Journal of Computational and Applied Mathematics 232 (2009) 415–426

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A filled function method applied to nonsmooth constrained global optimization[☆]

Ying Zhang[a,*], Yingtao Xu[a], Liansheng Zhang[b]

[a] *Department of Mathematics, Zhejiang Normal University, Jinhua Zhejiang 321004, China*

[b] *Department of Mathematics, Shanghai University, Baoshan Shanghai 200444, China*

ARTICLE INFO

Article history:
Received 5 August 2008
Received in revised form 2 March 2009

MSC:
90C56
90C30

Keywords:
Nonsmooth optimization
Global optimization
Filled function
Local minimizer
Global minimizer

ABSTRACT

The filled function method is an effective approach to find a global minimizer. In this paper, based on a new definition of the filled function for nonsmooth constrained programming problems, a one-parameter filled function is constructed to improve the efficiency of numerical computation. Then a corresponding algorithm is presented. It is a global optimization method which modify the objective function as a filled function, and which find a better local minimizer gradually by optimizing the filled function constructed on the minimizer previously found. Illustrative examples are provided to demonstrate the efficiency and reliability of the proposed filled function method.

1. Introduction

Many recent advances in engineering, finance and social science rely on the globally optimal solutions. So a study of global optimization problems has become a highly concerned topic. However, due to the existence of multiple local minimizers that differ from the global solution, we have to face two difficulties: how to jump from a local minimizer to a smaller one and how to judge that the current minimizer is a global one. Hence all these problems cannot be solved by classical nonlinear programming techniques directly. In general, global optimization methods can be classified into two categories: stochastic methods and deterministic methods. The stochastic methods such as those in [1–4], are based on biology or statistical physics, which jump to the local minimizer by using a probability based approach. Although these methods have their uses, they have some handicaps such as slow convergence or easiness of getting into local optimization. How to leave the local minimizer previously found quickly is an issue to be considered. Some researchers have proposed a method which uses several initial points to conduct multi-search to solve this problem. But this method employ random factors, for example points are chosen randomly. Thus every run of such an algorithm could yield a different result. However, deterministic methods such as those in [5–8], converge more rapidly, and can often escape from the previously found local minimizer to a better one. The results obtained by deterministic methods are always reproducible.

The filled function method, first proposed for smooth optimization in [5], and recently reconsidered in [9,7], is one of the effective deterministic global optimization methods. It provides us with a good idea to use the local optimization

☆ This work was supported by the National Natural Science Foundation (No. 10571116).

* Corresponding author.
E-mail address: znuzy@shu.edu.cn (Y. Zhang).

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.06.020

techniques to solve global optimization problems. The key idea of filled function method is to modify the objective function as a filled function via the current local minimizer of the primal optimization problem, with the property that the better local minimizer can gradually be obtained by optimizing the filled functions constructed on the minimizer previously found locally. Both the theoretical and the algorithmic studies show that the filled function method is competitive with other existing global optimization methods, such as the tunneling method [6], the branch-and-bound method [8] and stochastic methods [1–4].

Nonsmooth constrained global optimization is an even tougher area to tackle, and there is lack of well-developed methods. We note that all the existing filled function methods, such as those in [1,9,7], focus only on solving unconstrained global optimization problems or box constrained global optimization problems or smooth constrained global optimization problems, therefore we generalize the applicable area of the filled function method to nonsmooth constrained global optimization. On the other hand, most of the existing filled functions include parameters in exponential terms and the parameters are heavily restricted by the minimal radius of the local minimizers domain, the calculation performance is imperfect. Therefore, an effective filled function with easily adjusting parameters is worth investigating.

The aim of this paper is to develop the filled function with certain satisfactory properties in nonsmooth constrained global optimization. The remainder of the paper is organized as follows. Following this introduction, a one-parameter filled function is proposed in Section 2. Then, the properties of the filled function are investigated. In Section 3, a novel filled function method with satisfactory numerical results is presented. In Section 4, the performance of the proposed filled function method is compared to the performance of some well-known global optimization methods, the branch-and-bound method and the tunneling method. Finally, some suggestions and conclusions are given in Section 5.

2. A one-parameter filled function and its properties

2.1. Nonsmooth preliminaries

For readers' convenience, we recall some definition and lemmas from [10].

Definition 2.1. Let $f(x)$ be Lipschitz continuous with constant L at the point x, the generalized gradient of $f(x)$ at x is

$$\partial f(x) = \{\xi \in R^n | \langle \xi, d\rangle \le f^0(x; d), \forall d \in R^n\},$$

where

$$f^0(x; d) = \limsup_{y\to x, t\downarrow 0} \frac{f(y+td) - f(y)}{t}.$$

Lemma 2.1. *Let $f(x)$ be Lipschitz continuous with constant L at the point x, then*

(a) *$f^0(x; d)$ is finite, sublinear and satisfies $|f^0(x; d)| \le L\|d\|$; for any $d \in R^n$, one has $f^0(x; d) = \max\{\langle \xi, d\rangle | \xi \in \partial f(x)\}$.*
(b) *$\partial f(x)$ is a nonempty compact convex set; for any $\xi \in \partial f(x)$, one has $\|\xi\| \le L$.*
(c) *$\partial(\sum_{i=1}^n s_i f_i)(x) \subseteq \sum_{i=1}^n s_i \partial f_i(x)$, for any scalar $s_i \in R$.*

Lemma 2.2. *If x_1^* is a local minimizer of $f(x)$, then $0 \in \partial f(x_1^*)$.*

2.2. Problem formulation

The original intention of the filled function method was to solve unconstrained global optimization problems. When objective function $f : R^n \to R$ satisfies the condition $f(x) \to +\infty$, as $\|x\| \to +\infty$, that is, $f(x)$ is coercive, then it implies the existence of a closed bounded box $S \subset R^n$, called the operated region, such that S contains all minimizers of $f(x)$ and the value of $f(x)$ when x is on the boundary of S is greater than any values of $f(x)$ when x is inside S. But this is just for the purpose of analysis and such a region is assumed to exist already. The configuration of the filled function method in a constrained environment involves a constructive process that establishes the operated region S.

Consider the following nonsmooth constrained global optimization problem (P):

$$\min_{x\in\Omega} f(x) \tag{2.1}$$

where feasible region $\Omega = \{x \in R^n | g_i(x) \le 0, i = 1, \ldots, m\}$, $f(x)$ and $g_i(x)$, $i = 1, \ldots, m$ are Lipschitz continuous with constant L_f and L_{g_i}, $i = 1, \ldots, m$, respectively. For convenience, by $L(P)$ we denote the set of local minimizers for problem (P) and $G(P)$ the set of the global minimizers for problem (P).

To begin with, we have the following assumptions.

Assumption 1. The number of minimizers of problem (P) can be infinite, but the number of different function values at the minimizers is finite.

Assumption 2. For every local minimizer x_1^*, the set $L^{x_1^*} = \{x \in L(P) : f(x) = f(x_1^*)\}$ is a bounded closed set.

The following definitions of basin and hill from [11] will be needed in what follows.

Definition 2.2. A basin of $f(x)$ at an isolated minimizer x_1^* is a connected domain B_1 which contains x_1^* and in which starting from any point the steepest descent trajectory of $f(x)$ converges to x_1^*, but outside which the steepest descent trajectory of $f(x)$ does not converge to x_1^*. A hill of $f(x)$ at x_1^* is the basin of $-f(x)$ at its minimizer x_1^*, if x_1^* is a maximizer of $f(x)$.

Definition 2.3. A local minimizer x_2^* is said to be higher than x_1^* if and only if $f(x_2^*) > f(x_1^*)$, and, for this case, B_2 is said to be a higher basin than B_1.

The initial definition of the filled function was proposed in [5] as follows.

Definition 2.4. A function $P(x, x_1^*)$ is called a filled function of $f(x)$ at a local minimizer x_1^* if

(P1) x_1^* is a strictly maximizer of $P(x, x_1^*)$ and the whole basin B_1^* of $f(x)$ at x_1^* becomes a part of a hill of $P(x, x_1^*)$.
(P2) $P(x, x_1^*)$ has no minimizers or saddle points in any basin of $f(x)$ higher than B_1^*.
(P3) If $f(x)$ has a lower basin than B_1^*, then there is a point x' in such a basin that minimizes $P(x, x_1^*)$ on the line through x and x_1^*.

Definition 2.4 depends on the concepts of basin and hill of $f(x)$, then it needs the assumption that $f(x)$ has only a finite number of local minimizers and there exists a minimizer on the line joining x' and x_1^*. These features may affect the computability when applied to numerical optimization. We present a new definition of the filled function as follows.

Definition 2.5. A function $P(x, x_1^*)$ is called a filled function of $f(x)$ at a local minimizer x_1^* if

(P1) x_1^* is a strictly local maximizer of $P(x, x_1^*)$ on Ω.
(P2) To any $x \in \Omega_1 \setminus x_1^*$ or $x \in R^n \setminus \Omega$, one has $0 \notin \partial P(x, x_1^*)$, where $\Omega_1 = \{x \in R^n \mid f(x) \geq f(x_1^*), g_i(x) \leq 0, i = 1, \ldots, m\}$.
(P3) If $\Omega_2 = \{x | f(x) < f(x_1^*), x \in \Omega\}$ is not empty, then there exists a point $x_2^* \in \Omega_2$ such that x_2^* is a local minimizer of $P(x, x_1^*)$.

Note that Definition 2.5 about the filled function is different from Definition 2.4. This novel definition is stronger than that one. If $P(x, x_1^*)$ is a filled function in the sense of Definition 2.5 and x_1^* is not a global minimizer, then we can find a point $\bar{x}$ satisfying $f(\bar{x}) < f(x_1^*)$ in the course of minimizing $P(x, x_1^*)$. Therefore, we can obtain a local minimizer of $f(x)$ with lower objective value by searching $f(x)$ starting at $\bar{x}$ via local search schemes.

In a constrained global optimization problem, the global minimizers must be in the feasible region Ω. Consequently, the filled function method only needs to be effective throughout region Ω. It could let region S coincide with region Ω. This approach, however, has pitfalls. An analysis is perhaps helpful to gain the insight.

The filled function method consists of two phases. In Phase 1, a conventional local minimization is performed and a local minimizer x_1^* of $f(x)$ is found. In Phase 2, a filled function is constructed so as to admit x_1^* as a maximizer. A local method is then applied on the filled function to reach a new starting point for Phase 1. In many constrained problems, some of the minimizers reside near or on the boundaries. If we set $\Omega = S$, and the current iterate x_1^* lies near the boundary of S, then the local minimization of the filled function will lead to a point close to x_1^*, without improving on the current iterate. Therefore, the approach of $\Omega = S$, although simple, is not applicable. A feasible configuration scheme must have $\Omega \subset S$, and this leads to the discussion on what particular technique should be employed in Phase 1 to deal with the constraints.

Most effective methods for solving nonsmooth constrained problems are indirect in that the constrained formulation is converted to an unconstrained one. In the case of minimizers located on the boundaries of feasible region Ω, some indirect approaches, such as interior penalty function method [12], exterior penalty function method [13], exact penalty function method [14], may introduce a deviation.

2.3. A one-parameter filled function and its properties

We propose in this section a one-parameter filled function as follows:

$$P(x, x_1^*, \mu) = -\|x - x_1^*\| + \mu(f(x) - f(x_1^*) + \sum_{i=1}^{m}(g_i(x) - g_i(x_1^*))) + \frac{1}{\mu}\min[0, \max(f(x) - f(x_1^*), g_i(x), i = 1, \ldots, m)] \tag{2.2}$$

where $\mu > 0$ is a parameter, x_1^* is a current local minimizer of $f(x)$, $\|\cdot\|$ indicates the Euclidean vector norm.

The first transformation effect defined in (2.2) stretches the objective function $f(x)$ upwards in the region where the function values are higher than the obtained one $f(x_1^*)$, and makes all the local minimizers with values higher than the value of the obtained local minimizer x_1^* disappear. However, the minimizers with lower values of the objective function remain unaffected by the transformation. The second one makes the local minimizer x_1^* become a maximizer of $P(x, x_1^*, \mu)$. There is a big slop near the point x_1^*, and a stationary point of $P(x, x_1^*, \mu)$ must appear in the region lower than which x_1^* exists in. The proofs see the theorems as follows.

Theorem 2.1. *Assume that $x_1^* \in L(P)$. If $0 < \mu < \frac{1}{L_f + \sum_{i=1}^{m} L_{g_i}}$, then x_1^* is a strictly local maximizer of $P(x, x_1^*, \mu)$.*

Proof. Since $x_1^* \in L(P)$, there exists a neighborhood $O(x_1^*, \delta_1)$ of x_1^* with $\delta_1 > 0$ such that $f(x) \geq f(x_1^*)$ for all $x \in O(x_1^*, \delta_1) \bigcap \Omega$. We consider the following two cases:
Case 1: for all $x \in O(x_1^*, \delta_1) \bigcap \Omega$, we have

$$\min[0, \max(f(x) - f(x_1^*), g_i(x), i = 1, \ldots, m)] = 0.$$

Case 2: For all $x \in O(x_1^*, \delta_1) \bigcap (R^n \setminus \Omega)$, there exists at least an $i_0 \in \{1, \ldots, m\}$ such that $g_{i0}(x) > 0$. Then

$$\min[0, \max(f(x) - f(x_1^*), g_i(x), i = 1, \ldots, m)] = 0.$$

So, for any $x \in O(x_1^*, \delta_1), x \neq x_1^*$, when $0 < \mu < \frac{1}{L_f + \sum_{i=1}^m L_{g_i}}$, we have

$$\begin{aligned} P(x, x_1^*, \mu) &= -\|x - x_1^*\| + \mu(f(x) - f(x_1^*) + \sum_{i=1}^m (g_i(x) - g_i(x_1^*))) \\ &\leq -\|x - x_1^*\| + \mu\left(L_f + \sum_{i=1}^m L_{g_i}\right)\|x - x_1^*\| < 0 = P(x_1^*, x_1^*, \mu). \end{aligned}$$

Thus, x_1^* is a strictly local maximizer of $P(x, x_1^*, \mu)$. □

Theorem 2.2. *Assume that $x_1^* \in L(P)$ and $x \in \Omega_1 \setminus x_1^*$ or $x \in R^n \setminus \Omega$. If $0 < \mu < \frac{1}{L_f + \sum_{i=1}^m L_{g_i}}$, then one has $0 \notin \partial P(x, x_1^*, \mu)$.*

Proof. We consider the following two cases:
Case 1: For all $x \in \Omega_1 \setminus x_1^*$, we have

$$f(x) \geq f(x_1^*), g_i(x) \leq 0, \quad i = 1, \ldots, m,$$

then

$$\min[0, \max(f(x) - f(x_1^*), g_i(x), i = 1, \ldots, m)] = 0.$$

Case 2: For all $x \in R^n \setminus \Omega$, there exists at least an $i_0 \in \{1, \ldots, m\}$ such that $g_{i0}(x) > 0$. Then

$$\min[0, \max(f(x) - f(x_1^*), g_i(x), i = 1, \ldots, m)] = 0.$$

So, by $P(x, x_1^*, \mu) = -\|x - x_1^*\| + \mu(f(x) - f(x_1^*) + \sum_{i=1}^m (g_i(x) - g_i(x_1^*)))$, we have

$$\partial P(x, x_1^*, \mu) \subset -\frac{x - x_1^*}{\|x - x_1^*\|} + \mu\left(\partial f(x) + \sum_{i=1}^m \partial g_i(x)\right).$$

Denoting $d = \frac{x - x_1^*}{\|x - x_1^*\|}$, one has

$$\begin{aligned} \langle \partial P(x, x_1^*, \mu), d\rangle &\subset -\frac{\langle x - x_1^*, d\rangle}{\|x - x_1^*\|} + \mu\left(\langle \xi, d\rangle + \sum_{i=1}^m \langle \eta_i, d\rangle\right) \\ &= -\frac{(x - x_1^*)^{\mathrm{T}}(x - x_1^*)}{\|x - x_1^*\|^2} + \mu\left(\frac{\langle \xi, x - x_1^*\rangle}{\|x - x_1^*\|} + \sum_{i=1}^m \frac{\langle \eta_i, x - x_1^*\rangle}{\|x - x_1^*\|}\right) \\ &\leq -1 + \mu\left(\|\xi\| + \sum_{i=1}^m \|\eta_i\|\right) \\ &\leq -1 + \mu\left(L_f + \sum_{i=1}^m L_{g_i}\right) \\ &< 0, \end{aligned}$$

where $\xi \in \partial f(x)$, $\eta_i \in \partial g_i(x)$, $i = 1, \ldots, m$. So, to any $\zeta \in \partial P(x, x_1^*, \mu)$, when $0 < \mu < \frac{1}{L_f + \sum_{i=1}^m L_{g_i}}$, one has $\zeta^{\mathrm{T}} d < 0$. Then $0 \notin \partial P(x, x_1^*, \mu)$. □

Theorem 2.3. *Assume that $x_1^* \in L(P)$ but $x_1^* \notin G(P)$ and $clint\Omega = cl\Omega$. If $0 < \mu < \frac{1}{L_f + \sum_{i=1}^m L_{g_i}}$ and μ is appropriate small, then there exists a point $x_0^* \in \Omega_2$ such that x_0^* is a local minimizer of $P(x, x_1^*, \mu)$.*

Proof. Since $x_1^* \in L(P)$ but $x_1^* \notin G(P)$, there exists an x_2^* is another local minimizer of $f(x)$ such that $f(x_2^*) < f(x_1^*)$, $g_i(x_2^*) \leq 0$, $i = 1, \ldots, m$. Since $f(x), g_i(x)$, $i = 1, \ldots, m$ are Lipschitz continuous and $clint\Omega = cl\Omega$, there exists an $x_3^* \in int\Omega$ such that $f(x_3^*) < f(x_1^*)$, $g_i(x_3^*) < 0$, $i = 1, \ldots, m$.

When $0 < \mu < \frac{1}{L_f+\sum_{i=1}^{m} L_{g_i}}$, one has

$$\begin{aligned}
P(x_3^*, x_1^*, \mu) &= -\|x_3^* - x_1^*\| + \mu(f(x_3^*) - f(x_1^*) + \sum_{i=1}^{m}(g_i(x_3^*) - g_i(x_1^*))) \\
&\quad + \frac{1}{\mu}\max[f(x_3^*) - f(x_1^*), g_i(x_3^*), i = 1, \dots, m] \\
&\le \left(\mu\left(L_f + \sum_{i=1}^{m} L_{g_i}\right) - 1\right)\|x_3^* - x_1^*\| + \frac{1}{\mu}\max[f(x_3^*) - f(x_1^*), g_i(x_3^*), i = 1, \dots, m] \\
&< \frac{1}{\mu}\max[f(x_3^*) - f(x_1^*), g_i(x_3^*), i = 1, \dots, m] < 0.
\end{aligned}$$

Then

$$\lim_{\mu\to 0} P(x_3^*, x_1^*, \mu) = -\infty.$$

For all $x \in \partial\Omega$, there exists at least an $i_0 \in \{1, \dots, m\}$ such that $g_{i0}(x) = 0$. Then

$$\min[0, \max(f(x) - f(x_1^*), g_i(x), i = 1, \dots, m)] = 0,$$

when $0 < \mu < \frac{1}{L_f+\sum_{i=1}^{m} L_{g_i}}$, we have

$$\begin{aligned}
P(x, x_1^*, \mu) &= -\|x - x_1^*\| + \mu(f(x) - f(x_1^*) + \sum_{i=1}^{m}(g_i(x) - g_i(x_1^*))) \\
&\le \left(\mu\left(L_f + \sum_{i=1}^{m} L_{g_i}\right) - 1\right)\|x - x_1^*\|.
\end{aligned}$$

So, when μ is appropriate small, there must be

$$P(x_3^*, x_1^*, \mu) < P(x, x_1^*, \mu).$$

Therefore,

$$\min_{x\in\Omega} P(x, x_1^*, \mu) = \min_{x\in\Omega\setminus\partial\Omega} P(x, x_1^*, \mu) = P(x_0^*, x_1^*, \mu) \le P(x_3^*, x_1^*, \mu),$$

and $\Omega \setminus \partial\Omega$ is an open set. Then, If $0 < \mu < \frac{1}{L_f+\sum_{i=1}^{m} L_{g_i}}$ and μ is appropriate small, $x_0^* \in \Omega \setminus \partial\Omega$ is a local minimizer of $P(x, x_1^*, \mu)$, and by $P(x_0^*, x_1^*, \mu) \le P(x_3^*, x_1^*, \mu)$, there must be $f(x_0^*) < f(x_1^*), g_i(x_0^*) < 0, i = 1, \dots, m$, that is $x_0^* \in \Omega_2$. □

Theorems 2.1–2.3 show that under some conditions on parameter μ, the function $P(x, x_1^*, \mu)$ at point x_1^* is a filled function satisfying Definition 2.5. The following theorem further shows that under some conditions on parameter μ, $P(x, x_1^*, \mu)$ has a good property that the farther the point x leaves x_1^*, the smaller the value of $P(x, x_1^*, \mu)$ is. This property guarantees that in the minimizing search for local minimizers of $P(x, x_1^*, \mu)$, it will not return to the primary basin.

Theorem 2.4. *Assume that $x_1^* \in L(P)$. Suppose that $x_1, x_2 \in R^n$ such that $0 < \|x_1 - x_1^*\| + \eta < \|x_2 - x_1^*\|$, $\|x_1 - x_2\| \le \sigma$, $\sigma, \eta > 0$ or $x_1 \overline{\in} \Omega$ $x_2 \overline{\in} \Omega$, and $f(x_1) > f(x_1^*), f(x_2) > f(x_1^*)$. If $0 < \mu < \min\{\frac{\eta}{(L_f+\sum_{i=1}^{m} L_{g_i})\sigma}, \frac{1}{L_f+\sum_{i=1}^{m} L_{g_i}}\}$, then*

$$P(x_2, x_1^*, \mu) < P(x_1, x_1^*, \mu) < 0 = P(x_1^*, x_1^*, \mu).$$

Proof. For such x_1 and x_2 satisfying hypothesis, it is easy to get $\min[0, \max(f(x_1) - f(x_1^*), g_i(x_1), i = 1, \dots, m)] = 0$ and $\min[0, \max(f(x_2) - f(x_1^*), g_i(x_2), i = 1, \dots, m)] = 0$. We consider

$$\begin{aligned}
P(x_2, x_1^*, \mu) - P(x_1, x_1^*, \mu) &= -(\|x_2 - x_1^*\| - \|x_1 - x_1^*\|) + \mu\left(f(x_2) - f(x_1) + \sum_{i=1}^{m}(g_i(x_2) - g_i(x_1))\right) \\
&\le -(\|x_2 - x_1^*\| - \|x_1 - x_1^*\|) + \mu\left(L_f + \sum_{i=1}^{m} L_{g_i}\right)\|x_2 - x_1\| \\
&= (\|x_2 - x_1^*\| - \|x_1 - x_1^*\|)\cdot\left\{-1 + \mu\left(L_f + \sum_{i=1}^{m} L_{g_i}\right)\frac{\|x_2 - x_1\|}{\|x_2 - x_1^*\| - \|x_1 - x_1^*\|}\right\}
\end{aligned}$$

$$\leq (\|x_2 - x_1^*\| - \|x_1 - x_1^*\|) \cdot \left(-1 + \mu \left(L_f + \sum_{i=1}^{m} L_{g_i}\right) \frac{\sigma}{\eta}\right)$$
$$< 0.$$

Then, when $0 < \mu < \min\{\frac{\eta}{(L_f+\sum_{i=1}^{m} L_{g_i})\sigma}, \frac{1}{L_f+\sum_{i=1}^{m} L_{g_i}}\}$, combining Theorem 2.1, we have

$$P(x_2, x_1^*, \mu) < P(x_1, x_1^*, \mu) < 0 = P(x_1^*, x_1^*, \mu). \quad \square$$

3. Algorithm and numerical experiments

We introduce here the following optimization problem named filled function problem (F):

$$\min_{x\in R^n} P(x, x_1^*, \mu). \tag{3.1}$$

Based on the proposed filled function $P(x, x_1^*, \mu)$, we can obtain a filled function method for the nonsmooth constrained global optimization problem (P). The general idea of this filled function method is as follows: if the current local minimizer x_1^* is not a global minimizer of (P), then we can manage to obtain a point $x_k \in \Omega$ with $f(x_k) < f(x_1^*)$ by applying some local search schemes to problem (F). Consequently we can obtain a better local minimizer of (P) by applying local search schemes to problem (P) starting from x_k. Finally, a global minimizer of (P) can be obtained. The corresponding algorithm denoted by Algorithm NFFM is detailed as follows.

3.1. Algorithm NFFM

Initialization step

1. Choose a disturbance constant δ, e.g., set $\delta := 0.1$.
2. Choose a lower bound of μ such that $\mu_L > 0$, e.g., set $\mu_L := 10^{-8}$.
3. Choose a constant $\hat{\mu} > 0$, e.g., $\hat{\mu} = 0.1$.
4. Choose direction e_k, $k = 1, 2, \ldots, k_0$ with integer $k_0 \geq 2n$, where n is the number of variable.
5. Set $k := 1$.

Main step

1. Start from an initial point x, minimize the original problem (P) by implementing a nonsmooth local search procedure and obtain the first local minimizer x_1^* of $f(x)$.
2. Let $\mu = 1$.
3. Construct the filled function:
$$P(x, x_1^*, \mu) = -\|x - x_1^*\| + \mu \left(f(x) - f(x_1^*) + \sum_{i=1}^{m}(g_i(x) - g_i(x_1^*))\right) + \frac{1}{\mu} \min[0, \max(f(x) - f(x_1^*), g_i(x), i = 1, \ldots, m)]$$
4. **If** $k > k_0$ **then** go to 7.
   else set $x := x_1^* + \delta e_k$ as initial point, minimize the filled function problem (F) by implementing a nonsmooth local search procedure and obtain a local minimizer denoted x_k.
5. **If** $x_k \bar{\in} \Omega$ **then** set $k := k + 1$, go to 4
   else next step.
6. **If** x_k satisfies $f(x_k) < f(x_1^*)$ **then** set $x := x_k$ and $k := 1$, start from x as a new initial point, minimize the original problem (P) by implementing a local search procedure and obtain another local minimizer x_2^* of $f(x)$, set $x_1^* := x_2^*$, go to 2.
   else next step.
7. Decrease μ by setting $\mu := \hat{\mu}\mu$.
8. **If** $\mu \geq \mu_L$ **then** set $k := 1$, go to 3.
   else the algorithm is incapable of finding a better local minimizer. The algorithm stops and x_1^* is taken as a global minimizer.

Now we make some remarks.

(1). Algorithm NFFM consists of two phases, local minimization and filling:

Phase 1: In this phase, a local minimizer x_1^* of $f(x)$ is found.

Phase 2: In this phase, filled function $P(x, x_1^*, \mu)$ is constructed. Minimize $P(x, x_1^*, \mu)$ and phase 2 ends when such an x_k is found that x_k is in Ω_2. Then, Algorithm NFFM reenters phase 1, with x_k as the starting point to find a new minimizer x_2^* of $f(x)$(if such one exists), and so on. Actually, in the course of minimizing $P(x, x_1^*, \mu)$, once such an x_k in Ω_2 is found that $f(x_k) < f(x_1^*)$, then phase 2 ends, and x_k should not be a local minimizer of $P(x, x_1^*, \mu)$.

The above process is repeated until a stopping criterion is met. The last local minimizer to be found is assumed to be the global minimizer.

Table 3.1
Computational results with initial point $(-1.6, -1.0, 0.2)$.

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	$(-1.6, -1.0, 0.2)$	0.0480	$(-1.9802, 0.0013, -0.0006)$	-1.9410
2	0.1	$(1.1931, 0.6332, -1.1931)$	-3.9140	$(1.9889, -0.0001, -0.0111)$	-5.9446

(2). The motivation and mechanism behind Algorithm NFFM are explained below.

In Step 4 of the Initialization step, we can choose direction e_k, $k = 1, 2, \ldots, k_0$ as positive and negative unit coordinate vectors, at this case, $k_0 = 2n$. For example, when $n = 2$, the directions can be chosen as $(1, 0)$, $(0, 1)$, $(-1, 0)$, $(0, -1)$.

In steps 1, 6 of the Main step, we minimize the original problem (P) by nonsmooth constrained local optimization algorithms such as penalty function method, bundle trustering method, quasi-newton method and composite optimal method. In step 4 of the Main step, we minimize the filled function problem (F) by nonsmooth unconstrained local optimization algorithms such as cutting-planes method, Powell method and Hooke–Jeeve method. They are all effective methods to find local minimizers.

In Main step, we let $\mu = 1$ in step 2. Afterwards, it is gradually decreased via the two-phase cycle until it is smaller than sufficiently small positive scales. If the parameter μ is sufficiently small, we cannot find the point x with $f(x) < f(x_1^*)$ yet, then we believe that there does not exist a better local minimizer of $f(x)$, the last local minimizer to be found is assumed to be the global minimizer. The algorithm is terminated. This is the stopping condition of Algorithm NFFM.

(3). In (P), if $\Omega = \{a_i \le x_i \le b_i, i = 1, \ldots, n\}$, $-\infty < a_i < b_i < +\infty$, then let

$$g_i(x) = a_i - x_i, \qquad c_i(x) = x_i - b_i, \quad i = 1, \ldots, n,$$

obviously, $\Omega = \{a_i \le x_i \le b_i, i = 1, \ldots, n\}$ is equivalent to the following formulation $\Omega = \{x \in R^n \mid g_i(x) \le 0, c_i(x) \le 0, i = 1, \ldots, n\}$. That is to say, Algorithm NFFM can solve box constrained programming problems.

(4). The proposed filled function method also adapts to smooth constrained programming problems.

3.2. Numerical experiments

In this subsection we give some numerical examples to illustrate the efficiency of Algorithm NFFM. FORTRAN 95 is used to code this algorithm. The composite optimal method is used to find local minimizers of the original constrained problem, and the Hooke–Jeeve method is used to search for local minimizers of the filled function problems.

The iteration process show the expected behavior: in most cases the sequence of iteration points achieved some neighborhood of the global minimizer after a few number of iterations.

The computational results of Algorithm NFFM are summarized in tables. The following symbols are used: k is the iteration number in finding the kth local minimizer; μ is the parameter to find the $(k+1)$th local minimizer; x_k is the kth new initial point in finding the kth local minimizer; x_k^* is the kth local minimizer; $f(x_k)$ is the function value of the kth new initial point; $f(x_k^*)$ is the function value of the kth local minimizer.

Problem 3.1.

$$\begin{aligned} \min \quad & f(x) = -x_1^2 + x_2^2 + x_3^2 - x_1 \\ \text{s.t.} \quad & x_1^2 + x_2^2 + x_3^2 - 4 \le 0 \\ & \min\{x_2 - x_3, x_3\} \le 0. \end{aligned}$$

Algorithm NFFM succeeds in finding an approximate global minimizer $x^* = (1.9889, -0.0001, -0.0111)^{\mathrm{T}}$ with $f(x^*) = -5.9446$. The computational results are summarized in Table 3.1.

Problem 3.2.

$$\begin{aligned} \min \quad & f(x) = \max\{f_1(x), f_2(x), f_3(x)\} \\ \text{s.t.} \quad & x_1^2 - x_2 - x_4^2 \le 0 \\ & 0 \le x_i \le 3, \quad i = 1, \ldots, 4. \end{aligned}$$

where

$$\begin{aligned} & f_i(x) = f_0(x) + 10 \cdot g_i(x), \quad i = 1, 2, 3 \\ & f_0(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 \\ & g_1(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \\ & g_2(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \\ & g_3(x) = x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5. \end{aligned}$$

Algorithm NFFM succeeds in finding a global minimizer $x^* = (0, 1, 1, 1)^{\mathrm{T}}$ with $f(x^*) = -65$. The computational results are summarized in Table 3.2.

Table 3.2
Computational results with initial point (1, 1, 1, 2.5).

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	(1, 1, 1, 2.5)	26.7500	(0.0000, 1.0000, 0.0000, 2.0000)	−6.0000
2	0.1	(0.6078, 2.0003, 0.0003, 0.0319)	−22.9117	(0.9289, 0.8620, 0.2453, 0.0803)	−35.9939
3	0.01	(0.4012, 0.2524, 0.2288, 0.0000)	−49.4733	(0.0000, 1.0000, 1.0000, 1.0000)	−65.0000

Table 3.3
Computational results with initial point (−16, −1).

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	(−16, −1)	6.1184	(−15.0000, 0.0000)	5.7164
2	0.1	(−1.0585, 0.5165)	2.1433	(0.0001, −0.2094)	−0.3690
3	0.01	(0.0007, −0.0435)	−0.7470	(0.0000, 0.0000)	−2.7183

Table 3.4
Computational results with initial point 6.

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	6	2.9571	5.0000	2.0000
2	0.1	0.9681	1.0383	1.0029	1.0012

Table 3.5
Computational results with initial point −1.5.

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	−1.5	15.4856	−1.0000	3.0000
2	0.1	2.1416	2.1416	1.0000	1.0000

Problem 3.3.

$$\begin{aligned}
\min \quad & f(x) = -20\exp\left(-0.2\sqrt{\frac{|x_1|+|x_2|}{2}}\right) - \exp\left(\frac{\cos(2\pi x_1)+\cos(2\pi x_2)}{2}\right) + 20 \\
s.t. \quad & x_1^2 + x_2^2 \le 300 \\
& 2x_1 + x_2 \le 4 \\
& -30 \le x_i \le 30, \quad i = 1, 2.
\end{aligned}$$

Algorithm NFFM succeeds in finding a global minimizer $x^* = (0, 0)^{\mathrm{T}}$ with $f(x^*) = -2.7183$. The computational results are summarized in Table 3.3.

Problem 3.4.

$$\begin{aligned}
\min \quad & f(x) = \left|\frac{x-1}{4}\right| + \left|\sin\left(\pi\left(1+\frac{x-1}{4}\right)\right)\right| + 1 \\
s.t. \quad & -10 \le x \le 10.
\end{aligned}$$

Algorithm NFFM succeeds in finding an approximate global minimizer $x^* = 1.0029$ with $f(x^*) = 1.0012$. The computational results are summarized in Table 3.4.

Problem 3.5.

$$\begin{aligned}
\min \quad & f(x) = |x-1|(1+10|\sin(x+1)|) + 1 \\
s.t. \quad & -10 \le x \le 10.
\end{aligned}$$

Algorithm NFFM succeeds in finding the global minimizer $x^* = 1$ with $f(x^*) = 1$. The computational results are summarized in Table 3.5.

Problem 3.6.

$$\begin{aligned}
\min \quad & f(x) = \sum_{i=1}^{5} i|\cos((i+1)x+i)| + 5 \\
s.t. \quad & -10 \le x \le 10.
\end{aligned}$$

Table 3.6
Computational results with initial point 3.0.

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	3	14.6608	4.1406	12.4642
2	0.01	3.6169	11.8475	2.6556	11.5947
3	0.001	1.9523	9.8439	2.0219	6.7002

Table 3.7
Computational results with initial point (3, 3) for $n = 2$.

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	(3, 3)	5	(0.5000, 0.5000)	0.0000

Table 3.8
Computational results with initial point $(5, 5, \dots, 5)$ for $n = 10$, $m = 10$.

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	(5, 5, 5, 5, 5, 5, 5, 5 5, 5)	1823.8490	(1.0000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000 0.5000, 0.5000)	9.3968
2	0.1	(1.0000, 0.4923, 0.3741, 0.3411, 0.4908, 0.1613, 0.2356, 0.1201, 0.1981, 0.1974)	0.9581	(1.0000, 0.5000, 0.3333, 0.2500, 0.2000, 0.1667, 0.1429, 0.1250 0.1111, 0.1000)	0.0000

Algorithm NFFM succeeds in finding an approximate global minimizer $x^* = 2.0219$ with $f(x^*) = 6.7002$. The computational results are summarized in Table 3.6.

Problem 3.7.

$$\begin{aligned} \min \quad & f(x) = \sum_{i=1}^{n} |x_i - 0.5| \\ \text{s.t.} \quad & -5 \le x_i \le 5, \quad i = 1, 2, \dots, n. \end{aligned}$$

Algorithm NFFM succeeds in finding the global minimizer $x^* = (0.5, 0.5, \dots, 0.5)^{\mathrm{T}}$ with $f(x^*) = 0$, for all n. The computational results are summarized in Table 3.7, for $n = 2$.

Problem 3.8.

$$\begin{aligned} \min \quad & f(x) = \max_{j=1,\dots,m} \sum_{i=1}^{n} \frac{(ix_i - 1)^2}{i + j - 1} + \min_{j=1,\dots,m} \sum_{i=1}^{n} \frac{(ix_i - 1)^2}{i + j - 1} \\ \text{s.t.} \quad & -10 \le x_i \le 10, \quad i = 1, \dots, n. \end{aligned}$$

Algorithm NFFM succeeds in finding an approximate global minimizer $x^* = (1.0000, 0.5000, \dots, 0.1000)^{\mathrm{T}}$ with $f(x^*) = 0$. The computational results are summarized in Table 3.8, for $n = 10$ and $m = 10$.

We give the following numerical examples to illustrate that Algorithm NFFM also adapts to smooth constrained global optimization.

Problem 3.9.

$$\begin{aligned} \min \quad & f(x) = -x_1 - x_2 \\ \text{s.t.} \quad & x_2 \le 2x_1^4 - 8x_1^3 + 8x_1^2 + 2 \\ & x_2 \le 4x_1^4 - 32x_1^3 + 88x_1^2 - 96x_1 + 36 \\ & 0 \le x_1 \le 3 \\ & 0 \le x_2 \le 4. \end{aligned}$$

Algorithm NFFM succeeds in finding an approximate global minimizer $x^* = (2.3289, 3.1883)^{\mathrm{T}}$ with $f(x^*) = -5.5091$. This problem is taken from [3]. The computational results are summarized in Table 3.9.

Table 3.9
Computational results with initial point (0, 0).

	NFFM					
k	r	q	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	–	(0, 0)	0	(0.5992, 3.4391)	−4.0828
2	0.1	1	(2.1653, 2.2547)	−4.4200	(2.3289, 3.1883)	−5.5091

Table 3.10
Computational results with initial point $(6, 6, \dots, 6)$ and $n = 7$.

	NFFM				
k	μ	x_k	$f(x_k)$	x_k^*	$f(x_k^*)$
1	–	(6, 6, 6, 6, 6, 6, 6)	78.5399	(0.9961, −2.0137, −2.9958, −2.9982, −2.9961, −3.0011, −2.9953)	39.9687
2	0.1	(1.0241, 0.9999, 0.9898, −0.2180, 0.2243, −2.9832, −1.9997)	14.9230	(1.0036, 0.9999, 1.0000, 1.0032, 0.2284, −2.9343, −1.9997)	11.3654
3	0.01	(0.0099, 0.9999, 1.0029, 1.0024, 1.0026, 1.0000, 1.0078)	0.4443	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000)	0.0000

Problem 3.10.

$$\begin{aligned}
&\min \quad f(x) = \frac{\pi}{n}[10\sin^2 \pi x_1 + g(x) + (x_n - 1)^2] \\
&s.t. \quad -10 \le x_i \le 10, \quad i = 1, 2, \dots, n \\
&where \\
&g(x) = \sum_{i=1}^{n-1}[(x_i - 1)^2(1 + 10\sin^2 \pi x_{i+1})].
\end{aligned}$$

Algorithm NFFM succeeds in finding the global minimizer $x^* = (1, 1, \dots, 1)^{\mathrm{T}}$ with $f(x^*) = 0$, for all n. The computational results are summarized in Table 3.10, for $n = 7$.

Problem 3.11.

$$\begin{aligned}
&\min \quad f(x) = \max_{i=1,2,3} f_i(x) + \max_{i=4,5,6} f_i(x) \\
&s.t. \quad -10 \le x_1 \le 10, -10 \le x_2 \le 10 \\
&where \\
&f_1(x) = x_1^4 + x_2^2, \qquad f_2(x) = (2 - x_1)^2 + (2 - x_2)^2, \qquad f_3(x) = 2e^{-x_1+x_2}, \\
&f_4(x) = x_1^2 - 2x_1 + x_2^2 - 4x_2 + 4, \qquad f_5(x) = 2x_1^2 - 5x_1 + x_2^2 - 2x_2 + 4, \\
&f_6(x) = x_1^2 + 2x_2^2 - 4x_2 + 1.
\end{aligned}$$

For Problem 3.11 from [15] our algorithm did not succeed in finding the global minimizer. This is partly due to the fact that, in smooth global optimization, the desirable search directions are frequently interrelated with gradient and the search directions are deterministic; but in nonsmooth global optimization, the generalized gradient is a set and the search directions frequently become indeterministic, then, the choice of search directions is more difficult and complicated. It remains to say that in order to develop more effective algorithms for nonsmooth functions, much work has still to be done.

4. Comparison to other methods

We are now in a position to compare the proposed filled function method to some other methods.

4.1. Comparison to the branch-and-bound method

The branch-and-bound method works efficiently in nonsmooth global optimization. It aims to investigate the problem (P) of finding the set of global minimizers and it is assumed that the number of minimizers is only finite.

The branch-and-bound method usually applies several interval techniques which are called accelerating devices to reject regions in which the optimum can be guaranteed not to lie. For this reason, the original box Ω gets subdivided, and the subregions which cannot contain a global minimizer of objective function are discarded, while the other subregions are divided again until the desired accuracy of the boxes is achieved.

In Table 4.1, the proposed filled function method are compared to the branch-and-bound method. The criteria for comparison are the run-time and stopping-condition. The following notations are used: IM1 is the branch-and-bound

Table 4.1
Comparison of the proposed filled function method and the branch-and-bound method.

Problem	Run-time (in s)			Stopping-condition		
	NFFM	IM1	IM2	NFFM μ	IM1 ε	IM2 ε
Problem 3.4	0.0	0.020	0.0	10^{-8}	10^{-8}	10^{-10}
Problem 3.5	0.0	0.020		10^{-8}	10^{-8}	
Problem 3.6	0.0	0.430	0.0	10^{-8}	10^{-8}	10^{-10}
Problem 3.3	0.120		97.05	10^{-8}		10^{-3}
Problem 3.7	0.020		21.26	10^{-8}		10^{-4}

method of Ratz [16]; IM2 is the branch-and-bound method of Csendes [17]; run-time is the CPU time in seconds to obtain the final results, ε is the tolerance parameter of IM1 and IM2.

From Table 4.1, we observe the following results: the branch-and-bound method IM1 and IM2 are feasible for univariate nonsmooth functions, but, under several variables case, they must choose appropriate tolerance parameter ε. In contrast, the proposed filled function method overcome the shortcoming and obtain improvements. This is partly due to the fact that, the implement of branch-and-bound method depends on the quality of the objective function: the upper bound and the lower bound can be determined easily and appropriately. Under several variables case, the higher the dimension is, the larger the number of the subregions is. The similar phenomenon occurs in smooth case.

If problem (P) has several global minimizers, theoretically, the branch-and-bound method can find all global minimizers of the objective function. However, the proposed filled function method can only find a global minimizer in each algorithm execution. Using several initial points, the proposed filled function method may find more global minimizers but cannot guarantee to find all of them. Naturally, we have a question: for the proposed filled function method, can we design a criterion by using the mechanism of the branch-and-bound method to try finding all global minimizers?

4.2. Comparison to the tunneling method

Tunneling for global optimization was introduced in [6]. Their tunneling method is composed of a sequence of cycles, where each cycle has two phases: a local minimization phase and a tunneling phase. In the first phase, minimization algorithms such as Newton's method or steepest descent method are employed to find a local minimizer x_1^* of objective function $f(x)$. In the second phase, a tunneling function is defined,

$$T(x, x_1^*) = \frac{f(x) - f(x_1^*)}{[(x - x_1^*)^{\mathrm{T}}(x - x_1^*)]^{\alpha}}. \tag{4.1}$$

The tunneling phase searches for the root of $T(x, x_1^*)$. If we can get $\bar{x}$ such that $\bar{x} \neq x_1^*$ and $T(x, x_1^*) \leq 0$, i.e., $f(\bar{x}) \leq f(x_1^*)$, then $\bar{x}$ is a new starting point for the next iteration. The denominator of (4.1) is of a pole at x_1^* with power α which prevents from x_1^* from being a zero of the tunneling function.

From above, we can see that the proposed filled function method is essentially similar to that for the tunneling method. The only difference between a filled function and a tunneling function is the way used for finding in Phase 2.

The objective functions considered in [6] are of number of local minimizers varies between a few and several thousand. The proposed filled function method is also able to find the global minimizer about such functions, such as Problems 3.2, 3.4, 3.9 and 3.10. Furthermore, it is assumed in this paper that the number of minimizers of problem (P) can be infinite, but the number of different function values at the minimizers is finite. Thus, the proposed filled function method can find the global minimizer for problems which have infinite minimizers, theoretically.

However, the tunneling method in [6] has a number of disadvantages:

1. The pole strength α is problem dependent. While searching for a zero, α should be incrementally increased until the pole in the denominator of (4.1) becomes strong enough to eliminate the last local minimizer of higher order. Every increase in α requires the algorithm to be restarted, leading to increased computational effort.

2. The tunneling algorithm may find another local minimizer x_2^*, such that $f(x_1^*) = f(x_2^*)$. In this case, an additional pole must be placed at the second local minimizer, and the tunneling process must be restarted.

3. Division by a pole causes smoothing of $f(x)$ as $x \to \infty$. This smoothing increases with α, yielding a tunneling function that becomes very flat. In this case, zeros can be difficult to detect correctly.

4. The tunneling algorithm may find another local minimizer x_2^*, such that $f(x_1^*) > f(x_2^*)$. In this case, we have to give another tunneling function and calculate its local minimizer. Thus, it will increase the calculation.

The difficulties associated with finding the zeros of (4.1) have been partly overcome by the dynamical tunneling algorithm of Yao [18]. However, this approach also has a number of deficiencies (details can see [11]).

Since the tunneling function and the filled function possess some common properties, in some sense, a tunneling function can be viewed as a filled function, on the other hand, some filled functions can be looked on as tunneling functions. In this sense, both tunneling function and filled function are unifiable. We can use either filled function or tunneling function for finding a new initial point in the algorithm. Therefore, designing auxiliary functions combined with tunneling function and filled function becomes interesting.

5. Conclusion

In this study, we present a one-parameter filled function for nonsmooth constrained global optimization. Based on the analytical properties, we propose a filled function algorithm. Numerical experiments show that this method is effective and its parameters are easy to set.

The idea of finding a global minimizer by using the filled function can be explored in a number of fields such as face recognition system, license plate recognition system, fingerprint identification and retina identification which belong to the artificial intelligence. For example, in the face recognition system, ICP 3D data registration algorithm [19] and BP neural network classifier [20] are two important components, but both of them have handicaps such as slow convergence or easiness of getting into local optimization. By replacing the local search procedure in Algorithm NFFM by ICP algorithm and BP neural network respectively, we can get an improved face recognition method.

From this point of view, a lot of operations can be defined and relations among them can be studied. This opens an extensive area for research and, hoppingly, puts forward an interesting way for utilization of global optimization to modeling of phenomena. Extension of the filled function method in many fields such as engineering design, molecular biology, neural network training and social science may constitute a part of future investigation.

References

[1] F.J. Chen, Exponential stability of delayed cellular neural networks, J. Zhejiang Normal University (Natural Sciences) 30 (2007) 34–38.
[2] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulate annealing, Science 220 (1983) 671–680.
[3] V. Singh, A new criterion for global robust stability of interval delayed neural networks, J. Comput. Appl. Math. 221 (2008) 219–225.
[4] D.G. Sotiropoulos, E.C. Stavropoulos, M.N. Vrahatis, A new hybrid genetic algorithm for global optimization, Nonlinear Anal. 30 (1997) 4529–4538.
[5] R. Ge, The theory of filled function methods for finding global minimizers of nonlinearly constrained minimization problems, J. Comput. Math. 5 (1987) 1–9.
[6] A.V. Levy, A. Montalvo, The tunneling algorithm for the global minimization of functions, SIAM J. Sci. Stat. Comput. 6 (1985) 15–29.
[7] Z.Y. Wu, L.S. Zhang, K.L. Teo, F.S. Bai, New modified function for global optimization, J. Optim. Theory Appl.. 25 (2005) 181–203.
[8] L.S. Zhang, An interval algorithm for nondifferentiable global optimization, J. Appl. Math. Comput. 63 (1994) 101–122.
[9] Z.Y. Wu, F.S. Bai, H.W.J. Lee, Y.J. Yang, A filled function method for constrained global optimization, J. Global. Optim. 39 (2007) 495–507.
[10] F.H. Clarke, Optimization and Nonsmooth Analysis, John Wiley and Sons, New York, 1983.
[11] L.C. Dixon, J. Gomulka, S.E. Herson, Reflection on global optimization problems, in: Optimization in Action, Academic Press, New York, 1976.
[12] C. Carroll, The created response surface technique for optimization nonlinear restrained systems, Oper. Res. 9 (1961) 169–184.
[13] R. Courant, Variational methods for the solution of problems of equilibrium and vibration, Bull. Amer. Math. Soc. 49 (1943) 1–23.
[14] G.D. Pillo, F. Facchinei, Exact barrier function methods for Lipschitz programs, Appl. Math. Optim. 32 (1995) 1–31.
[15] B. Luderer, J. Weigelt, A solution method for a special class of nondifferentiable unconstrained optimization problems, Comput. Optim. Appl. 24 (2003) 83–93.
[16] T. Csendes, J. Pinter, The impact of accelerating tools on the interval subdivision algorithm for global optimization, European J. Oper. Res. 65 (1993) 314–320.
[17] D. Ratz, A nonsmooth global optimization technique using slopes: one-dimensional case, J. Global Optim. 14 (4) (1999) 365–393.
[18] Y. Yao, Dynamic tunneling algorithm for global optimization, IEEE Trans. Syst. Man. Cybern. 19 (5) (1989) 1222–1230.
[19] X.G. Lu, D. Colbry, A.K. Jain, Three-dimensional model based face recognition, Proc. Int. Conf. Pattern Recognit. 3 (2004) 700–703.
[20] K. Li, Y. Song, J.Q. Deng, Combined eigenface and BP-based neural networks for face recognition, Comput. Appl. Research. 6 (2005) 236–238.