Hindawi Publishing Corporation
Advances in Operations Research
Volume 2013, Article ID 513918, 7 pages
http://dx.doi.org/10.1155/2013/513918

Research Article

A Newton-Type Algorithm for Solving Problems of Search Theory

Liping Zhang

Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Liping Zhang; lzhang@math.tsinghua.edu.cn

Received 9 March 2012; Accepted 15 December 2012

Academic Editor: Chandal Nahak

In the survey of the continuous nonlinear resource allocation problem, Patriksson pointed out that Newton-type algorithms have not been proposed for solving the problem of search theory in the theoretical perspective. In this paper, we propose a Newton-type algorithm to solve the problem. We prove that the proposed algorithm has global and superlinear convergence. Some numerical results indicate that the proposed algorithm is promising.

1. Introduction

We consider the problem

$$\max_{\mathbf{x}\in X} \sum_{i=1}^{n} \mathbf{a}_i \left(1 - \exp\left(-\mathbf{b}_i \mathbf{x}_i\right)\right), \tag{1}$$

where $X = \{\mathbf{x} \in \mathbb{R}^n_+ \mid \mathbf{e}^T\mathbf{x} = c\}$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n_{++}$, $c > 0$, and $\mathbf{e} \in \mathbb{R}^n$ is the vector of ones. The problem described by (1) is called the *theory of search* by Koopman [1] and Patriksson [2]. It has the following interpretation: an object is inside box i with probability $\mathbf{a}_i$, and $-\mathbf{b}_i$ is proportional to the difficulty of searching inside the box. If the searcher spends $\mathbf{x}_i$ time units looking inside box i, then he/she will find the object with probability $1 - \exp(-\mathbf{b}_i\mathbf{x}_i)$. The problem described by (1) represents the optimum search strategy if the available time is limited to c time units. Such problems in the form of (1) arise, for example, in searching for a lost object, in distribution of destructive effort such as a weapons allocation problem [3], in drilling for oil, and so forth [2]. Patriksson [2] surveyed the history and applications as well as algorithms of Problem (1); see [2, Sections 2.1.4, 2.1.5, and 3.1.2]. Patriksson pointed out that Newton-type algorithms have not been theoretically analyzed for the problem described by (1) in the references listed in [2].

Recently, related problems and methods were considered in many articles, for example, [4–6]. For example, a projected pegging algorithm was proposed in [5] for solving convex quadratic minimization. However, the question proposed by Patriksson [2] was not answered in the literature. In this paper, we design a Newton-type algorithm to solve the problem described by (1). We show that the proposed algorithm has global and superlinear convergence.

According to the Fischer-Burmeister function [7], the problem described by (1) can be transformed to a semismooth equation. Based on the framework of the algorithms in [8, 9], a smoothing Newton-type algorithm is proposed to solve the semismooth equation. It is shown that the proposed algorithm can generate a bounded iteration sequence. Moreover, the iteration sequence superlinearly converges to an accumulation point which is a solution to the problem described by (1). Numerical results indicate that the proposed algorithm has good performance even for $n = 10000$.

The rest of this paper is organized as follows. The Newton-type algorithm is proposed in Section 2. The global and superlinear convergence is established in Section 3. Section 4 reports some numerical results. Finally, Section 5 gives some concluding remarks.

The following notation will be used throughout this paper. All vectors are column ones, the subscript T denotes transpose, $\mathbb{R}^n$ (resp., $\mathbb{R}$) denotes the space of n-dimensional real column vectors (resp., real numbers), and $\mathbb{R}^n_+$ and $\mathbb{R}_{++}$ denote the nonnegative and positive orthants of $\mathbb{R}^n$ and $\mathbb{R}$, respectively. Let Φ' denote the derivative of the function Φ. We define $N := \{1, 2, \ldots, n\}$. For any vector $\mathbf{x} \in \mathbb{R}^n$, we denote

by $\mathrm{diag}\{\mathbf{x}_i : i \in N\}$ the diagonal matrix whose ith diagonal element is $\mathbf{x}_i$ and $\mathrm{vec}\{\mathbf{x}_i : i \in N\}$ the vector $\mathbf{x}$. The symbol $\|\cdot\|$ stands for the 2-norm. We denote by S the solution set of Problem (1). For any $\alpha, \beta > 0$, $\alpha = O(\beta)$ (resp., $\alpha = o(\beta)$) means α/β is uniformly bounded (resp., tends to zero) as $\beta \to 0$.

2. Algorithm Description

In this section, we formulate the problem described by (1) as a semismooth equation and develop a smoothing Newton-type algorithm to solve the semismooth equation.

We first briefly recall the concepts of NCP, semismooth and smoothing functions [10–12].

Definition 1. A function $\phi : \mathbb{R}^2 \to \mathbb{R}$ is called an NCP function if

$$\phi(u, v) = 0 \Longleftrightarrow u \geq 0, \quad v \geq 0, \; uv = 0. \tag{2}$$

Definition 2. A locally Lipschitz function $F : \mathbb{R}^n \to \mathbb{R}^m$ is called semismooth at $\mathbf{x} \in \mathbb{R}^n$ if F is directionally differentiable at $\mathbf{x}$ and for all $Q \in \partial F(\mathbf{x} + \mathbf{d})$ and $\mathbf{d} \to \mathbf{0}$,

$$F(\mathbf{x} + \mathbf{d}) - F(\mathbf{x}) - Q\mathbf{d} = o(\|\mathbf{d}\|), \tag{3}$$

where ∂F is the generalized Jacobian of F in the sense of Clarke [13].

F is called strongly semismooth at $\mathbf{x} \in \mathbb{R}^n$ if F is semismooth at $\mathbf{x}$ and for all $Q \in \partial F(\mathbf{x} + \mathbf{d})$ and $\mathbf{d} \to \mathbf{0}$,

$$F(\mathbf{x} + \mathbf{d}) - F(\mathbf{x}) - Q\mathbf{d} = O\left(\|\mathbf{d}\|^2\right). \tag{4}$$

Definition 3. Let $\mu \neq 0$ be a parameter. Function $F_\mu(\mathbf{x})$ is called a smoothing function of a semismooth function $F(\mathbf{x})$ if it is continuously differentiable everywhere and there is a constant $\widetilde{c} > 0$ independent of μ such that

$$\left\|F_\mu(\mathbf{x}) - F(\mathbf{x})\right\| \leq \widetilde{c}\mu, \quad \forall \mathbf{x}. \tag{5}$$

The Fischer-Burmeister function [7] is one of the well-known NCP functions:

$$\phi(u, v) = u + v - \sqrt{u^2 + v^2}. \tag{6}$$

Clearly, the Fischer-Burmeister function defined by (6) is not smooth, but it is strongly semismooth [14]. Let $\varphi : \mathbb{R}^3 \to \mathbb{R}$ be the perturbed Fischer-Burmeister function defined by

$$\varphi(\mu, u, v) = u + v - \sqrt{u^2 + v^2 + \mu^2}. \tag{7}$$

It is obvious that for any $\mu > 0$, φ is differentiable everywhere and for each $\mu \geq 0$, we have

$$\left|\varphi(\mu, u, v) - \phi(u, v)\right| \leq \mu, \quad \forall (u, v) \in \mathbb{R}^2. \tag{8}$$

In particular, $\varphi(0, u, v) = \phi(u, v)$ for all $(u, v) \in \mathbb{R}^2$. Namely, φ defined by (7) is a smoothing function of ϕ defined by (6).

According to Kuhn-Tucker theorem, the problem described by (1) can be transformed to

$$\begin{gathered} \mathbf{e}^T\mathbf{x} = c, \\ \mathbf{x}_i \geq 0, \quad s - \mathbf{a}_i\mathbf{b}_i \exp\left(-\mathbf{b}_i\mathbf{x}_i\right) \geq 0, \quad \forall i \in N, \\ \mathbf{x}_i\left(s - \mathbf{a}_i\mathbf{b}_i \exp\left(-\mathbf{b}_i\mathbf{x}_i\right)\right) = 0, \quad \forall i \in N, \end{gathered} \tag{9}$$

where $s \in \mathbb{R}$.

Define

$$\begin{gathered} \Psi(s, \mathbf{x}) := \begin{pmatrix} \vdots \\ \phi\left(\mathbf{x}_i, s - \mathbf{a}_i\mathbf{b}_i \exp\left(-\mathbf{b}_i\mathbf{x}_i\right)\right) \\ \vdots \end{pmatrix}_{i \in N}, \\ H(s, \mathbf{x}) := \begin{pmatrix} \mathbf{e}^T\mathbf{x} - c \\ \Psi(s, \mathbf{x}) \end{pmatrix}. \end{gathered} \tag{10}$$

According to the Fischer-Burmeister function defined by (6), we formulate (9) as the following semismooth equation:

$$H(s, \mathbf{x}) = 0. \tag{11}$$

Based on the perturbed Fischer-Burmeister function defined by (7), we obtain the following smooth equation:

$$G(\mathbf{y}) := G(\mu, s, \mathbf{x}) := \begin{pmatrix} \mu \\ \mathbf{e}^T\mathbf{x} - c \\ \Phi(\mu, s, \mathbf{x}) \end{pmatrix} = 0, \tag{12}$$

where

$$\Phi(\mu, s, \mathbf{x}) := \begin{pmatrix} \vdots \\ \varphi(\mu, \mathbf{x}_i, s - \mathbf{a}_i\mathbf{b}_i \exp(-\mathbf{b}_i\mathbf{x}_i)) \\ \vdots \end{pmatrix}_{i \in N}. \tag{13}$$

Clearly, if $\mathbf{y}^* = (0, s^*, \mathbf{x}^*)$ is a solution to (12) then $\mathbf{x}^*$ is an optimal solution to the problem described by (1).

We give some properties of the function G in the following lemma, which will be used in the sequel.

Lemma 4. *Let G be defined by* (12). *Then G is semismooth on $\mathbb{R}^{n+2}$ and continuously differentiable at any $\mathbf{y} = (\mu, s, \mathbf{x}) \in \mathbb{R}_{++} \times \mathbb{R}^{n+1}$ with its Jacobian*

$$G'(\mathbf{y}) = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 0 & \mathbf{e}^T \\ \Phi'_\mu & \Phi'_s & \Phi'_{\mathbf{x}} \end{pmatrix}, \tag{14}$$

where

$$\begin{gathered} \Phi'_\mu := \mathrm{vec}\left\{-\frac{\mu}{w_i} : i \in N\right\}, \\ \Phi'_s := \mathrm{vec}\left\{1 - \frac{\left(s - \mathbf{a}_i\mathbf{b}_i \exp\left(-\mathbf{b}_i\mathbf{x}_i\right)\right)}{w_i} : \; i \in N\right\}, \\ \Phi'_{\mathbf{x}} := \mathrm{diag}\left\{\left(1 - \frac{\mathbf{x}_i}{w_i}\right) + \mathbf{a}_i\mathbf{b}_i^2 \exp\left(-\mathbf{b}_i\mathbf{x}_i\right)\left(\Phi'_s\right)_i : i \in N\right\}, \end{gathered} \tag{15}$$

with $w_i := \sqrt{(\mathbf{x}_i)^2 + (s - \mathbf{a}_i\mathbf{b}_i \exp(-\mathbf{b}_i\mathbf{x}_i))^2 + \mu^2}$, $i \in N$. *Moreover, the matrix* $G'(\mathbf{y})$ *is nonsingular on* $\mathbb{R}_{++} \times \mathbb{R}^{n+1}$.

Proof. G is semismooth on $\mathbb{R}^{n+2}$ due to the strong semismoothness of $\phi(u, v)$. G is continuously differentiable on $\mathbb{R}_{++} \times \mathbb{R}^{n+1}$. For any $\mu > 0$, (14) can be obtained by a straightforward calculation from (12). Clearly, we have for any $\mu > 0$,

$$0 < 1 - \frac{\mathbf{x}_i}{w_i} < 2, \quad 0 < \left(\Phi'_s\right)_i < 2, \\ \mathbf{a}_i\mathbf{b}_i^2 \exp\left(-\mathbf{b}_i\mathbf{x}_i\right) > 0, \quad \forall i \in N, \tag{16}$$

which implies that $(\Phi'_{\mathbf{x}})_{ii} > 0$ for all $i \in N$. Let

$$\begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 0 & \mathbf{e}^T \\ \Phi'_\mu & \Phi'_s & \Phi'_{\mathbf{x}} \end{pmatrix} \begin{pmatrix} u \\ v \\ \mathbf{z} \end{pmatrix} = 0. \tag{17}$$

Then, we have $u = 0$ and

$$\sum_{i=1}^{n} \mathbf{z}_i = 0, \quad (\Phi'_s)_i v + \left(\Phi'_{\mathbf{x}}\right)_{ii} \mathbf{z}_i = 0, \quad \forall i \in N. \tag{18}$$

The second equality in (18) implies

$$\mathbf{z}_i = \frac{-\left(\Phi'_s\right)_i}{\left(\Phi'_{\mathbf{x}}\right)_{ii}} v, \quad i \in N. \tag{19}$$

Since $(\Phi'_s)_i > 0$ and $(\Phi'_{\mathbf{x}})_{ii} > 0$ for $i \in N$, the first equality in (18) yields $v = 0$ and hence $\mathbf{z} = \mathbf{0}$. Therefore, the matrix $G'(\mathbf{y})$ defined by (14) is nonsingular for $\mu > 0$. □

We now propose a smoothing Newton-type algorithm for solving the smooth equation in (12). It is a modified version of the smoothing Newton method proposed in [8]. The main difference is that we add a different perturbed item in Newton equation, which allows the algorithm to generate a bounded iteration sequence. Let $\mathbf{y} = (\mu, s, \mathbf{x}) \in \mathbb{R}^{n+2}$ and $\gamma \in (0, 1)$. Define a function $\rho : \mathbb{R}^{n+2} \to \mathbb{R}_+$ by

$$\rho(\mathbf{y}) := \gamma \|G(\mathbf{y})\| \min\{1, \|G(\mathbf{y})\|\}. \tag{20}$$

Algorithm 5

Step 0. Choose $\delta, \sigma \in (0, 1)$ and $\mu^0 > 0$. Let $\overline{\mathbf{u}} := (\mu^0, 0, \mathbf{0}) \in \mathbb{R}_{++} \times \mathbb{R} \times \mathbb{R}^n$. Let $s^0 \in \mathbb{R}$ and $\mathbf{x}^0 \in \mathbb{R}^n$ be arbitrary points. Let $\mathbf{y}^0 := (\mu^0, s^0, \mathbf{x}^0)$. Choose $\gamma \in (0, 1)$ such that $\gamma\|G(\mathbf{y}^0)\| < 1$ and $\gamma\mu^0 < 1$. Set $k := 0$.

Step 1. If $G(\mathbf{y}^k) = 0$, stop. Otherwise, let $\rho_k := \rho(\mathbf{y}^k)$.

Step 2. Compute $\Delta\mathbf{y}^k := (\Delta\mu^k, \Delta s^k, \Delta\mathbf{x}^k) \in \mathbb{R}^{n+2}$ by

$$G\left(\mathbf{y}^k\right) + G'\left(\mathbf{y}^k\right)\Delta\mathbf{y}^k = \rho_k\overline{\mathbf{u}}. \tag{21}$$

Step 3. Let m_k be the smallest nonnegative integer such that

$$\left\|G\left(\mathbf{y}^k + \delta^{m_k}\Delta\mathbf{y}^k\right)\right\| \le \left[1 - \sigma\left(1 - \gamma\mu^0\right)\delta^{m_k}\right]\left\|G\left(\mathbf{y}^k\right)\right\|. \tag{22}$$

Let $\lambda_k := \delta^{m_k}$.

Step 4. Set $\mathbf{y}^{k+1} := \mathbf{y}^k + \lambda_k\Delta\mathbf{y}^k$ and $k := k + 1$. Go to Step 1.

The following theorem proves that Algorithm 5 is well defined.

Theorem 5. *Algorithm 5 is well defined. If it finitely terminates at* k*th iteration then* $\mathbf{x}^k$ *is an optimal solution to the problem described by* (1). *Otherwise, it generates an infinite sequence* $\{\mathbf{y}^k = (\mu^k, s^k, \mathbf{x}^k)\}$ *with* $\mu^k > 0$ *and* $\mu^k \ge \rho_k\mu^0$.

Proof. If $\mu^k > 0$ then Lemma 4 shows that the matrix $G'(\mathbf{y}^k)$ is nonsingular. Hence, Step 2 is well defined at the kth iteration. For any $0 < \alpha \le 1$, define

$$R(\alpha) := G\left(\mathbf{y}^k + \alpha\Delta\mathbf{y}^k\right) - G\left(\mathbf{y}^k\right) - \alpha G'\left(\mathbf{y}^k\right)\Delta\mathbf{y}^k. \tag{23}$$

It follows from (21) that

$$\Delta\mu^k = -\mu^k + \rho_k\mu^0. \tag{24}$$

Hence, for any $0 < \alpha \le 1$, we have

$$\mu^k + \alpha\Delta\mu^k = (1 - \alpha)\mu^k + \alpha\rho_k\mu^0 > 0. \tag{25}$$

From Lemma 4, G is continuously differentiable around $\mathbf{y}^k$. Thus, (23) implies that

$$\|R(\alpha)\| = o(\alpha). \tag{26}$$

On the other hand, (20) yields

$$\rho_k \le \gamma\left\|G\left(\mathbf{y}^k\right)\right\|, \qquad \rho_k \le \gamma\left\|G\left(\mathbf{y}^k\right)\right\|^2. \tag{27}$$

Therefore, for any sufficiently small $\alpha \in (0, 1]$,

$$\begin{aligned} \left\|G\left(\mathbf{y}^k + \alpha\Delta\mathbf{y}^k\right)\right\| &\le \|R(\alpha)\| + (1 - \alpha)\left\|G\left(\mathbf{y}^k\right)\right\| + \alpha\rho_k\mu^0 \\ &\le (1 - \alpha)\left\|G\left(\mathbf{y}^k\right)\right\| + \alpha\gamma\mu^0\left\|G\left(\mathbf{y}^k\right)\right\| + o(\alpha) \\ &= \left[1 - \left(1 - \gamma\mu^0\right)\alpha\right]\left\|G\left(\mathbf{y}^k\right)\right\| + o(\alpha), \end{aligned} \tag{28}$$

where the first inequality follows from (21) and (23), and the second one follows from (26) and (27). Inequality in (28) implies that there exists a constant $0 < \widetilde{\alpha} \le 1$ such that

$$\left\|G\left(\mathbf{y}^k + \alpha\Delta\mathbf{y}^k\right)\right\| \le \left[1 - \sigma\left(1 - \gamma\mu^0\right)\alpha\right]\left\|G\left(\mathbf{y}^k\right)\right\|, \quad \forall\alpha \in (0, \widetilde{\alpha}]. \tag{29}$$

This inequality shows that Step 3 is well defined at the kth iteration. In addition, by (24), Steps 3 and 4 in Algorithm 5, we have

$$\mu^{k+1} = (1 - \lambda_k)\mu^k + \lambda_k\rho_k\mu^0 > 0 \tag{30}$$

holds since $0 < \lambda_k \le 1$ and $\mu^k > 0$. Consequently, from $\mu^0 > 0$ and the above statements, we obtain that Algorithm 5 is well defined.

It is obvious that if Algorithm 5 finitely terminates at kth iteration then $G(\mathbf{y}^k) = 0$, which implies that $\mu^k = 0$ and

$(s^k, \mathbf{x}^k)$ satisfies (9). Hence, $\mathbf{x}^k$ is an optimal solution to the problem described by (1).

Subsequently, we assume that Algorithm 5 does not finitely terminate. Let $\{\mathbf{y}^k = (\mu^k, s^k, \mathbf{x}^k)\}$ be the sequence generated by the algorithm. It follows that $\mu^k > 0$. We want to prove that $\{\mathbf{y}^k\}$ satisfies $\mu^k \geq \rho_k \mu^0$ through the induction method. Clearly, $\rho(\mathbf{y}^0) \leq \gamma \|G(\mathbf{y}^0)\| < 1$, which yields $\mathbf{y}^0 \in \Omega$. Assume that $\mu^k \geq \rho_k \mu^0$; then (24) yields

$$\begin{aligned} \mu^{k+1} - \rho_{k+1}\mu^0 &= \mu^k + \lambda_k \Delta\mu^k \\ &= (1 - \lambda_k)\mu^k + \lambda_k \rho_k \mu^0 - \rho_{k+1}\mu^0 \\ &\geq (1 - \lambda_k)\rho_k \mu^0 + \lambda_k \rho_k \mu^0 - \rho_{k+1}\mu^0 \\ &\geq \mu^0 (\rho_k - \rho_{k+1}). \end{aligned} \tag{31}$$

Clearly,

$$\rho_k = \begin{cases} \gamma \left\| G\left(\mathbf{y}^k\right)\right\|^2, & \text{if } \left\| G\left(\mathbf{y}^k\right)\right\| < 1, \\ \gamma \left\| G\left(\mathbf{y}^k\right)\right\|, & \text{otherwise.} \end{cases} \tag{32}$$

It follows from (20) and (22) that

$$\begin{aligned} \left\| G\left(\mathbf{y}^{k+1}\right)\right\| &\leq \left\| G\left(\mathbf{y}^k\right)\right\|, \\ \rho_{k+1} &\leq \gamma \left\| G\left(\mathbf{y}^{k+1}\right)\right\|, \\ \rho_{k+1} &\leq \gamma \left\| G(\mathbf{y}^{k+1})\right\|^2. \end{aligned} \tag{33}$$

Hence, combining (31), (32), and (33), we obtain that $\mu^{k+1} \geq \rho_{k+1}\mu^0$ which gives the desired result. □

3. Convergence Analysis

In this section we establish the convergence property for Algorithm 5. We show that the sequence $\{\mathbf{y}^k = (\mu^k, s^k, \mathbf{x}^k)\}$ generated by Algorithm 5 is bounded and its any accumulation point yields an optimal solution to the problem described by (1). Furthermore, we show that the sequence $\{\mathbf{y}^k\}$ is superlinearly convergent.

Theorem 6. *The sequence $\{\mathbf{y}^k = (\mu^k, s^k, \mathbf{x}^k)\}$ generated by Algorithm 5 is bounded. Let $\mathbf{y}^* = (\mu^*, s^*, \mathbf{x}^*)$ denote an accumulation point of $\{\mathbf{y}^k\}$. Then $\mu^* = 0$, and $\mathbf{x}^*$ is the optimal solution of the problem described by* (1).

Proof. By Theorem 5,

$$\begin{aligned} \mu^{k+1} &= (1 - \lambda_k)\mu^k + \lambda_k \rho_k \mu^0 \\ &\leq (1 - \lambda_k)\mu^k + \lambda_k \mu^k = \mu^k, \end{aligned} \tag{34}$$

which implies that $\{\mu^k\}$ is monotonically decreasing and hence converges. It follows from (22) that

$$\left\| G\left(\mathbf{y}^{k+1}\right)\right\| \leq \left\| G\left(\mathbf{y}^k\right)\right\|, \quad \forall k. \tag{35}$$

Hence, $\{\|G(\mathbf{y}^k)\|\}$ also converges. Consequently, there exists a constant $M > 0$ such that $\|G(\mathbf{y}^k)\| \leq M$ for all k. This implies that $\{\mu^k\}$ and $\{\mathbf{x}^k\}$ are bounded, and that for any k and $i \in N$,

$$\begin{aligned} &\left| \mathbf{x}_i^k + \left(s^k - \mathbf{a}_i \mathbf{b}_i \exp\left(-\mathbf{b}_i \mathbf{x}_i^k\right)\right)\right. \\ &\left. - \sqrt{\left(\mathbf{x}_i^k\right)^2 + \left(s^k - \mathbf{a}_i \mathbf{b}_i \exp\left(-\mathbf{b}_i \mathbf{x}_i^k\right)\right)^2 + \left(\mu^k\right)^2} \right| \leq M. \end{aligned} \tag{36}$$

This shows that $\{s^k\}$ is also bounded. Consequently, $\{\mathbf{y}^k\}$ is bounded. Let $\mathbf{y}^* = (\mu^*, s^*, \mathbf{x}^*)$ be an accumulation point of $\{\mathbf{y}^k\}$. We assume, without loss of generality, that $\{\mathbf{y}^k\}$ converges to $\mathbf{y}^*$. Then, we have

$$\lim_{k \to \infty} \left\| G\left(\mathbf{y}^k\right)\right\| = \left\| G\left(\mathbf{y}^*\right)\right\|, \qquad \lim_{k \to \infty} \mu^k = \mu^*. \tag{37}$$

By (20),

$$\lim_{k \to \infty} \rho_k = \rho_* := \gamma \left\| G\left(\mathbf{y}^*\right)\right\| \min\left\{1, \left\| G\left(\mathbf{y}^*\right)\right\|\right\}. \tag{38}$$

Suppose $\|G(\mathbf{y}^*)\| > 0$ by contradiction. Then $\rho_* > 0$. From Theorem 5 and (22), we have

$$\mu^* \geq \rho_* \mu^0 > 0, \qquad \lim_{k \to \infty} \lambda_k = 0. \tag{39}$$

Hence, from Step 3 of Algorithm 5, we obtain

$$\left\| G\left(\mathbf{y}^k + \left(\frac{\lambda_k}{\delta}\right)\Delta\mathbf{y}^k\right)\right\| > \left[1 - \sigma\left(1 - \gamma\mu^0\right)\left(\frac{\lambda_k}{\delta}\right)\right]\left\| G\left(\mathbf{y}^k\right)\right\|. \tag{40}$$

Taking $k \to \infty$ in (40) and then combining with (21), we have

$$-\left\| G\left(\mathbf{y}^*\right)\right\|^2 + \rho_* G(\mathbf{y}^*)^T \overline{\mathbf{u}} \geq -\sigma\left(1 - \gamma\mu^0\right)\left\| G(\mathbf{y}^*)\right\|^2, \tag{41}$$

which yields

$$\rho_* \mu^0 \geq \left[1 - \sigma\left(1 - \gamma\mu^0\right)\right]\left\| G\left(\mathbf{y}^*\right)\right\|. \tag{42}$$

Since $\rho_* \leq \gamma \|G(\mathbf{y}^*)\|$, (42) implies

$$(1 - \sigma)\left(1 - \gamma\mu^0\right) \leq 0, \tag{43}$$

which contradicts $\sigma < 1$ and $\gamma\mu^0 < 1$. Therefore, $\|G(\mathbf{y}^*)\| = 0$ and then $\mu^* = 0$. Consequently, $\mathbf{x}^*$ is the optimal solution of the problem described by (1). □

We next analyze the rate of convergence for Algorithm 5. By Theorem 6, we know that Algorithm 5 generates a bounded iteration sequence $\{\mathbf{y}^k\}$ and it has at least one accumulation point. The following lemma will be used in the sequel.

Lemma 7. *Suppose that $\mathbf{y}^* = (\mu^*, s^*, \mathbf{x}^*)$ is an accumulation point of the iteration sequence $\{\mathbf{y}^k\}$ generated by Algorithm 5. Let Q be a matrix in $\partial H(s^*, \mathbf{x}^*)$. Then the matrix Q is nonsingular.*

Proof. Define two index sets:

$$N_1 := \left\{ i \in N : \left(\mathbf{x}_i^*\right)^2 + \left(s^* - \mathbf{a}_i\mathbf{b}_i \exp\left(-\mathbf{b}_i\mathbf{x}_i^*\right)\right)^2 \neq 0 \right\},$$
$$N_2 := \left\{ j \in N : \mathbf{x}_j^* = 0,\ s^* = \mathbf{a}_j\mathbf{b}_j \right\}. \tag{44}$$

By a direct computation, we get

$$Q = \begin{pmatrix} 0 & \mathbf{e}^T \\ \boldsymbol{\alpha} & A \\ \boldsymbol{\beta} & B \end{pmatrix}, \tag{45}$$

where A is an $|N_1| \times n$ matrix with all elements being zero except the (i,i)-th as A_i for $i \in N_1$, B is an $|N_2| \times n$ matrix with all elements being zero except the (j,j)-th as B_j for $j \in N_2$, and

$$\boldsymbol{\alpha} = \operatorname{vec}\left\{ 1 - \frac{s^* - \mathbf{a}_i\mathbf{b}_i \exp\left(-\mathbf{b}_i\mathbf{x}_i^*\right)}{\sqrt{\left(\mathbf{x}_i^*\right)^2 + \left(s^* - \mathbf{a}_i\mathbf{b}_i \exp\left(-\mathbf{b}_i\mathbf{x}_i^*\right)\right)^2}} : i \in N_1 \right\},$$
$$A_i = 1 - \frac{\mathbf{x}_i^*}{\sqrt{\left(\mathbf{x}_i^*\right)^2 + \left(s^* - \mathbf{a}_i\mathbf{b}_i \exp\left(-\mathbf{b}_i\mathbf{x}_i^*\right)\right)^2}} + \mathbf{a}_i\mathbf{b}_i^2 \exp\left(-\mathbf{b}_i\mathbf{x}_i^*\right)\boldsymbol{\alpha}_i, \quad i \in N_1,$$
$$\left(\boldsymbol{\beta}_j, B_j\right) \in \left\{ \left(u, v + \mathbf{a}_j\mathbf{b}_j^2 u\right) : (u-1)^2 + (v-1)^2 \leq 1 \right\}, \quad j \in N_2. \tag{46}$$

Obviously,

$$0 < \boldsymbol{\alpha}_i < 2, \quad 0 < A_i < 2\left(1 + \mathbf{a}_i\mathbf{b}_i^2\right), \quad i \in N_1,$$
$$\boldsymbol{\beta}_j \geq 0, \quad B_j > 0, \quad j \in N_2. \tag{47}$$

Let $Q(w, \mathbf{z}) = 0$. Then we have

$$\sum_{i \in N_1} \mathbf{z}_i + \sum_{j \in N_2} \mathbf{z}_j = 0;$$
$$\boldsymbol{\alpha}_i w + A_i \mathbf{z}_i = 0, \quad i \in N_1; \tag{48}$$
$$\boldsymbol{\beta}_j w + B_j \mathbf{z}_j = 0, \quad j \in N_2,$$

which implies

$$\mathbf{z}_i = -\frac{\boldsymbol{\alpha}_i}{A_i} w, \quad i \in N_1; \qquad \mathbf{z}_j = -\frac{\boldsymbol{\beta}_j}{B_j} w, \quad j \in N_2. \tag{49}$$

Therefore,

$$\left(\sum_{i \in N_1} \frac{\boldsymbol{\alpha}_i}{A_i} + \sum_{j \in N_2} \frac{\boldsymbol{\beta}_j}{B_j} \right) w = 0, \tag{50}$$

which, together with (47), yields $w = 0$. Thus, (49) implies $\mathbf{z} = \mathbf{0}$, and hence the matrix Q is nonsingular. □

Theorem 8. *Let $\{\mathbf{y}^k\}$ be the iteration sequence generated by Algorithm 5. Then $\{\mathbf{y}^k\}$ superlinearly converges to $\mathbf{y}^*$, that is, $\|\mathbf{y}^{k+1} - \mathbf{y}^*\| = o(\|\mathbf{y}^k - \mathbf{y}^*\|)$.*

Proof. By Theorem 6, $\{\mathbf{y}^k\}$ is bounded and then let $\mathbf{y}^* = (\mu^*, s^*, \mathbf{x}^*)$ be its any accumulation point. Hence, $G(\mathbf{y}^*) = 0$ and all matrices $Q \in \partial G(\mathbf{y}^*)$ are nonsingular from Lemma 7. By Proposition 3.1 in [12],

$$\left\| G'\left(\mathbf{y}^k\right)^{-1} \right\| = O(1) \tag{51}$$

for all $\mathbf{y}^k$ sufficiently close to $\mathbf{y}^*$. From Lemma 4, we know that G is semismooth at $\mathbf{y}^*$. Hence, for all $\mathbf{y}^k$ sufficiently close to $\mathbf{y}^*$,

$$\left\| G\left(\mathbf{y}^k\right) - G'\left(\mathbf{y}^k\right)\left(\mathbf{y}^k - \mathbf{y}^*\right) \right\| = o\left(\left\|\mathbf{y}^k - \mathbf{y}^*\right\|\right). \tag{52}$$

Since G is locally Lipschitz continuous near $\mathbf{y}^*$. Therefore, for all $\mathbf{y}^k$ sufficiently close to $\mathbf{y}^*$,

$$\left\| G(\mathbf{y}^k) \right\|^2 = O\left(\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2\right), \tag{53}$$

which implies that

$$\rho_k \mu_0 \leq \gamma \mu^0 \left\| G\left(\mathbf{y}^k\right) \right\|^2 = O\left(\left\| G\left(\mathbf{y}^k\right) \right\|^2\right) = O\left(\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2\right). \tag{54}$$

This inequality, together with (51) and (52), yields

$$\begin{aligned} &\left\|\mathbf{y}^k + \Delta\mathbf{y}^k - \mathbf{y}^*\right\| \\ &= \left\|\mathbf{y}^k + G'\left(\mathbf{y}^k\right)^{-1}\left(-G\left(\mathbf{y}^k\right) + \rho_k\bar{\mathbf{u}}\right) - \mathbf{y}^*\right\| \\ &\leq \left\| G'\left(\mathbf{y}^k\right)^{-1} \right\| \left(\left\| G\left(\mathbf{y}^k\right) - G'\left(\mathbf{y}^k\right)\left(\mathbf{y}^k - \mathbf{y}^*\right) \right\| + \rho_k\mu^0 \right) \\ &= o\left(\left\|\mathbf{y}^k - \mathbf{y}^*\right\|\right). \end{aligned} \tag{55}$$

Following the proof of Theorem 3.1 in [15], we obtain $\|\mathbf{y}^k - \mathbf{y}^*\| = O(\|G(\mathbf{y}^k)\|)$ for all $\mathbf{y}^k$ sufficiently close to $\mathbf{y}^*$. This implies, from (55), that for all $\mathbf{y}^k$ sufficiently close to $\mathbf{y}^*$,

$$\begin{aligned} \left\| G\left(\mathbf{y}^k + \Delta\mathbf{y}^k\right) \right\| &= O\left(\left\|\mathbf{y}^k + \Delta\mathbf{y}^k - \mathbf{y}^*\right\|\right) \\ &= o\left(\left\|\mathbf{y}^k - \mathbf{y}^*\right\|\right) = o\left(\left\| G\left(\mathbf{y}^k\right) \right\|\right). \end{aligned} \tag{56}$$

Since $\lim_{k\to\infty} \|G(\mathbf{y}^k)\| = 0$, it follows from (56) that $\lambda_k = 1$ can satisfy (22) when $\mathbf{y}^k$ is sufficiently close to $\mathbf{y}^*$. Therefore, for all $\mathbf{y}^k$ sufficiently close to $\mathbf{y}^*$, we have

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \Delta\mathbf{y}^k, \tag{57}$$

which, together with (55), proves $\|\mathbf{y}^{k+1} - \mathbf{y}^*\| = o(\|\mathbf{y}^k - \mathbf{y}^*\|)$. Namely, $\{\mathbf{y}^k\}$ superlinearly converges to $\mathbf{y}^*$. □

Table 1: Problem (1) with $\mathbf{a}, \mathbf{b}, c$ randomly generated in different intervals.

DIM	Inter	Gval	CPU (sec.)	AT [5]
100	7	$4.0397e-011$	0.0016	0.0008
500	13.6	$5.7544e-010$	0.0078	0.0021
1000	16.1	$9.3739e-010$	0.0167	0.0290
5000	22.6	$9.2203e-010$	0.0844	0.0980
10000	26	$9.6499e-010$	0.2182	0.3946

Table 2: Problem (1) with with $\mathbf{a}, \mathbf{b}, c$ randomly generated in $[0, 1]$.

DIM	Inter	Gval	CPU (sec.)	AT [5]
100	12.3	$2.0675e-009$	0.006	0.0017
500	15.7	$1.3929e-009$	0.015	0.0021
1000	16.8	$1.2247e-009$	0.037	0.0524
5000	19.2	$1.0316e-009$	0.119	0.1082
10000	21.6	$1.9056e-009$	0.246	0.4693

Table 3

Activity	A	B	C	D
a	4×10^6	3×10^6	2×10^6	10^6
b	2×10^{-6}	3×10^{-6}	10^{-6}	10^{-6}

Table 4

Region	I	II	III	IV	V
a	0.1013	0.3205	0.1323	0.2730	0.1730
b	0.01	0.02	0.01	0.02	0.01

4. Computational Experiments

In this section, we report some numerical results to show the viability of Algorithm 5. First, we compare the numerical performance of Algorithm 5 and the algorithm in [5] on two randomly generated problems. Second, we apply Algorithm 5 to solve two real world examples. Throughout the computational experiments, the parameters used in Algorithm 5 were $\delta = 0.75$, $\sigma = 0.25$, $\mu^0 = 0.001$, and $\gamma = \min\{1/\|G(\mathbf{y}^0)\|, 0.99\}$. In Step 1, we used $\|G(\mathbf{y}^k)\| \leq 10^{-8}$ as the stopping rule. The vector of ones is all the starting points.

Firstly, problems in the form of (1) with 100, 500, 1000, 5000, and 10000 variables were computed. In the first randomly generated example, $\mathbf{a}_i$ was randomly generated in the interval $[10, 20]$, $\mathbf{b}_i$ was randomly generated in the interval $[1, 2]$ for each $i \in N$, and c was randomly generated in the interval $[50, 51]$. In the second randomly generated example, $\mathbf{a}_i$ with $\sum_{i=1}^{n} \mathbf{a}_i = 1$ and $\mathbf{b}_i$ was randomly generated in the interval $[0, 1]$ for each $i \in N$, and c was also randomly generated in the interval $[0, 1]$. Each problem was run 30 times. The numerical results are summarized in Tables 1 and 2, respectively. Here, Dim denotes the number of variable; AT [5] denotes the average run time in seconds used by the algorithm in [5]. In particular, we list more items for Algorithm 5, Inter denotes the average number of iterations, CPU (sec.) is the average run time, and Gval denotes the average values of $\|G(\mathbf{y}^k)\|$ at the final iteration.

The numerical results reported in Tables 1 and 2 show that the proposed algorithm solves the test problems much faster than the algorithm in [5] when the size of problem is large.

Secondly, we apply Algorithm 5 to solve two real world problems. The first example is described in [16]. This problem is how to allocate a maximum amount of total effort, c, among n independent activities, where $\mathbf{a}_i(1-\exp(-\mathbf{b}_i\mathbf{x}_i))$ is the return from the ith activity, that is, effort x_i, to yield the maximum total return. Note that here $\mathbf{a}_i$ is the potential attainable and b_i is the rate of attaining the potential from effort $\mathbf{x}_i$. When no effort is devoted to the ith activity, the value of $\mathbf{x}_i$ is zero. This example usually arises in the marketing field; the activities may correspond to different products, or the same product in different marketing areas, in different advertising media, and so forth. In this example we wish to allocate one million dollars among four activities with values of $\mathbf{a}_i$ and $\mathbf{b}_i$ as in Table 3.

For this problem Algorithm 5 obtained the maximum total return 4.8952×10^6 after 25 iterations and elapsing 0.0625 second CPU time. The total effort one million dollars was allocated, 0.5764 million and 0.4236 million to activities A and B, respectively.

The second example is to search an object in 5 regions, where $\mathbf{a}_i$ is the prior probability with $\sum_{i=1}^{5} \mathbf{a}_i = 1$ of an object of search being in the ith region and $1 - \exp(-\mathbf{b}_i\mathbf{x}_i)$ is the probability of finding an object known to be in the ith region with $\mathbf{x}_i$ time units. The data of this example, listed in Table 4, come from Professor R. Jiang of Jiaozhou Bureau of Water

Conservancy: the available time is $c = 30$ time unite, and after 16 iterations and elapsing 0.0781 second CPU time Algorithm 5 computed the result $\mathbf{x} = (0, 16.6037, 0, 13.3963, 0)$. This shows that we will spend 17 time units in Region II and 13 time units in Region IV to find water.

5. Conclusions

In this paper we have proposed a Newton-type algorithm to solve the problem of search theory. We have shown that the proposed algorithm has global and superlinear convergence. Some randomly generated problems and two real world problems have been solved by the algorithm. The numerical results indicate that the proposed algorithm is promising.

Acknowledgment

This work was supported by NSFC Grant (11271221).

References

[1] B. O. Koopman, *Search and Screening. General Principles with Historical Applications*, Military Operations Research Society, Alexandria, Va, USA, 1999.

[2] M. Patriksson, "A survey on the continuous nonlinear resource allocation problem," *European Journal of Operational Research*, vol. 185, no. 1, pp. 1–46, 2008.

[3] V. V. Popovich, Y. A. Ivakin, and S. S. Shaida, "Theory of search for moving objects," in *Proceedings of the IEEE Ocean Conference and Exhibition*, pp. 1319–1329, Biloxi, Miss, USA, October 2002.

[4] W. Chen, Y. J. Shi, H. F. Teng, X. P. Lan, and L. C. Hu, "An efficient hybrid algorithm for resource-constrained project scheduling," *Information Sciences*, vol. 180, no. 6, pp. 1031–1039, 2010.

[5] S. M. Stefanov, "Convex quadratic minimization subject to a linear constraint and box constraints," *Applied Mathematics Research Express*, vol. 2004, no. 1, pp. 17–42, 2004.

[6] C. L. Sun, J. C. Zeng, and J. S. Pan, "An improved vector particle swarm optimization for constrained optimization problems," *Information Sciences*, vol. 181, no. 6, pp. 1153–1163, 2011.

[7] A. Fischer, "A special Newton-type optimization method," *Optimization*, vol. 24, no. 3-4, pp. 269–284, 1992.

[8] L. Zhang, S.-Y. Wu, and T. Gao, "Improved smoothing Newton methods for P_0 nonlinear complementarity problems," *Applied Mathematics and Computation*, vol. 215, no. 1, pp. 324–332, 2009.

[9] L. P. Zhang and Y. Zhou, "A new approach to supply chain network equilibrium models," *Computers & Industrial Engineering*, vol. 63, pp. 82–88, 2012.

[10] R. Mifflin, "Semismooth and semiconvex functions in constrained optimization," *SIAM Journal on Control and Optimization*, vol. 15, no. 6, pp. 959–972, 1977.

[11] L. Qi, "Regular pseudo-smooth NCP and BVIP functions and globally and quadratically convergent generalized Newton methods for complementarity and variational inequality problems," *Mathematics of Operations Research*, vol. 24, no. 2, pp. 440–471, 1999.

[12] L. Q. Qi and J. Sun, "A nonsmooth version of Newton's method," *Mathematical Programming*, vol. 58, no. 3, pp. 353–367, 1993.

[13] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, NY, USA, 1983.

[14] X. Chen, L. Qi, and D. Sun, "Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities," *Mathematics of Computation*, vol. 67, no. 222, pp. 519–540, 1998.

[15] L. Q. Qi, "Convergence analysis of some algorithms for solving nonsmooth equations," *Mathematics of Operations Research*, vol. 18, no. 1, pp. 227–244, 1993.

[16] C. Wilkinson and S. K. Gupta, "Allocation promotional effort to competing activities: a dynamic programming approach," in *Proceedings of the International Federation of Operational Research Societies Conference (IFORS '69)*, pp. 419–432, Venice, Italy, 1969.