RECURSIVE APPROXIMATION OF THE HIGH DIMENSIONAL max FUNCTION

Ş. İ. BİRBİL, S.-C. FANG, J.B.G. FRENK, AND S. ZHANG

Abstract. An alternative smoothing method for the high dimensional max function has been studied. The proposed method is a recursive extension of the two dimensional smoothing functions. In order to analyze the proposed method, a theoretical framework related to smoothing methods has been discussed. Moreover, we support our discussion by considering some application areas. This is followed by a comparison with an alternative well-known smoothing method.

Contents

1. Introduction

In various areas of mathematical programming, the max function plays an important role. A straightforward application is the min − max optimization [13], which often arises within economics and engineering [7, 11]. Although the max function has desirable properties like convexity, monotonicity and Lipschitz continuity, it is not differentiable everywhere. Therefore, powerful computational techniques, like the Newton's method of nonlinear optimization, can not be directly applied. In order to overcome the nondifferentiability, many researchers have proposed different smoothing approximation schemes. Particularly, when the max function has two arguments, several successful approximations can be found in the literature [3, 12]. However, for the high dimensional max function the number of approximation alternatives decreases drastically.

The max function is also used in reformulating the complementarity constraints along with the nonnegativity constraints. Hence, a successful smoothing scheme leads to the development of effective tools for solving nonlinear complementarity

Date: December 26, 2002.

Key words and phrases. smoothing methods, n dimensional max function, recursive approximation, vertical linear complementarity (VLCP).

Affiliations of the authors are at the end of the manuscript.

(NCP) problems, linear complementarity (LCP) problems, and vertical linear complementarity (VLCP) problems [4, 9, 5, 11, 10]. Consequently, difficult optimization problems, like mathematical programs with equilibrium constraints (MPEC) [8], become more tractable.

The main purpose of this paper is to provide a generic approximation for the high dimensional max function. In order to achieve this, we propose a new method, which utilizes the approximation functions for the two dimensional max function and extends them to higher dimensions via a recursive mechanism. We start with preliminary results related to approximation functions. This is followed by the fundamental section on the recursive smoothing method. In order to show the value of the proposed method, we provide illustrative applications and compare the performance of the proposed method with a well-known method from the literature.

A couple of words on notation: the n dimensional real vector space is denoted by $\mathbb{R}^n$, a vector $x \in \mathbb{R}^n$ is considered as a row vector $(x_1, \cdots, x_n)$, the partial ordering $\geq$ (or $\leq$) on $\mathbb{R}^n$ is viewed as component-wise comparison, i.e., for x, $y \in \mathbb{R}^n$ we say $x \geq y$ if and only if $x_i \geq y_i$ for $i = 1, \cdots, n$.

2. Preliminary Results on Smoothing Methods

We give an elementary exposition of the basic results of a smoothing technique already discussed in [1]. These results are needed in a subsequent section on recursive smoothing. Contrary to the proof of the main inequality in [1], based on the Fenchel-Moreau theorem and on the dual representation of the so-called recession function, we only need an elementary observation about convex functions. To start with our exposition, let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function and introduce the function, $F : \mathbb{R}^n \to [-\infty, \infty]$ given by

$$F(x) := \lim_{t \downarrow 0} t f(t^{-1}x). \tag{2.1}$$

As will be shown in Theorem 1, this limit always exists. In applications generally, the function F is nonsmooth, while the convex function f is chosen to be differentiable. Notice that by relation (2.1), it is immediately clear that the function F is also positively homogeneous and convex. The next elementary lemma is now crucial for the derivation of the main inequality.

Lemma 1. *For a given function $f : \mathbb{R}^n \to \mathbb{R}$, the following conditions are equivalent:*

(1) *The function f is convex.*

(2) *For every $x, y \in \mathbb{R}^n$, the function $h : [0, \infty) \to \mathbb{R}$ given by*

$$h(t) := f(y + tx) - f(y)$$

is convex on $[0, \infty)$.

(3) *For every $x, y \in \mathbb{R}^n$, the function $s : (0, \infty) \to \mathbb{R}$ given by*

$$s(t) := t^{-1}(f(y + tx) - f(y))$$

is non-decreasing on $(0, \infty)$.

Proof. The proof of the implication (1) ⇒ (2) is obvious. To show (2) ⇒ (3) we observe for every $0 < \alpha < 1$ that

$$s(\alpha t) = \frac{h(\alpha t)}{\alpha t} \leq \frac{\alpha h(t) + (1 - \alpha) h(0)}{\alpha t} = \frac{\alpha h(t)}{\alpha t} = s(t).$$

To prove (3) ⇒ (1) we introduce for some $x,\ y \in \mathbb{R}^n$ the function $s : (0,\infty) \to \mathbb{R}$ given by

$$s(t) = t^{-1}(f(y + t(x-y)) - f(y)).$$

Since for every $0 < \alpha < 1$, we obtain by (3) that

$$f((1-\alpha)y + \alpha x) - f(y) = \alpha s(\alpha) \leq \alpha s(1) = \alpha(f(x) - f(y)),$$

the desired result in (1) follows. □

In the next result we show that the limit in relation (2.1) always exists, and at the same time we provide an upper bound on the smoothing function $x \longmapsto f(t^{-1}x)$ for every $t > 0$. This inequality is also shown in [1] by using the duality results from convex analysis. However, we do not need these duality results, since the desired inequality is an immediate consequence of the above elementary lemma.

Theorem 1. *For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, the function F, given by relation (2.1), is well defined. Moreover, for every $x \in \mathbb{R}^n$ and $t > 0$, the inequality*

$$tf(t^{-1}x) - F(x) \leq tf(0)$$

holds.

Proof. Since f is a convex function, it follows by Lemma 1 that for every $x \in \mathbb{R}^n$ the function $t \longmapsto t(f(t^{-1}x) - f(0))$ is non-increasing on $(0,\infty)$. Hence, we obtain

$$\begin{aligned} \sup\nolimits_{t>0} t(f(t^{-1}x) - f(0)) &= \lim\nolimits_{t\downarrow 0} t(f(t^{-1}x) - f(0)) \\ &= \lim\nolimits_{t\downarrow 0} tf(t^{-1}x) \end{aligned} \tag{2.2}$$

and this shows the result. □

Introduce for the convex function $f : \mathbb{R}^n \to \mathbb{R}$, the set of functions $f(\cdot;t) : \mathbb{R}^n \to \mathbb{R}$, $t > 0$, given by

$$f(x;t) := tf(t^{-1}x). \tag{2.3}$$

It is shown in the next result under some additional assumptions that the pointwise convergence in relation (2.1) can be replaced by *supnorm* convergence. In the sequel, for every compact set $A \subseteq \mathbb{R}^n$ we denote $\sup\{|h(x)| : x \in A\}$ by $\|h\|_A$.

Lemma 2. *For a given convex function $f : \mathbb{R}^n \to \mathbb{R}$, the following conditions are equivalent:*

(1) *The function F, given by relation (2.1), is finite everywhere.*
(2) *For every compact set $A \subseteq \mathbb{R}^n$, we have $\lim_{t\downarrow 0} \|F - f(\cdot;t)\|_A = 0$.*

Proof. The implication of (2) ⇒ (1) is obvious. To show (1) ⇒ (2), we observe the following: Since F and f are finite convex functions, we obtain by Corollary 10.1.1 of [14] that they are both continuous. Moreover, if we introduce the functions $\overline{f}(\cdot;t) : \mathbb{R}^n \to \mathbb{R}, t > 0$ given by

$$\overline{f}(x;t) := t(f(t^{-1}x) - f(0)),$$

then by part (3) of Lemma 1 we have $\overline{f}(\cdot;s) \geq \overline{f}(\cdot;t)$ for every $0 < s < t$. Also, by relation (2.2), we know that $\lim_{t\downarrow 0} \overline{f}(x;t) = F(x)$. Hence the conditions of Dini's theorem (see Theorem 7.13 of [15]) hold and so for every compact set A, we obtain $\lim_{t\downarrow 0} \|F - \overline{f}(\cdot;t)\|_A = 0$. Using now

$$\|F - f(\cdot;t)\|_A \leq \|F - \overline{f}(\cdot;t)\|_A + t|f(0)|,$$

the desired result follows. □

We want to remark that in the first condition of the above lemma, the finiteness of F is implicitly satisfied whenever the function f is Lipschitz continuous on its domain.

Let us now discuss two relations, which will be useful within the next section on recursive smoothing. For any convex function f and function F given by relation (2.1), we know for finite F that F is continuous. It is now an easy consequence of Lemma 2 that for any sequence x_t satisfying $\lim_{t\downarrow 0} x_t = x_0$, it holds that

$$\lim_{t\downarrow 0} f(x_t;t) = F(x_0). \tag{2.4}$$

Moreover, since F is continuous, we have

$$\beta := \max\{F(x) : \|x\|_\infty = 1\} < \infty,$$

where $\|.\|_\infty$ denotes the Chebyshev norm on $\mathbb{R}^n$ and by Theorem 1 also

$$f(x) - f(0) \leq F(x) = \|x\|_\infty F(\|x\|_\infty^{-1} x) \leq \beta \|x\|_\infty \tag{2.5}$$

for every $x \neq 0$. When we consider the max function in the next section, it is important to note that β in the above relation is equal to 1.

Up to now, we only focused on deriving an upper bound. To derive a lower bound, we observe that in many applications one can choose the differentiable convex function f satisfying $f \geq F$. Since for every $t > 0$ and $x \in \mathbb{R}^n$

$$0 \leq tf(t^{-1}x) - tF(t^{-1}x) = f(x;t) - F(x),$$

we obtain in this case a trivial lower bound.

At this point a natural question, related to real life applications, arises: given F, is it possible to find a function f satisfying equation (2.1)? Clearly, the answer to this question depends on the function F. Moreover, depending on F, there may exist different alternatives for the function f. Nevertheless, to our belief finding a generic procedure to provide an answer to this question is quite difficult and it is a kind of art rather than a mathematical skill.

3. Recursive Smoothing

There exist different approximation functions with stable convergence properties, particularly for the two dimensional max function. However, an analytic form may not be easily extended from two dimensions to n dimensions. Before providing a solution to this problem, first notice that the n dimensional max function can be written recursively as follows

$$\max\{x_1, \cdots, x_n\} = \max\{\max\{x_1, \cdots, x_m\}, \max\{x_{m+1}, \cdots, x_n\}\},$$

where $1 < m < n$. Following this simple observation, we propose a recursive procedure to construct a high dimensional approximation for the max function with more than two variables. In the sequel, $f : \mathbb{R}^2 \to \mathbb{R}$ refers to the function that is used for approximating the two dimensional max function, i.e.,

$$\lim_{t\downarrow 0} tf(t^{-1}x) = \max\{x_1, x_2\}.$$

We start with defining the recursive functions $f_{i,j} : \mathbb{R}^{j-i+1} \to \mathbb{R}$, $1 \leq i < j \leq n$, associated with the function f.

Definition 1. *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a nondecreasing convex function, and define for every $1 \leq i \leq n-1$ the function $f_{i,i+1} : \mathbb{R}^2 \to \mathbb{R}$ by*

$$f_{i,i+1}(x) := f(x).$$

Moreover, for $1 \leq i < j \leq n$ *and* $k = j - i + 1 > 2$*, let the function* $f_{i,j} : \mathbb{R}^k \to \mathbb{R}$ *be given by*

$$f_{i,j}(x_i, ..., x_j) = f(f_{i,u_k}(x_i, ..., x_{u_k}), f_{l_k,j}(x_{l_k}, ..., x_j)), \tag{3.1}$$

where

$$u_k := i + \lceil \frac{k}{2} \rceil - 1 \text{ and } l_k := \begin{cases} u_k & \text{if } k \text{ is odd} \\ u_k + 1 & \text{if } k \text{ is even.} \end{cases}$$

To simplify the notation for the above recursive procedure, we introduce for every $1 \leq i < j \leq n$, the vectors

$$x^{(1)} := (x_i, ..., x_{u_k}) \text{ and } x^{(2)} := (x_{l_k}, ...x_j),$$

and the associated vector

$$x = \begin{cases} (x_i, ..., x_h, ..., x_j) & \text{if } k \text{ is odd} \\ (x_i, ..., x_{l_k}, x_{u_k}, ..., x_j) & \text{if } k \text{ is even,} \end{cases}$$

where $x_h = x_{u_k} = x_{l_k}$ whenever k is odd. In other words, the vector x is the proper concatenation of the vectors $x^{(1)}$ and $x^{(2)}$. The recursive procedure, defined in relation (3.1), can now be rewritten as

$$f_{i,j}(x) = f(f_{i,u_k}(x^{(1)}), f_{l_k,j}(x^{(2)})), 1 \leq j < i \leq n. \tag{3.2}$$

Before considering the computational scheme of evaluating the above recursive function, we first focus on its theoretical properties. As shown by the next lemma, it follows by induction that for every $i < j$, the function $f_{i,j}$ is an increasing convex function.

Lemma 3. *If* $f : \mathbb{R}^2 \to \mathbb{R}$ *is an increasing convex function then the function* $f_{i,j} : \mathbb{R}^{j-i+1} \to \mathbb{R}$ *with* $1 \leq i < j$ *is also an increasing convex function.*

Proof. Clearly for $k := j - i + 1 = 2$ the function $f_{i,j} = f$ is an increasing convex function. Suppose now the induction hypothesis holds for the functions $f_{i,j}$ satisfying $k \leq p$ and consider a function $f_{i,j}$ satisfying $k = p + 1$. Since $u_{p+1} \leq j - 1$ and $l_{p+1} \geq i + 1$, we obtain $u_{p+1} - i + 1 \leq p$ and $j - l_{p+1} + 1 \leq p$. This shows by our induction hypothesis that the functions $f_{i,u_{p+1}}$ and $f_{l_{p+1},j}$ are increasing and convex. Hence by relation (3.2), the function

$$f_{i,j}(x) = f(f_{i,u_{p+1}}(x^{(1)}), f_{l_{p+1},j}(x^{(2)}))$$

is increasing and convex. □

By Theorem 1 and Lemma 5, we immediately obtain for the increasing convex function f the following results:

(1) For every $1 \leq i < j \leq n$, the function $F_{i,j} : \mathbb{R}^k \to (-\infty, \infty]$ given by

$$F_{i,j}(x) := \lim_{t\downarrow 0} t f_{i,j}(t^{-1}x)$$

exists, and it is positively homogeneous and convex. Observe that the functions $F_{i,i+1}$, $1 < i < n - 1$ refer to the same function, which is also denoted by F.

(2) For every $1 \leq i < j \leq n$, $t > 0$, and $x \in \mathbb{R}^{j-i+1}$

$$t f_{i,j}(t^{-1}x) - F_{i,j}(x) \leq t f_{i,j}(0). \tag{3.3}$$

(3) If additionally $f \geq F$, then as a result of the monotonicity and by induction on every $1 \leq i < j \leq n$ and $x \in \mathbb{R}^k$ we have

$$tf_{i,j}(t^{-1}x) \geq F_{i,j}(x).$$

Lemma 4. *If f is an increasing convex function and the function F, given by relation (2.1), is finite everywhere, then for every $1 \leq i < j$, the function $F_{i,j}$ is finite everywhere. Moreover, the recursion*

$$F_{i,j}(x) = F(F_{i,u_k}(x^{(1)}), F_{l_k,j}(x^{(2)}))$$

holds.

Proof. Clearly for $k := j - i + 1 = 2$ the function $F_{i,i+1} = F$ is finite everywhere. Suppose now that the result holds for the functions $F_{i,j}$ satisfying $k \leq p$ and consider some function $F_{i,j}$ with $k = p + 1$. By relations (2.1) and (3.2), it follows with $x_t := (tf_{i,u_{p+1}}(t^{-1}x^{(1)}), tf_{l_{k+1}}(t^{-1}x^{(2)}))$ that

$$F_{i,j}(x) = \lim\nolimits_{t\downarrow 0} tf(t^{-1}x_t). \tag{3.4}$$

Also by relation (3.2), we obtain $\lim_{t\downarrow 0} x_t = x_0$ where

$$x_0 = (F_{i,u_{p+1}}(x^{(1)}), F_{l_{p+1},j}(x^{(2)})).$$

This shows by our induction hypothesis that the vector x_0 belongs to $\mathbb{R}^2$ and, since the function F is finite everywhere, this implies by relations (2.4) and (3.4) that

$$F_{i,j}(x) = F(F_{i,u_{p+1}}(x^{(1)}), F_{l_{p+1},j}(x^{(2)})).$$

□

We are ready to give an upper bound for the relation (3.3). This upper bound can be constructed by using relation (2.5) and induction.

Lemma 5. *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be an increasing convex function and for every $1 \leq i < j \leq n$, $F_{i,j}$ be finite. Then,*

$$tf_{i,j}(t^{-1}x) - F_{i,j}(x) \leq t(\log_2(k-1) + 1)f(0)$$

where $k = j - i + 1$.

Proof. Clearly, for $k := j - i + 1 = 2$ the function $f_{i,j}(0) = f(0)$. So the result is true for $k = 2$. Suppose by induction that the result is true for $k \leq p$ and consider some $f_{i,j}(0)$ with $k = p + 1$. By relation (2.5), we obtain

$$f_{i,j}(0) = f(f_{i,u_{p+1}}(0), f_{l_{p+1},j}(0)) \leq \max(f_{i,u_{p+1}}(0), f_{l_{p+1},j}(0)) + f(0).$$

Since for every $k \geq 3$, $2(u_k - i + 1) \leq k + 1$ and hence $2(u_{p+1} - i) \leq p$, we obtain by our induction hypothesis that

$$\begin{aligned} f_{i,j}(0) &\leq (\log_2(u_{p+1} - i) + 1)f(0) + f(0) \\ &\leq (\log_2(p)f(0) + f(0) \\ &= (\log_2(p) + 1)f(0). \end{aligned}$$

Combining this by relation (3.3), we have the desired result

$$tf_{i,j}(t^{-1}x) - F_{i,j}(x) \leq tf_{i,j}(0) \leq t(\log_2(k-1) + 1)f(0).$$

□

It is desirable to have a fast computation scheme for the recursive approximation and its higher order information. Before we show the computational complexity of these evaluations, notice that the recursive function $f_{i,j}(x)$ is a composition of several $f(x)$ functions. Therefore, for $t > 0$ if we define the smooth approximation of $f(x)$ by

$$f(x;t) := tf(t^{-1}x),$$

then the approximation function defined as

$$f_{i,j}(x;t) := tf_{i,j}(t^{-1}x)$$

also becomes differentiable. Furthermore, we assume that there exists an oracle, which provides one of the following outputs with an appropriate call:

(1) The function value, $f(x;t)$ at point x for $t > 0$.
(2) The gradient vector, $\nabla f(x;t)$ at point x for $t > 0$.
(3) The hessian matrix, $\nabla^2 f(x;t)$ at point x for $t > 0$.

Proposition 1. *Given $k > 2$ and the vector $x \in \mathbb{R}^k$, for every $i < j$ it requires $O(k)$ calls to the oracle to compute the value of $f_{i,j}(x;t)$.*

Proof. A simple counting reveals that the procedure requires at most $\lceil \log_2(k-1) \rceil$ number of recursive steps to complete the computation. At each step, say $s+1$, twice step s calls are made. Hence, the overall complexity is bounded by a constant times $2^{\lceil \log_2(k-1) \rceil} \leq 2(k-1)$. Therefore, computing the value of $f_{i,j}(x;t)$ requires at most $2(k-1)$ calls to the oracle. □

The next proposition shows that the higher order information of $f_{i,j}(x;t)$ can also be computed in polynomial time.

Proposition 2. *Given $k > 2$ and the vector $x \in \mathbb{R}^k$, for every $i < j$ it requires $O(k)$ calls to the oracle to compute the value of the gradient, $\nabla f_{i,j}(x;t)$ and similarly, $O(k)$ calls to compute the value of the Hessian, $\nabla^2 f_{i,j}(x;t)$.*

Proof. Since

$$f_{i,j}(x;t) = f(f_{i,u_k}(x^{(1)};t), f_{l_k,j}(x^{(2)};t);t), \tag{3.5}$$

by using the chain rule, we obtain

$$\nabla f_{i,j}(x;t) = \frac{\partial f(f_{i,u_k}(x^{(1)};t);t)}{\partial f_{i,u_k}(x^{(1)};t)} \nabla f_{i,u_k}(x^{(1)};t) + \frac{\partial f(f_{l_k,j}(x^{(2)};t);t)}{\partial f_{l_k,j}(x^{(2)};t)} \nabla f_{l_k,j}(x^{(2)};t) \tag{3.6}$$

and similarly,

$$\begin{aligned}\nabla^2 f_{i,j}(x;t) = \quad & \tfrac{\partial f(f_{i,u_k}(x^{(1)};t);t)}{\partial f_{i,u_k}(x^{(1)};t)} \nabla^2 f_{i,u_k}(x^{(1)};t) + \tfrac{\partial f(f_{l_k,j}(x^{(2)};t);t)}{\partial f_{l_k,j}(x^{(2)};t)} \nabla^2 f_{l_k,j}(x^{(2)};t) \\ & + \tfrac{\partial^2 f(f_{i,u_k}(x^{(1)};t);t)}{\partial^2 f_{i,u_k}(x^{(1)};t)} \nabla f_{i,u_k}(x^{(1)};t)(\nabla f_{i,u_k}(x^{(1)};t))^T \\ & + \tfrac{\partial^2 f(f_{l_k,j}(x^{(2)};t);t)}{\partial^2 f_{l_k,j}(x^{(2)};t)} \nabla f_{l_k,j}(x^{(2)};t)(\nabla f_{l_k,j}(x^{(2)};t))^T \\ & + \tfrac{\partial^2 f(f_{i,u_k}(x^{(1)};t);t)}{\partial f_{i,u_k}(x^{(1)};t)\partial f_{l_k,j}(x^{(2)};t)} \nabla f_{i,u_k}(x^{(1)};t)(\nabla f_{i,u_k}(x^{(1)};t))^T \\ & + \tfrac{\partial^2 f(f_{l_k,j}(x^{(2)};t);t)}{\partial f_{i,u_k}(x^{(1)};t)\partial f_{l_k,j}(x^{(2)};t)} \nabla f_{l_k,j}(x^{(2)};t)(\nabla f_{l_k,j}(x^{(2)};t))^T\end{aligned}$$

The complexity result follows from the same arguments as in the proof of Proposition 1. □

4. Numerical Examples

As was pointed out in the introduction section, the recursive smoothing method can be applied to diverse problems modelled by a high dimensional max function. For instance, two important applications are $\min - \max$ optimization and vertical linear complementarity (VLCP) problems.

In $\min - \max$ optimization the smooth approximation is directly used to replace the max function, and so that the resulting model becomes a regular nonlinear programming problem [2]. On the other hand, in VLCP problems the system of equalities and nonnegativity constraints is equivalently modelled as a system of equalities composed of high dimensional max functions. After replacing the max functions with their approximations, a system of nonlinear inequalities are formed. Then different methods, like Newton's method, are utilized to solve this system [10]. In the remaining part of this section we will give two elementary examples that will focus on the main idea of these applications. Also, we will discuss the solution approach with the recursive approximation method and compare its performance with another approximation function of the high dimensional max function.

Before we consider the examples, let us introduce the following entropy-type function $g : \mathbb{R}^n \to \mathbb{R}$, which is the most frequently used approximation in the literature:

$$g(x) = \log\left(\sum_{i=1}^{n} e^{x_i}\right). \tag{4.1}$$

This function shows the following convergence property

$$\lim_{t\downarrow 0} tg(t^{-1}x) = \lim_{t\downarrow 0} t\log\left(\sum_{i=1}^{n} e^{\frac{x_i}{t}}\right) = \max\{x_1, \cdots, x_n\}.$$

Therefore, (4.1) is successfully utilized in solving both $\min - \max$ optimization and VLCP problems [2, 10].

Notice that an overflow easily occurs when the exponential function in (4.1) is computed with a very large argument. A well-known trick to handle this potential problem is introducing a constant $z \geq \max\{x_1, \cdots, x_n\}$ and then computing

$$g_z(x;t) := t\log\left(\sum_{i=1}^{n} e^{\frac{x_i - z}{t}}\right) + z. \tag{4.2}$$

In order to apply the recursive approximation method, we choose the following nondecreasing convex function $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x) = \frac{\sqrt{(x_1 - x_2)^2 + 1} + x_1 + x_2}{2}.$$

which leads to the following approximation function

$$f(x;t) := tf(t^{-1}x) = \frac{\sqrt{(x_1 - x_2)^2 + t^2} + x_1 + x_2}{2}. \tag{4.3}$$

The right hand side of the equation (4.3) is the well-known Chen-Harker-Kanzow-Smale function [3]. This function has been extensively used for approximating the two dimensional max function, it is known to be more stable than (4.1).

Although the trick in (4.2) effectively overcomes the difficulty of computing (4.1), another overflow problem occurs whenever the gradient or the Jacobian is computed. However, with the recursive approximation function using (4.3), the computation of the higher order information does not cause any overflow problems. The following elementary examples demonstrate this result.

Example 1. *Suppose we want to solve the following optimization problem*

$$\min_{x\in\mathbb{R}^3} \max\{2x_1, 3x_2 - 4, 10x_3^2\}.$$

The arguments of the max function in Example 1 are differentiable. Hence, if we replace the max function with its smooth approximation, the resulting problem becomes a smooth unconstrained nonlinear programming problem. By using (4.2), the approximation becomes

$$g_z(x;t) = t\log(e^{t^{-1}(2x_1-z)} + e^{t^{-1}(3x_2-4-z)} + e^{t^{-1}(10x_3^2-z)}) + z, \tag{4.4}$$

where $z \geq \max\{2x_1, 3x_2 - 4, 10x_3^2\}$. On the other hand by using (4.3), we have the following recursive approximation

$$f_{1,3}(x;t) := f(f(2x_1, 3x_2 - 4; t), f(3x_2 - 4, 10x_3^2; t); t). \tag{4.5}$$

Then for $t > 0$, we are interested in comparing the computational aspects of solving

$$\min_{x\in\mathbb{R}^3} g_z(x;t)$$

and

$$\min_{x\in\mathbb{R}^3} f_{1,3}(x;t).$$

A straightforward approach to solve these optimization problems is the gradient method [2]. In this approach, for $t > 0$ we need to compute the gradients of both (4.4) and (4.5) denoted by

$$\bigtriangledown g_z(x;t) = (\bigtriangledown g_z^{(1)}(x;t), \bigtriangledown g_z^{(2)}(x;t), \bigtriangledown g_z^{(3)}(x;t))^T \tag{4.6}$$

and

$$\bigtriangledown f_{1,3}(x;t) = (\bigtriangledown f_{1,3}^{(1)}(x;t), \bigtriangledown f_{1,3}^{(2)}(x;t), \bigtriangledown f_{1,3}^{(3)}(x;t))^T, \tag{4.7}$$

respectively. Consider the computation of the first component of (4.6)

$$\bigtriangledown g_z^{(1)}(x;t) = \frac{2(e^{t^{-1}(2x_1-z)})}{e^{t^{-1}(2x_1-z)} + e^{t^{-1}(3x_2-4-z)} + e^{t^{-1}(10x_3^2)}}.$$

Notice that the numerator converges fast to zero when the difference between z and the arguments of the max function is big. Therefore, if one uses (4.6), the first order information required for the gradient method diminishes. On the other hand, the computation of the gradient by (4.7) is more stable as a direct consequence of using (4.3). In order to support these statements, we have applied the unconstrained optimization procedure fminunc in MATLAB. We have the selected the vector $(1,1,1)$ for our starting point, and we have used the equations (4.4-4.7) as the function and the gradient pointers. As we expected, even with parameter t set to 1.0e-4, the first approximation (4.4) using (4.6) has led to an overflow problem but the recursive approximation (4.5) using (4.7) has converged to the point, $(-9,-1,0)$.

Example 2. *Suppose we want to find a solution $x \in \mathbb{R}^3$ such that*

$$H(x) := \begin{bmatrix} \max\{2x_1, 3x_2 - 4, 10x_3^2\} \\ \max\{x_1^2, x_2 + 11, x_3 - 1\} \\ \max\{x_1, x_2 - 0.4, 2x_3^2\} \end{bmatrix} = 0.$$

In this example we are interested in solving a system of nonlinear equations. We first replace the max functions with their smooth approximations. Let $z_1 \geq \max\{2x_1, 3x_2 - 4, 10x_3^2\}$, $z_2 \geq \max\{x_1^2, x_2 + 11, x_3 - 1\}$, and $z_3 \geq \max\{x_1, x_2 - 0.4, 2x_3^2\}$ then by using (4.2) and (4.3), we can write the approximation functions

$$\begin{aligned}
g_{z_1}^{(1)}(x;t) \quad &: \quad = t\log(e^{t^{-1}(2x_1 - z_1)} + e^{t^{-1}(3x_2-4-z_1)} + e^{t^{-1}(10x_3^2 - z_1)}) + z_1,\\
g_{z_2}^{(2)}(x;t) \quad &: \quad = t\log(e^{t^{-1}(x_1^2 - z_2)} + e^{t^{-1}(x_2+11-z_2)} + e^{t^{-1}(x_3 - 1 - z_2)}) + z_2,\\
g_{z_3}^{(3)}(x;t) \quad &: \quad = t\log(e^{t^{-1}(x_1 - z_3)} + e^{t^{-1}(x_2-0.4-z_3)} + e^{t^{-1}(2x_3^2 - z_3)}) + z_3,
\end{aligned}$$

and

$$\begin{aligned}
f_{1,3}^{(1)}(x;t) \quad &: \quad = f(f(2x_1, 3x_2 - 4; t), f(3x_2 - 4, 10x_3^2; t); t),\\
f_{1,3}^{(2)}(x;t) \quad &: \quad = f(f(x_1^2, x_2 + 11; t), f(x_2 + 11, x_3 - 1; t); t),\\
f_{1,3}^{(3)}(x;t) \quad &: \quad = f(f(x_1, x_2 - 0.4; t), f(x_2 - 0.4, 2x_3^2; t); t).
\end{aligned}$$

Suppose we define

$$G_z(x;t) := (g_{z_1}^{(1)}(x;t), g_{z_2}^{(2)}(x;t), g_{z_3}^{(3)}(x;t))^T,$$

where $z := (z_1, z_2, z_3)$ and

$$F_{1,3}(x;t) := (f_{1,3}^{(1)}(x;t), f_{1,3}^{(2)}(x;t), f_{1,3}^{(3)}(x;t))^T.$$

Then for $t > 0$, we want to compare the solution efforts invested in solving

$$G_z(x;t) = 0, \tag{4.8}$$

and

$$F_{1,3}(x;t) = 0. \tag{4.9}$$

Different methods to solve a system of nonlinear equations require the computation of the first order information. Therefore, we need to compute the Jacobians of the functions $G_z(x;t)$ and $F_{1,3}(x;t)$. Similar to the computation of the gradient (4.6), the computation of the Jacobian of $G_z(x;t)$ creates overflow problems. More importantly, the Jacobian becomes almost singular. In other words the reciprocal condition of the Jacobian approaches to 0 fast [6]. On the other hand, the recursive computation of the Jacobian of $F_{1,3}(x;t)$ is more stable and leads to a nonsingular Jacobian. To solve (4.8) and (4.9), we have used the MATLAB procedure fsolve, which is designed to solve a system of nonlinear equations. In both problems, we have provided the function and Jacobian pointers as an input to the fsolve procedure, and as before we have selected the vector $(1, 1, 1)$ as our starting point.

Table 1 shows the results with both approximations. The column 1 shows different t values. For each t value, the approximations (4.8) and (4.9) are solved by fsolve, and the respective results, denoted by H^*, are reported in columns 2 and 3. Clearly, decreasing the t value leads to better solutions. However, it also causes the Jacobian of $G_z(x;t)$ converge to a singular matrix rapidly. Therefore, as the figures in the second column suggest, overflow problems occur with the first approximation. On the other hand, the third column shows that the recursive approximation behaves more stable than the first approximation, and gives improving solutions with decreasing t.

t	Approximation (4.8)	Approximation (4.9)
1.0e-1	$H^* = (4.1e\text{-}3, 3.6e\text{-}2, 7.0e\text{-}4)^T$	$H^* = (3.2e\text{-}6, 1.1e\text{-}2, 6.0e\text{-}6)^T$
1.0e-2	No Solution	$H^* = (1.3e\text{-}8, 6.0e\text{-}4, 2.0e\text{-}9)^T$
1.0e-4	No Solution	$H^* = (7.0e\text{-}15, 2.1e\text{-}5, 1.3e\text{-}15)^T$
1.0e-6	No Solution	$H^* = (4.1e\text{-}14, 3.0e\text{-}8, 8.0e\text{-}16)^T$

Table 1. Comparison of the results of the two approximation schemes for Example 2.

5. Conclusion

We have discussed an alternative smoothing method for the high dimensional max function. The novelty of the method comes from the fact that the high dimensional max function can be written as a recursive composition of the two dimensional max functions. After noticing this fact, we have used smooth approximation functions of the two dimensional max function to approximate the high dimensional case. We have given a mathematical treatment of the proposed method. It is also shown that the proposed method has more desirable properties than the well-known entropy function.

There exist many applications of the high dimensional max function. As a continuation of this research, the performance of the proposed method may be studied on these problems. Another interesting research area could be in nonlinear optimization, where there exists a set of difficult constraints. These constraints can be replaced with a single constraint after aggregating them by the max function. Again a high dimensional approximation would be required for smoothing the max function. In all these applications a rigorous analysis is required, since the arguments of the max function will be nontrivial functions. Overall we believe that a further research along these lines is fruitful and promising.

References

[1] Ben-Tal, A. and M.Teboulle. *A Smoothing Technique for Nondifferentiable Optimization Problems.* In Dolecki, editor, *Optimization*, Lectures notes in Mathematics 1405, pages 1–11, New York, 1989. Springer Verlag.

[2] Bertsekas, D.P. *Constrained Optimization and Lagrange Multiplier Methods.* Academic Press, New York, 1982.

[3] Chen, C. and O.L. Mangasarian. *Smoothing Methods for Convex Inequalities and Linear Complementarity Problems. Mathematical Programming*, 71:51–69, 1995.

[4] Cottle, R.W. and G.B. Dantzig. *A Generalization of The Linear Complementarity Problem. Journal of Combinatorial Theory*, 8:79–90, 1970.

[5] Cottle, R.W., Pang, J.-S. and R.E. Stone. *Linear Complementarity Problem.* Academic Press, Boston, MA, 1992.

[6] Golub, G.H. and C.F. Loan. *Matrix Computations.* Johns Hopkins University, 3rd Ed., Baltimore, MD, 1996.

[7] Love, R.F., Morris, J.G. and G.O. Wesolowsky. *Facilities Location: Models and Methods.* North-Holland, New York, 1988.

[8] Luo, Z.-Q., Pang, J.-S. and D. Ralph. *Mathematical Programs with Equilibrium Constraints.* Cambridge University Press, Cambridge, 1997.

[9] Outrata, J.V., Kocvara, M. and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications and Numerical Results.* Kluwer Academic Publishers, Dordrecht, 1998.

[10] Peng, J.M. and Z. Lin. *A Non-interior Continuation Method for Generalized Linear Complementarity Problems. Mathematical Programming*, 86:533–563, 1999.

[11] Polak, E. *Optimization: Algorithms and Consistent Approximations.* Springer Verlag, Applied Mathematical Sciences, 124, 1997.

[12] Qi, L. and X. Chen. *A Globally Convergent Successive Approximation Methods for Nonsmooth Equations. SIAM J. Control Optim.*, 33:402–418, 1995.

[13] Rockafellar, R.T. *Minimax Theorems and Conjugate Saddle Functions. Math. Scand.*, 14:151, 1964.

[14] Rockafellar, R.T. *Convex Analysis*, volume 28 of *Princeton Mathematical Series.* Princeton University Press, Princeton, New Jersey, 1972.

[15] Rudin, W. *Principles of Mathematical Analysis.* McGraw-Hill, New York, 1982.

(**Ş. İ. Birbil**), Erasmus Research Institute of Management(ERIM), Erasmus University, Postbus 1738, 3000 DR, Rotterdam, The Netherlands
E-mail address: sibirbil@few.eur.nl
URL: http://www.few.eur.nl/few/people/sibirbil

(**S. -C. Fang**), Industrial Engineering and Operations Research, North Carolina State University, Raleigh, NC 26695-7906, USA.
E-mail address: fang@eos.ncsu.edu
URL: http://www.ie.ncsu.edu/~fangroup

(**J.B.G. Frenk**), Econometric Institute, Erasmus University, Postbus 1738, 3000 DR Rotterdam, The Netherlands.
E-mail address: frenk@few.eur.nl
URL: http://www.few.eur.nl/few/people/frenk

(**S. Zhang**), Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.
E-mail address: zhang@se.cuhk.edu.hk
URL: http://www.se.cuhk.edu.hk/~zhang/