2.

$$
\begin{aligned}
f(x) &= x_3^2 + x_4^2, \\
g_{i,j}(x) &= (t_i - x_1)^2 / x_3^2 + (t'_j - x_2)^2 / x_4^2 - 1, \\
t_i &= i / (\sqrt{m} - 1), \ i = 0, \dots, \sqrt{m} - 1, \\
t'_j &= j / (\sqrt{m} - 1), \ j = 0, \dots, \sqrt{m} - 1. \\
h_1(x) &= x_1 - x_2, \\
h_2(x) &= x_3 - x_4.
\end{aligned}
$$

TABLE 2.1: Test result for Example 3.2 with $x^{(0)} = (0, 0, 100, 100)$.

m	Method	$f(x^*)$	x^*	Time
100	CSH	0.9999	(0.5000,0.5000,0.7071,0.7071)	0.5616
	CHIP	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.5640
	KNITRO	1.0000	(0.5000,0.5000,-0.7071,-0.7071)	0.0310
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.4938
400	CSH	0.9999	(0.5000,0.5000,0.7071,0.7071)	1.5540
	CHIP	1.0000	(0.5000,0.5000,0.7071,0.7071)	1.1906
	KNITRO	1.0000	(0.5000,0.5000,-0.7071,-0.7071)	0.0940
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.6075
900	CSH	0.9999	(0.5000,0.5000,0.7071,0.7071)	3.6011
	CHIP	1.0000	(0.5000,0.5000,0.7071,0.7071)	2.6922
	KNITRO	1.0000	(0.5000,0.5000,-0.7071,-0.7071)	0.1410
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.6875
10000	CSH	-	-	fail[1]
	CHIP	1.0000	(0.5000,0.5000,0.7071,0.7071)	55.6361
	KNITRO	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.9530
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.7334
90000	CSH	-	-	fail[1]
	CHIP	1.0000	(0.5000,0.5000,0.7071,0.7071)	5606.9141
	KNITRO	1.0000	(0.5000,0.5000,-0.7071,-0.7071)	25.6720
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	2.6480

TABLE 2.2: Test result for Example 3.2 with $x^{(0)} = (10, 9, 90, 85)$.

m	Method	$f(x^*)$	x^*	Time
100	CSH	0.9999	(0.5000,0.5000,0.7071,0.7071)	0.6755
	CHIP	-	-	fail[2]
	KNITRO	1.0000	(0.5000,0.5000,-0.7071,-0.7071)	0.0470
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.5252
400	CSH	0.9999	(0.5000,0.5000,0.7071,0.7071)	1.4476
	CHIP	-	-	fail[2]
	KNITRO	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.1090
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.5866
900	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	1.0000	(0.5000,0.5000,-0.7071,-0.7071)	0.1410
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	0.5943
10000	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	1.0000	(0.5000,0.5000,0.7071,0.7071)	7.4840
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	1.4217
90000	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	1.0000	(0.5000,0.5000,-0.7071,-0.7071)	75.0310
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	2.5076

Example 3.3.

$$\begin{aligned}
&f(x) = x_1^2/3 + x_1/2 + x_2^2,\\
&g_i(x) = (1 - x_1^2 t_i^2)^2 - x_1 t_i^2 - x_2^2 + x_2,\\
&t_i = i/(m-1), i = 0, \ldots, m-1.\\
&h_1(x) = x_1 + 0.75.
\end{aligned}$$

TABLE 3.1: Test result for Example 3.3 with $x^{(0)} = (-0.75, 100)$.

m	Method	$f(x^*)$	x^*	Time
100	CSH	-	-	fail[1]
	CHIP	-	-	fail[1]
	KNITRO	2.4305	(-0.7500,1.6180)	0.0930
	CSSSH	2.4305	(-0.7500,1.6180)	0.1548
1000	CSH	-	-	fail[1]
	CHIP	-	-	fail[1]
	KNITRO	2.4305	(-0.7500,1.6180)	0.0940
	CSSSH	2.4305	(-0.7500,1.6180)	0.1810
10000	CSH	-	-	fail[1]
	CHIP	-	-	fail[4]
	KNITRO	2.4305	(-0.7500,1.6180)	0.5940
	CSSSH	2.4305	(-0.7500,1.6180)	0.3693
100000	CSH	-	-	fail[4]
	CHIP	-	-	fail[4]
	KNITRO	2.4305	(-0.7500,1.6180)	3.3280
	CSSSH	2.4305	(-0.7500,1.6180)	2.6806
1000000	CSH	-	-	fail[4]
	CHIP	-	-	fail[4]
	KNITRO	-	-	fail[3]
	CSSSH	1.0000	(0.5000,0.5000,0.7071,0.7071)	36.9158

TABLE 3.2: Test result for Example 3.3 with $x^{(0)} = (-1, 20)$.

m	Method	$f(x^*)$	x^*	Time
100	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	2.4305	(-0.7500,1.6180)	0.0630
	CSSSH	2.4305	(-0.7500,1.6180)	0.0499
1000	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	2.4305	(-0.7500,1.6180)	0.0780
	CSSSH	2.4305	(-0.7500,1.6180)	0.5866
10000	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	2.4305	(-0.7500,1.6180)	0.5310
	CSSSH	2.4305	(-0.7500,1.6180)	0.1046
100000	CSH	-	-	fail[4]
	CHIP	-	-	fail[2]
	KNITRO	2.4305	(-0.7500,1.6180)	2.4840
	CSSSH	2.4305	(-0.7500,1.6180)	0.6515
1000000	CSH	-	-	fail[4]
	CHIP	-	-	fail[2]
	KNITRO	-	-	fail[3]
	CSSSH	2.4305	(-0.7500,1.6180)	8.1487

Example 3.4.

$$
\begin{aligned}
f(x) &= \frac{1}{n} \sum_{1 \le k \le n} (x_k - 1)^2, \\
g_i(x) &= \prod_{1 \le k \le n} \cos(t_i x_k) + t_i \sum_{1 \le k \le n} x_k^3, \\
t_i &= 0.5 + \pi i/(m-1), i = 0, \ldots, m-1, \\
h_i(x) &= x_i - x_{i+1}, i = 0, \ldots, n-1.
\end{aligned}
$$

TABLE 4.1: Test result for Example 3.4 with $n = 100$, $x^{(0)} = (-0.5, \cdots, -0.5)$.

m	Method	$f(x^*)$	x^*	Time
100	CSH	-	-	fail[1]
	CHIP	1.4914	(-0.2212,$\cdots$,-0.2212)	54.2700
	KNITRO	1.4915	(-0.2213,$\cdots$,-0.2213)	1.0160
	CSSSH	1.4914	(-0.2212,$\cdots$,-0.2212)	1.1826
1000	CSH	-	-	fail[1]
	CHIP	1.4914	(-0.2212,$\cdots$,-0.2212)	1123.2969
	KNITRO	1.4915	(-0.2213,$\cdots$,-0.2213)	5.6560
	CSSSH	1.4914	(-0.2212,$\cdots$,-0.2212)	3.0975
10000	CSH	-	-	fail[3]
	CHIP	-	-	fail[3]
	KNITRO	-	-	fail[3]
	CSSSH	1.4914	(-0.2212,$\cdots$,-0.2212)	8.4600
100000	CSH	-	-	fail[3]
	CHIP	-	-	fail[3]
	KNITRO	-	-	fail[3]
	CSSSH	1.4914	(-0.2212,$\cdots$,-0.2212)	60.2280

TABLE 4.2: Test result for Example 3.4 with $n = 100$,
$x^{(0)} = (0.9, 0.8, 0.7, 0.6, 0.5, 0.4, -2, \cdots, -2)$.

m	Method	$f(x^*)$	x^*	Time
100	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	1.4915	(-0.2213,$\cdots$,-0.2213)	1.4530
	CSSSH	1.4914	(-0.2212,$\cdots$,-0.2212)	2.0319
1000	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	1.4915	(-0.2213,$\cdots$,-0.2213)	8.4840
	CSSSH	1.4914	(-0.2212,$\cdots$,-0.2212)	2.7178
10000	CSH	-	-	fail[3]
	CHIP	-	-	fail[2]
	KNITRO	-	-	fail[3]
	CSSSH	1.4914	(-0.2212,$\cdots$,-0.2212)	8.2156
100000	CSH	-	-	fail[3]
	CHIP	-	-	fail[2]
	KNITRO	-	-	fail[3]
	CSSSH	1.4914	(-0.2212,$\cdots$,-0.2212)	96.7384

TABLE 4.3: Test result for Example 3.4 with $m = 1000$, $x^{(0)} = (-0.5, \cdots, -0.5)$.

n	Method	$f(x^*)$	x^*	Time
150	CSH	-	-	fail[1]
	CHIP	1.4147	(-0.1894,$\cdots$, -0.1894)	2832.2024
	KNITRO	1.4147	(-0.1894,$\cdots$, -0.1894)	12.0000
	CSSSH	1.4147	(-0.1894,$\cdots$, -0.1894)	4.8622
200	CSH	-	-	fail[1]
	CHIP	1.3677	(-0.1695,$\cdots$,-0.1695)	5704.1973
	KNITRO	1.3678	(-0.1695,$\cdots$,-0.1695)	38.9380
	CSSSH	1.3677	(-0.1695,$\cdots$,-0.1695)	11.4528
250	CSH	-	-	fail[3]
	CHIP	-	-	fail[3]
	KNITRO	1.3351	(-0.1555,$\cdots$,-0.1555)	56.2500
	CSSSH	1.3350	(-0.1554,$\cdots$,-0.1554)	26.1072
300	CSH	-	-	fail[3]
	CHIP	-	-	fail[3]
	KNITRO	1.3105	(-0.1448,$\cdots$,-0.1448)	88.1090
	CSSSH	1.3105	(-0.1447,$\cdots$,-0.1447)	35.5979

TABLE 4.4: Test result for Example 3.4 with $m = 1000$, $x^{(0)} = (0.9, 0.8, 0.7, 0.6, 0.5, 0.4, -2, \cdots, -2)$.

n	Method	$f(x^*)$	x^*	Time
150	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	1.4147	(-0.1894,$\cdots$, -0.1894)	19.0160
	CSSSH	1.4147	(-0.1894,$\cdots$, -0.1894)	8.4100
200	CSH	-	-	fail[1]
	CHIP	-	-	fail[2]
	KNITRO	1.3678	(-0.1695,$\cdots$,-0.1695)	33.1720
	CSSSH	1.3677	(-0.1695,$\cdots$,-0.1695)	14.7899
250	CSH	-	-	fail[3]
	CHIP	-	-	fail[2]
	KNITRO	1.3351	(-0.1555,$\cdots$,-0.1555)	51.0630
	CSSSH	1.3350	(-0.1554,$\cdots$,-0.1554)	26.1072
300	CSH	-	-	fail[3]
	CHIP	-	-	fail[2]
	KNITRO	1.3105	(-0.1448,$\cdots$,-0.1448)	87.0630
	CSSSH	1.3105	(-0.1447,$\cdots$,-0.1447)	55.5347

3.3 Remarks

Now we give some remarks on numerical results.

1. For nonlinear programming with large number of complicated inequality constraints and equality constraints, the CSSSH method can save much computation of the gradient and the Hessian of inequality constraint functions and hence it is more efficient than the CHIP and CSH method. Even compare with very successful optimization softwares like KNITRO, the CSSSH method is very encouraging by above preliminary numerical tests. It is our future work to test examples with application's background.

2. In example 3.1, the inequality constraints are linear or quadratic functions, their gradient and Hessian calculations are cheap. So the efficiency of CSSSH-S-N is lower than KNITRO.

3. The Algorithm 3.1 is a simple implementation of the CSSSH method. Our numerical tests compare it fairly with similar simple implementations of CHIP method and CSH method. It needs to do much work to improve implementation of the CSSSH method on all processes of numerical path tracing, say, schemes of predictor and corrector, step length updating, linear system solving, end game. Other practical strategies like sparsity exploitation and memory allocation, e.g., [2,3,7,8,13], are also very important for improving the efficiency.

References

[1] R. Abraham and J. Robbin, *Transversal Mappings and Flows*, Benjamin, New York, 1967.

[2] R. Byrd, M. Hribar and J. Nocedal, An interior point algorithm for large-scale nonlinear programming, *SIAM J. Optim.* 9 (1999) 877–900.

[3] A. Conn, N. Gould and P.L. Toint, LANCELOT: A Fortran package for large-scale nonlinear optimization (release A), Springer Series in Computational Mathematics, vol. 17. Springer-Verlag, New York, 1992.

[4] L. Dong, B. Yu and G.H. Zhao, A smoothing spline homotopy method for nonconvex nonlinear programming, submitted, (2013).

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

[6] G.C. Feng and B. Yu, 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)*, Lecture Notes Numer. Appl. Anal., Kinokuniya, Tokyo, vol. 14 1995, pp. 9–16.

[7] R. Fletcher and S. Leyffer, Nonlinear programming without a penalty function, *Math. Program. Ser. A.* 91 (2002) 239–269.

[8] P. Gill, W. Murray and M. Saunders, SNOPT: an SQP algorithm for large-scale constrained optimization, *SIAM J. Optim.* 12 (2002) 979–1006.

[9] Z.H. Lin, Y. Li and B. Yu, A combined homotopy interior point method for general nonlinear programming problems, *Appl. Math. Comput.* 80 (1996) 209–224.

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

[11] Q.H. Liu, B. Yu and G.C. Feng, An interior point path following method for nonconvex nonlinear programming problem with quasi normal cone condition, *Adv. Math.* 19(4) (2000) 281–282.

[12] Y.F. Shang and B. Yu, Boundary moving combined homotopy method for nonconvex nonlinear programming and its convergence, *J. Jilin Univ. Sci.* 44 (2006) 357–361.

[13] R. Vanderbei and D. Shanno, An interior-point algorithm for nonconvex nonlinear programming, *Comput. Optim. Appl.* 13 (1999) 231–252.

[14] R.A. Waltz and T.D. Plantenga Ziena Optimization, Inc. www.ziena.com (2010).

[15] L. Yang, B. Yu and Q. Xu, *A Combined homotopy infeasible interior point method for nonconvex programming*, *Pacific J. Optim.* 8 (2012) 89–101.

[16] L. Yang, B. Yu and Q. Xu, A constraint shifting homotopy method for general nonlinear programming, *Optimization* DOI: 10.1080/02331934.2012.668189.

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

[18] B. Yu, Q.H. Liu and G.C. Feng, A combined homotopy interior point method for nonconvex programming with pseudo cone condition, *Northeast, Math. J.* 16 (2000) 383–386.

[19] G.H. Zhao, Z.R. Wang and H.N. Mou, Uniform Approximation of Min/Max Functions by Smooth Splines, *J. Com. Appl. Math.* 236 (2011) 699–703.

Manuscript received 30 March 2013
revised 9 September 2013
accepted for publication 13 October 2013

Li Dong
School of Mathematical Sciences, Dalian University of Technology
Dalian, 116024, P. R. China
College of Science, Dalian Nationalities University
Dalian, 116600, P. R. China
E-mail address: dongli@dlnu.edu.cn

Bo Yu
School of Mathematical Sciences, Dalian University of Technology
Dalian, 116024, PR China
School of Science, Dalian University of Technology
Dalian at Panjin, 124221, P. R. China
E-mail address: yubo@dlut.edu.cn

Zhengyong Zhou
School of Mathematics and Computer Sciences, Shanxi Normal University
Linfen, Shanxi 041004, P. R. China
E-mail address: zzy198300@163.com