Level Function Method for Quasiconvex Programming

H. XU

Communicated by J. P. Crouzeix

Abstract. In this paper, we introduce the notion of level function for a continuous real-valued quasiconvex function. The existence, construction, and application of level functions are discussed. Further, we propose a numerical method based on level functions for the solution of quasiconvex minimization problems. Several versions of the algorithms are presented. Also, we apply the idea of the level function method to the solution of a class of variational inequality problems. Finally, the results of numerical experiments on the proposed algorithms are reported.

Key Words. Level functions, quasiconvex functions, level sets, level function methods.

1. Introduction

The cutting plane method (Ref. 1) is one of the oldest and most important methods in optimization. In the convex case, a cutting plane function is an affine function, and it can be constructed through a gradient or a subgradient which can be identified locally. In a quasiconvex setting, an affine function cannot necessarily be a cutting plane function. Plastria (Ref. 2) introduced the notion of lower subgradient and used it to construct a cutting half-plane function; consequently, he extended the classic cutting plane method.

1This work is a part of the author’s PhD Thesis under the supervision of Professor A. Rubinov, Dr. B. Glover, and Dr. B. Craven at the School of Information Technology and Mathematical Science, University of Ballarat, Victoria, Australia. The author thanks his supervisors for their guidance and Professor H. Tuy and two anonymous referees for their insightful comments. Finally, he is grateful to the Associate Editor J. P. Crouzeix for his help.

2Postdoctoral Research Fellow, Australian Graduate School of Management, University of New South Wales, Sydney, Australia.
method to the quasiconvex setting. While we commend these remarkable results, we also realize that the computation of a lower subgradient for a general quasiconvex function is difficult. Thus, the extension of the cutting plane method to the quasiconvex setting remains a challenging subject in optimization at least computationally.

In this paper, we are motivated to develop a class of algorithms which can be implemented with or without derivative information, depending on the different situation, for quasiconvex programming. For this purpose, we introduce the notion of level function. We prove that a continuous real-valued function has a level function if and only if it is quasiconvex. We also show that a level function of a quasiconvex function at a point can be constructed through either local or global first-order information of the original function when such information is not difficult to be obtained, or through other information otherwise. A method using level functions is consequently proposed for solving a class of quasiconvex minimization problems. The essence of the method is that, at each iteration, the maximum of the level functions, rather than that of the cutting plane functions, is minimized. In comparison with the cutting plane methods, our method has at least two advantages in dealing with quasiconvex minimization problems:

(a) the level functions can be more flexibly constructed;
(b) the maximum of the level functions is a finite continuous convex function; thereby, it can be more easily minimized in comparison with the maximum of support functions. Furthermore, when applied to solve quasiconvex fractional programming problems, the algorithm subsumes the well-known Dinkelbach algorithm and its generalizations. Note that Lemarechal, Nemirovskii, and Nesterov (Ref. 3) proposed recently a variant of the bundle method, known as the level set method, for solving convex minimization problems. In their algorithm, the next iterate is the Euclidean projection of a certain point on a level set of a collection of support functions. Nemirovskii (Ref. 4) made a substantial extension of this method to the quasiconvex setting. In this paper, we make a further extension by making a projection on a level set of the maximum of the level functions. We also note that the proposed level function method is close to the outer approximation method due to Hoffman (Ref. 5), Tuy (Ref. 6), and Thieu, Tam, and Ban (Ref. 7), in the sense that both methods approximate the solution of a problem through level sets instead of function values. The difference is that, at every iterate, the outer approximation method makes a cut which is a linear function and which does not necessarily exist for a quasiconvex function.

In general, if a function has a nonaffine convex level function, it has an affine level function; consequently, we can prove that a continuous function is quasiconvex if and only if the function has an affine level function.
Naturally, this raises a question: why a nonaffine convex level function is needed? Indeed, an affine level function can be constructed by the Greenberg–Pierskalla quasisubdifferential. To answer this question, we recall that our motivation for introducing the notion of level function is to design a class of algorithms which can be implemented with or without derivative information, depending on different situations. An opposite question can therefore be asked: why do we restrict ourselves to the use of an affine level function if it is not necessary to do so? From this point of view, employing a nonaffine level function can be interpreted as introducing a certain flexibility. Computationally, this will be convenient if we are allowed to use a nonaffine and/or an affine level function depending on the different situation; this is particularly true when the level function method is applied to fractional programming. Theoretically, not being confined to affine level functions may make the notion more applicable. We will discuss this point later on.

In this paper, we shall consider a continuous real-valued function \( f: \mathbb{R}^n \to \mathbb{R} \). We will let \( S_f(\alpha) = \{ x \in \mathbb{R}^n : f(x) \leq \alpha \} \) denote a level set of \( f \), and we will let \( T_f(\alpha) = \{ x \in \mathbb{R}^n : f(x) < \alpha \} \) denote a strict level set. Moreover, we will use \( \langle x, y \rangle \) to denote the scalar product of two vectors in \( \mathbb{R}^n \) and \( \| \cdot \| \) to denote the Euclidean norm; \( \text{cl} \Omega \) will represent the closure of a set \( \Omega \) and \( \text{bd} \Omega \) will represent the boundary of \( \Omega \); finally, the set \( \{1, \ldots, m\} \) will be represented by \( m \).

The rest of this paper is organized as follows. In Section 2, we introduce the notion of level function for a continuous real-valued function. The existence, construction, and application of level functions are discussed. In Section 3, we propose a numerical method based on level functions for the solution of quasiconvex minimization problems. Several versions of the algorithms are presented. In Section 4, we apply the idea of the level function method to the solution of a class of variational inequality problems. Finally, in Section 5, we report the results of numerical experiments on the proposed algorithms.

2. Level Functions

In this section, we introduce the notion and main properties of a level function. First, we recall that a continuous real-valued function \( f: \mathbb{R}^n \to \mathbb{R} \) is said to be quasiconvex if \( S_f(\alpha) \) is convex for every \( \alpha \in \mathbb{R} \). It is well known
that the quasiconvexity of $f$ is equally characterized by the convexity of $T_f(\cdot)$. Quasiconvexity is a significant extension of convexity and it has received intensive investigations for its wide applications in mathematical economics, mathematical programming, and other areas.

Defnition 2.1. Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuous real-valued function and $x \in \mathbb{R}^n$. A function $\sigma: \mathbb{R}^n \rightarrow \mathbb{R}$ is called a level function of $f$ at $x$ if it satisfies the following conditions:

(a) $\sigma(x) = 0$;

(b) $\sigma$ is a continuous convex function;

(c) $T_f(f(x)) \subset T_\sigma(0)$.

Throughout, we will denote the set of level functions of a function $f$ at point $x$ by $\mathcal{O}(f,x)$. The following propositions are straightforward from Defnition 2.1.

Proposition 2.1. Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuous function, and let $x \in \mathbb{R}^n$; let $\sigma_1, \sigma_2 \in \mathcal{O}(f,x)$. Then:

(a) $\max\{\sigma_1, \sigma_2\} \in \mathcal{O}(f,x)$;

(b) $\sigma_1 + \sigma_2 \in \mathcal{O}(f,x)$;

(c) for every constant $\alpha \in \mathbb{R}^n$, $\alpha \sigma_1 \in \mathcal{O}(f,x)$;

(d) for every $a \in \partial \sigma_1(x)$, the function $\langle a, y-x \rangle \in \mathcal{O}(f,x)$; here and later, $\partial$ denotes the usual convex subdifferential in convex analysis.

Recall that a set is said to be evenly convex if it is an intersection of open half spaces. It is well-known that all open or closed convex sets are evenly convex. A function is said to be evenly quasiconvex if every level set of the function is an evenly convex set.

Proposition 2.2. Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuous real-valued function. The following statements are equivalent:

(a) $f$ is quasiconvex on $\mathbb{R}^n$;

(b) for every $x \in \mathbb{R}^n$, $\mathcal{O}(f,x)$ is nonempty.

Proof. (a) $\Rightarrow$ (b). Let $x \in \mathbb{R}^n$, and assume that $T_f(f(x)) \neq \emptyset$. It is well known that a continuous quasiconvex function is evenly quasiconvex and that $T_f(f(x))$ is an evenly convex set. Therefore, $T_f(f(x))$ can be represented as

$$\bigcap_{i=1} \{y \in \mathbb{R}^n : \langle a_i, y \rangle - b_i < 0\}.$$
where \( I \) is the index set. Since \( x \not\in T_f(f(x)) \), there exists \( i \in I \) such that
\[
\langle a_i, x \rangle - b_i \geq 0
\]
and, for all \( y \in T_f(f(x)) \)
\[
\langle a_i, y \rangle - b_i < 0.
\]
Let
\[
\sigma_x(y) = \langle a_i, y - x \rangle.
\]
Then, for all \( y \in T_f(f(x)) \)
\[
\sigma_x(y) = \langle a_i, y - x \rangle \leq \langle a_i, y \rangle - b_i < 0.
\]
Clearly, \( \sigma_x(y) \) is a level function of \( f \) at the point \( x \).

(b) \( \Rightarrow \) (a). Let \( r \in \mathbb{R} \). Then, \( S_r(r) \) either coincides with \( \mathbb{R}^n \) or is empty or coincides with \( S_r(f(x)) \) for some \( x \in \mathbb{R}^n \). If suffices to show that \( S_r(f(x)) \) is evenly convex. Let \( y \in \mathbb{R}^n \setminus S_r(f(x)) \) and \( \sigma_y \in \partial (f, y) \). Then,
\[
S_r(f(x)) = \bigcap_{y \in \mathbb{R}^n \setminus S_r(f(x))} T_{\sigma_y}(0).
\]
Thus, \( S_r(f(x)) \) is an intersection of open convex sets, whence it is evenly convex. This completes the proof.

The importance of Proposition 2.2 is that, on the one hand, it proves the existence of a level function under moderate conditions, on the other hand it provides an alternative way to investigate the quasiconvexity of a function; that is, if a continuous function has a level function at every point, then it is quasiconvex. We will discuss this at the end of this section.

We note that, in Definition 2.1(c), if we replace \( T_f(f(x)) \subset T_{\sigma}(0) \) with \( S_r(f(x)) \subset S_r(0) \), then a continuous quasiconvex function may not have a level function other than a zero function at a point. For instance, \( f(x) = \min \{ |x|, 1 \} \) does not have a nonzero level function at any point of \( \{ x : |x| > 1 \} \).

Note also that Definition 2.1 does not exclude the case that \( \sigma \equiv 0 \). Indeed, if \( 0 \not\in \partial (f, x) \), then \( T_f(f(x)) \) is an empty set and \( f \) achieves its global minimum at \( x \). Combining this comment with Proposition 2.1(d), we have the following conclusion.

**Proposition 2.3.** Let \( f: \mathbb{R}^n \rightarrow \mathbb{R} \) be a continuous quasiconvex function, and let \( x \in \mathbb{R}^n \). If there exists a level function \( \sigma \in \partial (f, x) \) such that \( 0 \in \partial \sigma(x) \), then \( x \) is a global minimizer of \( f \).
A level function is defined through strict level sets. In what follows, we show that the normal cone of the strict level set of a quasiconvex function can be characterized by level functions.

Let $N(x, T_f(f(x)))$ denote the normal cone of $T_f(f(x))$ at $x$.

**Proposition 2.4.** Suppose that $f: \mathbb{R}^n \to \mathbb{R}$ is a continuous quasiconvex function such that, for every $x \in \mathbb{R}^n$, $\text{cl} T_f(f(x))$ is a compact set whenever $T_f(f(x)) \neq \emptyset$. Then,

$$\text{int } N(x, T_f(f(x))) = \text{cone } \bigcup_{\sigma \in (f,x)} \partial \sigma(x),$$

where int denote the interior of a set and

$$\text{cone } \Omega = \{ \alpha \omega : \alpha > 0, \omega \in \Omega \}.$$

**Proof.** Let $x \in \mathbb{R}^n$. Suppose that $T_f(f(x)) \neq \emptyset$. By assumption, $\text{cl} T_f(f(x))$ is a compact set. For $h \in \mathbb{R}^n$, define

$$\psi(h) = \max\{ \langle h, y - x \rangle : y \in \text{cl} T_f(f(x)) \}.$$

Then, $\psi(h)$ is continuous and sublinear. Thus,

$$N(x, T_f(f(x))) = \{ h \in \mathbb{R}^n : \psi(h) \leq 0 \}$$

and

$$\text{int } N(x, T_f(f(x))) = \{ h \in \mathbb{R}^n : \langle h, y - x \rangle < 0, \forall y \in T_f(f(x)) \}.$$

Let

$$h \in \text{int } N(x, T_f(f(x))).$$

Then, for all $y \in T_f(f(x))$,

$$\langle h, y - x \rangle < 0.$$

Let

$$\sigma(y) = \langle h, y - x \rangle.$$

Obviously, $\sigma \in (f,x)$ and $h \in \partial \sigma(x)$. Thus,

$$h \in \text{cone } \bigcup_{\sigma \in (f,x)} \partial \sigma(x).$$

Conversely, let

$$h \in \text{cone } \bigcup_{\sigma \in (f,x)} \partial \sigma(x).$$
By definition, there exists a $\lambda > 0$ and a vector $d \in \bigcup_{x \in \mathcal{F}(x)} \partial \sigma(x)$ such that $h = \lambda d$.

Thus, there exists a level function $\sigma \in \mathcal{O}(f, x)$ such that $d \in \partial \sigma(x)$. By Proposition 2.1(d) and (c), $\langle \lambda d, y - x \rangle$ is a level function. Thus,

$$\lambda d \in \text{int} \ N(x, T_{f_f}(f(x))).$$

The proof is complete. $\square$

In general, a level function is not necessarily affine. It is easy to observe that, if a function has a nonaffine level function, it has an affine level function. Thus, we can prove that a continuous function is quasiconvex if and only if the function has an affine level function. However, we shall see that considering a nonaffine level function is not trivial. Computationally, this will be convenient if we are allowed to use a nonaffine and/or an affine level function in different situations; this is particularly so when the level function method is applied to a fractional program (Ref. 8); see Examples 2.4–2.6. Theoretically, not being confined to affine level functions may make the notion more applicable. At the end of this section, we will show how the notion of level function can be applied to investigate the quasiconvexity of an optimal value function. It would unnecessarily complicate the proof of Theorem 2.1 if we insisted on using affine level functions.

In what follows, we will present some instances which show how to construct in practice a level function for a function quasiconvex at a point.

**Example 2.1.** Suppose that $f: \mathbb{R}^n \to \mathbb{R}$ is a continuous quasiconvex function such that, for every $x \in \mathbb{R}^n$, $\text{cl} \ T_{f_f}(f(x))$ is compact whenever $T_{f_f}(f(x)) \neq \emptyset$. Let $x \in \mathbb{R}^n$ and $h \in \text{int} \ N(x, T_{f_f}(f(x)))$. Then,

$$\sigma(y) = \langle h, y - x \rangle \in \mathcal{O}(f, x).$$

Gromicho (Ref. 9) investigated the elements of the normal cone of a level set using the upper Dini derivative. See Ref. 9 for details.

**Example 2.2.** Let $f: \mathbb{R}^n \to \mathbb{R}$ be a continuous quasiconvex function, and let it be differentiable on $\mathbb{R}^n$. If $\nabla f(x) \neq 0$, then

$$\sigma(y) = \langle \nabla f(x), y - x \rangle$$

is a level function.

Recall that a continuous quasiconvex function is lower subdifferentiable at $x$ if there exists a vector $g \in \mathbb{R}^n$ such that

$$\partial f(y) = \{g \in \mathbb{R}^n: f(y) - f(x) \geq \langle g, y - x \rangle, \text{ for all } y \in T_{f_f}(f(x)) \} \neq \emptyset.$$ (1)
$f$ is said to be boundedly lower subdifferentiable if there exists a constant $L$ such that, for every $x \in \mathbb{R}^n$, 
\[ \partial_- f(x) \cap \{ g \in \mathbb{R}^n : \|g\| \leq L \} \neq \emptyset. \]

Lower subdifferentiability was introduced by Plastria (Ref. 2). It was proved that a continuous real-valued function is boundedly lower subdifferentiable if and only if it is quasiconvex and Lipschitz.

**Example 2.3.** Suppose that $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous and lower subdifferentiable function. Then, for $g \in \partial_- f(x)$,
\[ \sigma(y) = \langle g, y - x \rangle \in \mathcal{C}(f, x). \]

It is possible to use other subdifferentials of a quasiconvex function to construct a level function. See Ref. 10, which is an earlier version of this paper. In what follows, we show that a level function can be constructed also without first-order information.

**Example 2.4.** Consider the following fractional programming problem:
\[ \min \ f(x) = \frac{p(x)}{q(x)}, \tag{2a} \]
\[ \text{s.t.} \ \ x \in C, \tag{2b} \]
where $C$ is a convex compact set, $p: \mathbb{R}^n \rightarrow \mathbb{R}$ is convex, $q: \mathbb{R}^n \rightarrow \mathbb{R}$ is concave and $\min_{x \in C} q(x) > 0$. Let $x \in C$. Then,
\[ \sigma(y) = p(y) - f(x)q(y) \in \mathcal{C}(f, x). \tag{3} \]

**Example 2.5.** Let the functions $f_i: \mathbb{R}^n \rightarrow \mathbb{R}_{+}$ be quasiconvex, and let $m$ be a natural number. Let
\[ f = \max_{i \leq m} f_i(x) \quad \text{and} \quad x \in \text{dom } f. \]
Denote the active index sets of $f$ at $x$ by
\[ I_r(x) = \{ i \in m : f(x) = f_i(x) \}. \]
Assume that, for $i \in I_r(x)$, $\sigma_i \in \mathcal{C}(f_i, x)$. Then,
\[ \sigma = \max_{i \in I_r(x)} \sigma_i \in \mathcal{C}(f, x). \]
Example 2.6. Consider the following generalized fractional programming problem:

\[
\begin{align*}
\text{min} & \quad f(x) \equiv \max_{r \in \mathcal{m}} p_r(x)/q_r(x), \quad (4a) \\
\text{s.t.} & \quad x \in C, \quad (4b)
\end{align*}
\]

where \( p_r : C \to \mathbb{R}, \ r \in \mathcal{m}, \) is convex, \( q_r : C \to \mathbb{R}, \ r \in \mathcal{m}, \) is concave, and \( \min_{x \in C} \min_{r \in \mathcal{m}} q_r(x) > 0. \)

Let \( x \in C. \) Then,

\[
\sigma(p) = \max_{r \in \mathcal{m}} \{ p_r(y) - f(x)q_r(y) \} \in \mathcal{C}(f, x). \quad (5)
\]

Finally, we show how the notion of level function can be applied to investigate the quasiconvexity of some optimal value functions.

Let \( p \in \mathbb{R}^m. \) We consider the following optimal value function:

\[
\begin{align*}
\hat{u}(p) & = \sup \{ u(x) : x \in \mathbb{R}^n, g(p, x) \leq 0 \}, \quad (6) \\
X(p) & = \{ x \in F(p) : \hat{u}(p) = u(x) \}.
\end{align*}
\]

Theorem 2.1. Let \( \hat{v} \) be defined by (6). Assume that, for every fixed \( x \in \mathbb{R}^n, \ g(p, x) \) is a continuous concave function of \( p, \) and that, for every \( \bar{p} \in \mathbb{R}^m, \ X(\bar{p}) \cap \{ x \in \mathbb{R}^n : g(\bar{p}, x) = 0 \} \) is nonempty. Then:

(a) for every \( x_\bar{p} \in X(\bar{p}) \cap \{ x \in \mathbb{R}^n : g(\bar{p}, x) = 0 \}, \)
\[
\sigma(p) = -g(p, x_\bar{p}) \in \mathcal{C}(\hat{v}, \bar{p}). \quad (7)
\]

(b) if for every \( p \in \mathbb{R}^m, \ X(p) \cap \{ x \in \mathbb{R}^n : g(p, x) = 0 \} \neq \emptyset, \) then \( \hat{v} \) is quasi-convex on \( \mathbb{R}^m. \)

Proof. Part (b) follows from part (a) and Proposition 2.2. Thus, we prove only part (a). Note that \( \sigma(p) \) defined in (7) satisfies Definition 2.1(a) and (b). Therefore, it suffices to prove that

\[
T_a(\hat{v}(\bar{p})) \subset T_a(0).
\]

Let \( p \in T_a(\hat{v}(\bar{p})). \) Then, there exists \( x_p \in X(p) \) such that

\[
\hat{v}(p) = u(x_p).
\]
Note that
\[ u(x_p) < u(x_p) = v(\bar{p}), \quad \text{for any } x_p \in X(\bar{p}). \]
Consequently, if \( g(p, x_p) \leq 0 \), then \( x_p \in F(p) \), which contradicts the fact that \( x_p \) is the global maximizer of \( u \) over \( F(p) \). Thus,
\[ g(p, x_p) > 0 \quad \text{and} \quad \sigma(p) < 0. \]

We note that the conclusion of Theorem 2.1 is rather general. Indeed, the assumption that \( X(\bar{p}) \cap \{ x \in \mathbb{R}^n : g(\bar{p}, x) = 0 \} \neq \emptyset \)
depends largely on \( u(x) \); for instance, \( u(x) \) is monotonic. We will not undertake further discussion considering the length of this paper, but further discussion along this line can be found in Ref. 11. This result shows that the notion of level function can be used in theoretical exploration. It also shows the advantage of adopting a nonaffine level function.

To conclude this section, we note that it is possible to consider a level function of an extended real-valued level function. Consequently, most of the results in this section can be restated; see Ref. 10. We confine our discussion to the continuous case only for simplicity and clarity. We also note that the notion of level function can be extended to characterizing a class of abstract convex functions; see also Ref. 10.

3. Level Function Method

In this section, we use the notion of level function to develop a method for the solution of quasiconvex minimization problems. First, we discuss a set of functions which enjoy some properties required throughout this section and later sections.

Let \( \Omega \) be a nonempty compact and convex set. We consider a sequence of continuous functions \( \{ h_i \} \), where \( h_i : \mathbb{R}^n \to \mathbb{R} \), \( i = 1, 2, \ldots \). Let
\[ f_i(x) = \max_{j \neq i} h_j(x) \]
and
\[ U_i = T_{f_i}(0) \cap \Omega. \] (8)

Let
\[ V_i = \int_{\delta U_i} f_i(x) \, dx \] (9)
and
\[ \mathcal{U}_i = - \min_{x \in \text{cl} U_i} f_i(x). \] (10)

The following statements are obvious:
\[ \Omega \supset U_1 \supset \cdots \supset U_i, \] (11)
\[ V_i \leq V_2 \leq \cdots \leq V_i \leq 0, \] (12)
\[ \mathcal{U}_1 \supseteq \mathcal{U}_2 \supseteq \cdots \supseteq \mathcal{U}_i \supseteq 0. \] (13)

The following result establishes a relationship between the value \( \mathcal{U}_i \) and the maximum ball contained in the set \( U_i \).

**Lemma 3.1.** Let \( \{h_i\} \), where \( h_i : \mathbb{R}^n \rightarrow \mathbb{R} \) be a sequence of continuous convex functions. Let \( U_i \) be defined by (8), and let \( \mathcal{U}_i \) be defined by (10). If \( \text{cl} U_i \) contains a closed ball \( B(x^*, r) \), then
\[ \mathcal{U}_i \supseteq c_i r, \]
where
\[ c_i = \min_{j \in i} \|\xi_j\|, \quad \xi_j \in \partial h_j(x^*). \]

**Proof.** Let \( b \in B \), where \( B \) is the unit sphere of \( \mathbb{R}^n \). Then,
\[ x^* + rb \in B(x^*, r) \subset \text{cl} U_i, \]
and for any \( j \in \tilde{i} \),
\[ h_j(x^* + rb) \leq 0. \]
Since \( h_j \) is convex,
\[ h_j(x^* + rb) \geq h_j(x^*) + r(\xi_j, b), \]
where \( \xi_j \in \partial h_j(x^*) \). Thus,
\[ h_j(x^*) \leq -r(\xi_j, b). \]
Since \( b \in B \) is arbitrary, then for \( j \in \tilde{i} \),
\[ h_j(x^*) \leq -r\|\xi_j\|. \]
Consequently,
\[ -\mathcal{U}_i \leq \max_{j \in i} h_j(x^*) \leq -c_i r. \]
The proof is complete. \( \square \)
Property (P). Let $\Omega$ be a compact convex set. A sequence of functions $\{h_i\}$, where $h_i: \mathbb{R}^n \to \mathbb{R}$, is said to satisfy Property (P) if the following conditions hold:

(a) for every $i$, $h_i$ is convex and $\{h_i\}$ is uniformly Lipschitz on $\Omega$ in the sense that there exists a positive constant $L$ such that, for all $i$,

$$|h_i(x) - h_i(y)| \leq L\|y - x\|, \quad \forall x, y \in \Omega;$$

(b) for $i = 1$, there exists $x_1 \in \Omega$ such that $h_1(x_1) = 0$.

For $i \geq 1$, let

$$f_i(x) = \max_{j \leq i} h_j(x),$$

and let $x_{i+1}$ be a minimizer of $f$ over $\Omega$, that is,

$$x_{i+1} \in \arg\min_{x \in \Omega} f_i(x).$$

Then,

$$f_i(x_{i+1}) < 0,$$

and

$$h_{i+1}(x_{i+1}) = 0.$$

Note that $h_{i+1}$ does not necessarily take the zero value at the other minimizers of $f_i$ over $\Omega$.

Lemma 3.2. Let $\{h_i(x)\}$ be a sequence of functions which satisfy Property (P). Then,

$$\lim_{i \to \infty} \Omega_i = 0.$$

Proof. Let $V_i$ be defined by (9). Then,

$$V_i - V_{i-1} = \int_{U_i} f_i(x) dx - \int_{U_{i-1}} f_{i-1}(x) dx.$$ 

By part (b) of Property (P), $x_{i+1} \in U_i$. Thus, $U_i$ is nonempty, and by (11), $\{\text{cl } U_i\}$ is monotonic and bounded. Let

$$P = \bigcap_{i=0}^{\infty} \text{cl } U_i.$$
Then, $P \neq \emptyset$. Moreover,
\[
V_i - V_{i-1} = \int_P [f_i(x) - f_{i-1}(x)] \, dx + \int_{(cl \ U_i) \setminus P} [f_i(x) - f_{i-1}(x)] \, dx
- \int_{d(U_{i-1} \cap dU_i)} f_{i-1}(x) \, dx.
\] (14)

Let
\[
\delta_1^i = \int_P [f_i(x) - f_{i-1}(x)] \, dx,
\]
\[
\delta_2^i = \int_{(cl \ U_i) \setminus P} [f_i(x) - f_{i-1}(x)] \, dx,
\]
\[
\delta_3^i = \int_{d(U_{i-1} \cap dU_i)} f_{i-1}(x) \, dx.
\]

By part (a) of Property (P), $\{h_i(x)\}$ is uniformly Lipschitz on $\Omega$. Then, $\{f_i(x)\}$ is also uniformly Lipschitz on $\Omega$, and consequently $\{f_i(x)\}$ is uniformly bounded. Note also that the Lebesgue measure of $(cl \ U_i) \setminus P$ and $cl \ U_{i-1} \setminus cl \ U_i$ tends to zero. Then,
\[
\lim_{i \to \infty} \delta_p^i = 0, \quad \text{for } p = 2, 3.
\] (15)

Combining (12)–(15), we have
\[
\delta_1^i \to 0, \quad \text{as } i \to \infty.
\]

Let
\[
\eta_i(x) = f_i(x) - f_{i-1}(x).
\]

Then,
\[
\eta_i(x) \geq 0.
\]

Thus, $\delta_1^i \to 0$ is equivalent to
\[
\lim_{i \to \infty} \int_P \eta_i(x) \, dx = 0.
\]

Note that $\{f_i(x)\}$ is uniformly bounded on $\Omega$; it follows that $\eta_i(x)$ is uniformly bounded and uniformly Lipschitz on $\Omega$. Thus, for every $x \in P$,
\[
\lim_{i \to \infty} \eta_i(x) = 0.
\]
Note that $x_{i+1} \in U_i \subset \text{cl } U_i$. Then, the sequence $\{x_i\}$ is bounded and every accumulation point $x^*$ of $\{x_i\}$ is located in $P$. Thus,

$$\eta_i(x^*) \rightarrow 0, \quad \text{as } i \rightarrow \infty.$$ 

Let $\{x_{i_0}\}$ be a subsequence of $\{x_i\}$ such that

$$x_{i_0} \rightarrow x^*, \quad \text{as } i_0 \rightarrow \infty.$$ 

By Lipschitz continuity, there exists a constant $M$ such that

$$|\eta_{i_0}(x_{i_0}) - \eta_{i_0}(x^*)| \leq M||x_{i_0} - x^*|| \rightarrow 0.$$ 

Thus,

$$\eta_{i_0}(x_{i_0}) \rightarrow 0, \quad \text{as } i_0 \rightarrow \infty.$$ 

Since

$$\eta_i(x_i) = f_i(x_i) - f_{i-1}(x_i) = \max\{0, -f_{i-1}(x_i)\} \quad \text{and} \quad -f_{i-1}(x_i) > 0,$$

it follows that

$$\eta_i(x_i) = -f_{i-1}(x_i) = \mathcal{U}_{i-1}.$$ 

Hence,

$$\lim_{i_0 \rightarrow \infty} \mathcal{U}_{i_0} = \lim_{i_0 \rightarrow \infty} \eta_{i_0}(x_{i_0}) = 0.$$ 

Since $\{\mathcal{U}_i\}$ is monotonically decreasing, we have

$$\lim_{i \rightarrow \infty} \mathcal{U}_i = 0.$$ 

The proof is complete. $\square$

Note that the class of functions enjoying Property (P) is very close to the functions generated by the outer approximation procedure (Refs. 5–7). The difference is that here $h_i$ is not necessarily linear and that the set $T_{f_j}(0) \cap \Omega$ instead of the set $S_{f_j}(0) \cap \Omega$ is used. Moreover, a cut in the outer approximation method does not exist for a quasiconvex function at a point where the function value is constant in a neighborhood of the point.

3.1. Exact Level Function Method. Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuous quasiconvex function, and let $C$ be a nonempty compact and convex set. We consider the following minimization problem:

$$(P) \quad \min f(x),$$

$$\text{s.t. } x \in C.$$
Using the notion of level function, we propose an algorithm which requires the computation of the level functions instead of the support functions for solving Problem (P).

Algorithm 3.1. Exact Level Function Method.

Step 1. Select a starting point \( x_0 \in C \), set \( i = 0 \);
Step 2. Calculate a level function \( \sigma_i(x) \) of \( f \) at \( x_i \) and set

\[
\sigma_i(x) = \max \{ \sigma_{i-1}(x), \sigma_i(x) \},
\]

where \( \sigma_{-1}(x) \equiv -\infty \). Let
\[
x_{i+1} \in \arg \min_{x \in C} \sigma_i(x)
\]

and
\[
\Delta(i) = -\sigma_i(x_{i+1}).
\]

Step 3. If \( \Delta(i) \leq 0 \), stop; otherwise, set \( i := i + 1 \); go to Step 2.

Before presenting a convergence analysis, we make a few comments on the proposed algorithm. We note that differently from the cutting plane algorithm, this algorithm does not use \( \sigma_i(x) \) to approximate the original function \( f \); instead, it uses the strict level set \( T_{\sigma_i}(0) \) to approximate a level set of \( f \). The minimization of \( \sigma_i(x) \) over \( C \) is relatively easy, since it is a standard convex minimization problem. We also note that, as we discussed in Section 2, the level functions of quasiconvex functions can be constructed through first-order information such as generalized subgradients or other information based on function values, depending on which information can be obtained more easily. In this sense, this algorithm is more flexible than a cutting plane algorithm.

Now, we analyze the convergence of Algorithm 3.1. First, we look at the case where the algorithm terminates in a finite number of iterations.

Proposition 3.1. If Algorithm 3.1 terminates at iteration \( i \), then the global minimum of \( f \) over \( C \) is attained at \( x_i \).

Proof. Note that \( \Delta(j) > 0 \), for \( j \leq i - 1 \). By (17),
\[
\Delta(i) = -\min_{x \in C} \max \{ \sigma_{i-1}(x), \sigma_i(x) \}.
\]

Since \( \Delta(i) \leq 0 \), then
\[
\max \{ \sigma_{i-1}(x), \sigma_i(x) \} \geq 0, \quad \text{for all} \; x \in C.
\]
Note that
\[ \Delta(i-1) = -\sigma_{i-1}(x_i) > 0 \]
and that \( \sigma_i(x) \) is continuous. Then, there exists a \( \delta > 0 \) such that, for all \( x \in C \cap S(x_i, \delta) \),
\[ \sigma_{i-1}(x) < 0. \]
Thus,
\[ \sigma_{x_i}(x) \equiv 0, \quad \text{for all } x \in C \cap S(x_i, \delta). \]
Note also that \( \sigma_{x_i}(x) \) is convex and
\[ \sigma_{x_i}(x_i) = 0. \]
Then, \( x_i \) is a global minimizer of \( f \) over \( C \).

In what follows, we analyze the case when \( \{x_i\} \) generated by Algorithm 3.1 is an infinite sequence. Let
\[ T_i = T_0(0) \cap C. \]

**Proposition 3.2.** Let \( \sigma_i \) and \( T_i \) be given by (16) and (18). Then, the following statements hold:

(a) the sequence of level functions \( \{\sigma_{x_i}\} \) generated by the algorithm satisfies Property (P);

(b) \( T_i \subset T_{i-1} \subset \cdots \subset T_1 \subset C \);

(c) if \( T_i \neq \emptyset \), then \( \text{cl } T_i = S_0(0) \cap C \);

(d) if \( T_i \neq \emptyset \), then \( x_i \in \text{cl } T_i \);

(e) if \( T_i \neq \emptyset \), then
\[ X^* \cap \text{cl } T_i \neq \emptyset, \]
where \( X^* \) denotes the set of global minimizers of \( f \) over \( C \).

**Proof.** Parts (a) and (b) are obvious. Part (c) follows from the fact that \( \sigma_i \) is a continuous convex function and \( C \) is a compact set. Part (d) follows from (a) and (c). Indeed, from part (a), we see that \( \sigma_{i-1}(x_i) < 0 \). Thus,
\[ \sigma_i(x_i) = \sigma_{x_i}(x_i) = 0, \]
which implies
\[ x_i \in S_0(0) \cap C = \text{cl } T_i. \]
We now prove part (e). We proceed by induction. For $i = 1$, if $f(x_1) > f(x^*)$, then
\[ x^* \in T_I(f(x_1)) \cap C \subset T_I; \]
otherwise, $x_1$ is also a global minimizer of $f$ over $C$ and, from part (d),
\[ x_1 \in \text{cl} \ T_I. \]
This proves (19) for $i = 1$. Suppose that, for $i \in \mathbb{N}$, (19) holds. By part (d),
\[ x_{k+1} \in \text{cl} \ T_{k+1}. \]
If $x_{k+1} \in X^*$, then (19) holds for $i = k+1$. Otherwise, $f(x_{k+1}) > f(x^*)$, for all $x^* \in \text{cl} \ T_k \cap X^*$, which is nonempty by assumption. By the definition of level function, $\sigma_{x_{k+1}}(x^*) < 0$. Thus, $\sigma_{k+1}(x^*) \leq 0$ and $x^* \in S_{k+1}(0) \cap C$. By part (c),
\[ x^* \in \text{cl} \ T_{k+1}(0) = \text{cl} \ T_{k+1}. \]
\[ \square \]

**Corollary 3.1.** Let $\Delta(i)$ be defined by (17). Then, the following statements hold:

(a) if $T_i \neq \emptyset$, then $\Delta(i) = -\min_{x \in \text{cl} \ T_i} \sigma_i(x)$;
(b) $\Delta(1) \geq \Delta(2) \geq \cdots \geq \Delta(i)$; hence, if $\Delta(i) > 0$, then $\Delta(j) > 0$ for $j < i$.

The following result establishes a relationship between $\Delta(i)$ and the radius of the maximum ball contained in $T_I(f(x_i)) \cap C$.

**Proposition 3.3.** Let $s$ be the largest number from the set $\bar{I}$ such that $f(x_s) = \min_{j \in \bar{I}} \{f(x_j)\}$. Assume that $T_I(f(x_s)) \cap C$ contains a closed ball $B(x^*, r)$. Then,
\[ \Delta(i) \geq c_i r, \]
where
\[ c_i = \min_{j \in \bar{I}} \|\xi_j\|, \quad \xi_j \in \partial \sigma_j(x^*). \]

**Proof.** By definition, for $j \in \bar{I}$, we have
\[ T_I(f(x_j)) \cap C \subset T_{\sigma_{x_j}(0)}(0) \cap C, \]
\[ T_I(f(x_j)) \subset T_J(f(x_j)) = T_I(f(x_j)). \]
Hence,
\[ T_I(f(x_j)) \cap C \subset T_I. \]
The rest is straightforward using Lemma 3.1. \[ \square \]
Note that, in Proposition 3.3, $c_i$ depends on the level functions of $f$ at the points $\{x_j : j \in \bar{l}\}$. However, if $c_i$ is positive uniformly and if $\Delta(i)$ decreases to zero as $i$ tends to infinity, then the volume of the intersection of the $i$ level sets shrinks to zero. A sufficient condition ensuring that $c_i > 0$ is that

$$\frac{c_i}{G_{\inf}} \frac{\max_{x \in C} \min_{\xi \in \partial \sigma_x(x)}}{H_0^i} > 0.$$  \hspace{1cm} (20)

A specific example is that

$$\sigma_{x_j}(x) = \langle a_j, x - x_j \rangle, \quad j \in \bar{l},$$

are linear functions. In such a case, if the sequence $\{||a_j||\}$ is uniformly bounded below by a positive constant, then $c' > 0$.

**Proposition 3.4.** Let $\{x_i\}$ be an infinite sequence generated by Algorithm 3.1. Assume that one of the following conditions is satisfied:

(a) there exists a positive constant $\alpha$ such that, for every $x_j \in \{x_i\}$, there exists a level function $\sigma_{x_j} \in (f, x_j)$ such that, for every $\epsilon > 0$,

$$x \in T_{\sigma_j}(0) \quad \text{and} \quad \sigma_{x_j}(x) \geq -\epsilon \text{ implies } f(x) \geq f(x_j) - \alpha \epsilon;$$

(b) (20) holds for the sequence $\{x_i\}$.

If $\lim_{i \to \infty} \Delta(i) = 0$, then there exists a subsequence of $\{x_i\}$ converging to a global minimizer of $f$ over $C$.

**Proof.** First, we assume that condition (a) is satisfied. Since

$$\lim_{i \to \infty} \Delta(i) = 0,$$

for every $\epsilon > 0$, there exists $i$ such that

$$\min_{x \in C} \sigma_i(x) \leq -\epsilon.$$

Let $x'$ denote a global minimizer of $f$ over $C$. We have

$$\sigma_{x'}(x') \leq -\epsilon.$$

By definition, there exists $j \in \bar{l}$, such that

$$\sigma_{x_j}(x') \leq -\epsilon.$$

By assumption, there exists $\alpha > 0$ such that

$$f(x') \geq f(x_j) - \alpha \epsilon.$$

The conclusion follows since $f$ is continuous and $C$ is compact.
Now, assume that condition (b) is satisfied. Given \( \varepsilon > 0 \), there exists \( i \) such that \( \Delta(i) < \varepsilon \). Let \( s \) the largest number from \( i \) such that

\[ f(x_s) = \min\{f(x_j) : j \in \overline{i}\}. \]

If \( s = i \), then by Proposition 3.2(d),

\[ x_s \in \text{cl } T_i. \]

If \( s < i \), by definition,

\[ f(x_j) > f(x_s), \quad \text{for } s + 1 \leq j \leq i. \]

Thus,

\[ x_s \in T_s(f(x_{s+1})) \cap C \subset T_{\sigma_{s+1}}(0) \cap C. \]

Moreover, by Proposition 3.2(d),

\[ x_s \in \text{cl } T_s. \]

Hence,

\[ x_s \in \text{cl } T_{s+1}. \]

By induction, we can eventually prove that

\[ x_s \in \text{cl } T_i. \]

Let \( x' \) be a global minimizer of \( f \) over \( C \) such that \( x' \in \text{cl } T_i \). The existence of such an \( x' \) is guaranteed by Proposition 3.2(e). Thus, we have that

\[ x_s, x' \in \text{cl } T_i. \]

By Proposition 3.3, we have that

\[ \|x_s - x'\| \leq 2\Delta(i)/c' < 2\varepsilon/c'. \]

The conclusion follows, since \( \varepsilon \) can be made arbitrarily small. \( \Box \)

The conditions of Proposition 3.4 are not straightforward. Note that we have discussed condition (b) following Proposition 3.3. Here, we comment only on condition (a). Consider Example 2.4. Assume that, at point \( x_k \), \( \sigma_{x_k}(x) \) in (3) is greater than or equal to \( -\varepsilon \). Then,

\[ f(x) \geq f(x_k) - \varepsilon/q(x). \]

By assumption, there exists a positive constant \( \delta \) such that

\[ q(x) \geq \delta, \quad \text{for all } x \in C. \]

Thus,

\[ f(x) \geq f(x_k) - \varepsilon/\delta. \]
Also, it can be verified easily that a lower subdifferentiable function satisfies condition (a).

**Theorem 3.1.** Let \( f: \mathbb{R}^n \to \mathbb{R} \) be a continuous quasiconvex function, and let \( \{ x_i \} \) be an infinite sequence generated by Algorithm 3.1. Assume that the sequence of level functions \( \{ \sigma_{x_i}(x) \} \) is uniformly Lipschitz on \( C \) and that the conditions of Proposition 3.4 are satisfied. Then, \( \lim_{i \to \infty} \Delta(i) = 0 \) and there exists a subsequence of \( \{ x_i \} \) converging to a global minimizer of \( f \) over \( C \).

**Proof.** Under the assumptions, the sequence of level functions \( \{ \sigma_{x_i}(x) \} \) generated by Algorithm 3.1 satisfies Property (P) for \( \Omega = C \). The rest is straightforward using Lemma 3.2 and Proposition 3.4.

Consider again Example 2.4. We can apply Algorithm 3.1 to solve (3) with level functions identified by (3). If \( \Delta(i) = 0 \), then the global minimum of \( f \) over \( C \) is attained. Otherwise,

\[
\sigma_{x_i}(x_{i+1}) < 0,
\]

which implies that

\[
f(x_{i+1}) < f(x_i).
\]

Thus,

\[
\sigma_{x_{i-1}}(x) > \sigma_{x_i}(x) \quad \text{and} \quad \sigma_{x_{i+1}}(x) = \sigma_{x_{i-1}}(x) = p(x) - f(x_{i-1})q(x).
\]

Consequently, the algorithm reduces to the Dinkelback algorithm for fractional programs. It is well known that the latter enjoys superlinear convergence under certain conditions; see Ref. 12.

Analogously, we can apply Algorithm 3.1 to solve (5) in Example 2.6 by taking level functions in the form (5). Consequently we obtain the generalized Dinkelbach algorithm; see also Ref. 12.

Theoretically, Algorithm 3.1 requires an infinite convex minimization process at each iteration if a level function is not affine. In some cases, it may be preferable to employ a finite process to update an iterate. Let \( \sigma_{x_i}(y) \) be a level function of \( f \) at \( x \in \text{dom} f \), and let \( g \) be a subgradient of \( \sigma_{x_i} \) at \( x \). Then, for all \( y \in \mathbb{R}^n \),

\[
\sigma_{x_i}(y) \geq \langle g, y - x \rangle.
\]

Obviously \( y \mapsto \langle g, y - x \rangle \) is a level function of \( f \) at \( x \). Note also that, if \( g \neq 0 \), then \( \langle g, y - x \rangle/\|g\| \) is also a level function of \( f \) at \( x \). The difference is that
the algorithm based on the latter can satisfy easily the conditions of Theorem 3.1. Applying these ideas to Algorithm 3.1, we obtain the following algorithm.

**Algorithm 3.2.** Exact Scaled Linearized Level Function Method.

Step 1. Select a starting point $x_0 \in C$; set $i := 0$;

Step 2. Calculate a level function $\sigma_{i}(x)$ and a subgradient $g_i \in \partial \sigma_{i}(x)$. If $g_i = 0$, stop. Otherwise, set $\sigma'_{i}(x) = (g_i, y - x)/\|g_i\|$, $\sigma_i(x) = \max\{\sigma'_{i-1}(x), \sigma_{i-1}(x)\}$, where $\sigma'_{i+1}(x) \equiv -\infty$. Let $x_{i+1} \in \arg\min_{x \in C} \sigma'_{i}(x)$.

Step 3. If $\Delta(i) \leq 0$, stop; otherwise, set $i := i + 1$; go to Step 2.

**Theorem 3.2.** Let $f: \mathbb{R}^n \to \mathbb{R}$ be a continuous quasiconvex function, and let $\{x_i\}$ be an infinite sequence generated by Algorithm 3.2. Then, $\lim_{i \to \infty} \Delta(i) = 0$ and there exists a subsequence of $\{x_i\}$ converging to a global minimizer of $f$ over $C$.

The result is straightforward using Theorem 3.1.

3.2. Projected Level Function Method. We continue to discuss the level function method for solving Problem (P). Note that both Algorithm 3.1 and Algorithm 3.2 take a minimizer of $\sigma_i(x)$ as the next iterate. An obvious drawback is that we are unable to predict the maximum number of iterations required to reduce $\Delta(i)$ to a prescribed precision. Thus, these algorithms may converge slowly for some problems. A possible solution is to modify the algorithms by updating an iterate using a projection of the current point to a level set of $\sigma_i(x)$. This may also generalize the algorithms, since the exact minimization is indeed a projection on the set of minima. The projection idea belongs to Lemarechal, Nemirovskii, and Nesterov, who applied it first to convex programming in Ref. 3. Nemirovskii (Ref. 4) made a significant extension to the general quasiconvex setting. The following algorithm can be regarded as a generalization of the Nemirovskii algorithm from Ref. 4.

**Algorithm 3.3.** Projected Level Function Method.

Step 1. Let $\epsilon > 0$ be a constant, and select a constant $\lambda \in (0, 1)$ and a starting point $x_0 \in C$; set $i = 0$;
Step 2. Calculate a level function \( \sigma_{x_i}(x) \) of \( f \) at \( x_i \), and set
\[
\sigma_{x_i}(x) = \max\{\sigma_{x_{i-1}}(x), \sigma_{x_i}(x)\},
\]
where \( \sigma_{x_{i-1}}(x) \equiv -\infty \). Let
\[
x_i = \arg\min\{f(x_j) : j \in \overline{1} \} \quad \text{and} \quad x_{i+1} \in \pi\{x_i, Q_i\},
\]
where
\[
Q_i = \{x \in C : \sigma_{x_i}(x) \leq -\lambda \Delta(i)\}, \quad \Delta(i) = \min_{x \in C} \sigma_{x_i}(x),
\]
and \( \pi(x, Q) \) is the Euclidean projection of the point \( x \) on a set \( Q \).

Step 3. If \( \Delta(i) \leq \epsilon \), stop; otherwise, set \( i := i + 1 \); go to Step 2.

When \( \lambda = 1 \), \( Q \) becomes the set of minimizers of \( \sigma_x \) over \( C \). Consequently, Algorithm 3.3 collapses to Algorithm 3.1. In what follows, we assume that \( \lambda \in (0, 1) \).

**Theorem 3.3.** Let \( \{x_i\} \) be generated by Algorithm 3.3. Assume that \( f: \mathbb{R}^n \to \mathbb{R} \) is a continuous quasiconvex function and that the sequence of level functions \( \{\sigma_{x_i}(x)\} \) is uniformly Lipschitz with constant \( M \). Then,
\[
\Delta(i) \leq \epsilon, \quad \text{for} \quad i > M^2 d^2 \epsilon^{-2} \lambda^{-2}(1 - \lambda^2)^{-1},
\]
where \( d \) is the diameter of \( C \) and \( \epsilon \) is given in Algorithm 3.3.

**Proof.** Assume that, for all \( i \leq N \), we have \( \Delta(i) > \epsilon \). Let \( \{1, \ldots, N\} \) be classified into the groups \( I_1, \ldots, I_k \) such that
\[
j_i = N, \\
i \in I_i \setminus I_{i-1}, \quad l \in \overline{k}, \quad I_0 = \emptyset, \\
\Delta(j_i) \geq \lambda \Delta(i), \quad \forall i \in I_i, \\
j_i = \max_{i \in b} i.
\]
Let \( u_i \) be such that
\[
\sigma_{j_i}(u_i) = -\Delta(j_i).
\]
Then, for all \( i \in I_i \),
\[
\sigma_i(u_i) \leq \sigma_{j_i}(u_i) = -\Delta(j_i) \leq -\lambda \Delta(i).
\]
Thus, \( u_i \) is a common point of
\[
Q_i = \{x : \sigma_x(x) \leq -\lambda \Delta(i)\}.
\]
Thus,
\[ \tau_{i+1} \equiv \|x_{i+1} - u_i\|^2 \leq \tau_i - \|x_i - x_{i+1}\|^2. \]

Since
\[ \sigma_i(x_i) - \sigma_i(x_{i+1}) \equiv \lambda \Delta(i), \]
from the Lipschitz continuity of \( \sigma_i \), we have that
\[ \|x_i - x_{i+1}\| \leq M^{-1}|\sigma_i(x_i) - \sigma_i(x_{i+1})| \leq M^{-1}\lambda \Delta(i). \]

Consequently,
\[ \tau_{i+1} \leq \tau_i - M^{-2}\lambda^2 \Delta(i)^2 \leq \tau_i - M^{-2}\lambda^2 \Delta(j_i)^2. \]

Note that
\[ 0 \leq \tau_i \leq d^2 \quad \text{and} \quad \Delta(j_{i-1}) < \lambda \Delta(j_i). \]

Then, the number \( N_i \) of elements in \( I_i \) satisfies
\[
N_i \leq M^2\lambda^{-2}d^2\Delta(j_i)^{-2} \\
< M^2\lambda^{-2}d^2\lambda^2\Delta(j_{i-1})^{-2} \\
< \cdots < M^2\lambda^{-2}d^2\Delta(N)^{-2}\lambda^{2l-1} \\
< M^2\lambda^{-2}d^2\epsilon^{-2}\lambda^{2l-1}.
\]

Thus,
\[
N = \sum_{i=1}^{k} N_i \\
\leq M^2\lambda^{-2}d^2\epsilon^{-2} \sum_{i=1}^{k} \lambda^{2l-1} \\
\leq M^2\lambda^{-2}d^2\epsilon^{-2}(1 - \lambda^2)^{-1}.
\]

The proof is complete. \( \square \)

In comparison with Theorem 3.1, we note that Theorem 3.3 is stronger in that we give an estimation of the maximum number of iterations required to reduce \( \Delta(i) \) to a prescribed precision.

4. Exact Level Function Method for Variational Inequality Problems

In this section, we apply the idea of the level function method to solve a variational inequality problem. Let \( F: \mathbb{R}^n \rightarrow \mathbb{R}^n \) be a continuous mapping.
and let $C$ be a nonempty, compact, and convex subset of $\mathbb{R}^n$. We consider the following variational inequality problem: find a point $x^* \in C$ such that

$$\text{(VIP)} \quad \langle F(x^*), x - x^* \rangle \geq 0, \quad \forall x \in C.$$  \hfill (21)

Recall that a mapping $F: \mathbb{R}^n \to \mathbb{R}^n$ is said to be monotone on $C$ if, for every $x \in C$,

$$\langle F(y) - F(x), y - x \rangle \geq 0, \quad \forall y \in C.$$  

Let $x \in C$ and define

$$l_x(y) = \langle F(x), y - x \rangle.$$  \hfill (22)

It is easy to verify that, if $F$ is monotone on $C$ and if $x^*$ is a solution of (21), then for every $x \in C$,

$$l_x(x^*) = 0.$$  

Thus, $x^*$ is contained in level set $S_x(0)$ for every $x \in C$. We call $l_x$ defined as in (22) a level function of (VIP). Note that $l_x$ here has nothing to do with a level set of a function, nor does it satisfy the conditions in the definition of level function in the previous section. We use the notion only because it will play a role similar to a level function of a quasiconvex function.

Let $x_1 \in C$ and, for $j = 1, \ldots, i - 1$, let

$$x_{j+1} \in \arg \min_{y \in C} l_{x_j}(y) \text{ with } l_{x_j}(y) = \max_{s \in j} l_s(y).$$  \hfill (23)

Let

$$\Delta(i) = -l_{x_{i+1}} = \min_{y \in C} l_s(y).$$  \hfill (24)

Then,

$$\Delta(1) \geq \Delta(2) \geq \cdots \geq \Delta(i).$$

Hence, if $\Delta(i) > 0$, then $\Delta(j) > 0$ for $j < i$.

**Proposition 4.1.** Let $\Delta(i)$ be defined by (24). If $\Delta(j) > 0$, for $j = 1, \ldots, i - 1$, and $\Delta(i) \equiv 0$, then $x_i$ is a solution of (VIP).

**Proof.** Since

$$\Delta(i - 1) > 0,$$

then for $j = 1, \ldots, i - 1$,

$$l_{x_j}(x_i) < 0,$$
Thus, there exists $\delta > 0$ such that, for all $y$ in the closed ball $B(x_i, \delta)$,

$$l_{x_i}(y) < 0, \quad j = 1, \ldots, i - 1.$$ 

Suppose to the contrary that there exists $x' \in B(x_i, \delta) \cap C$ such that $l_{x_i}(x') < 0$. Then,

$$\Delta(i) \geq -l_i(x') > 0,$$

which contradicts the assumption. Thus, for all $y \in B(x_i, \delta) \cap C$,

$$l_{x_i}(y) \geq 0.$$ 

Note that

$$l_{x_i}(x_i) = 0.$$ 

Then, $x_i$ is a local minimizer of $l_{x_i}(y)$ over $C$. Since $l_{x_i}(y)$ is a linear function, then $x_i$ is also a global minimizer over $C$; that is,

$$l_{x_i}(y) \geq 0, \quad \text{for all } y \in C,$$

which shows that $x_i$ is a solution of (VIP). \qed

**Proposition 4.2.** Let $x_j \in C, j \in \mathbb{T}$; let $l_{x_j}(x)$ be a level function of (VIP) at $x_j$, and let $T_i = T_i(0) \cap C$. Assume that $c_i$ contains a ball of radius $r$. Then,

$$\Delta(i) \geq c_ir,$$

where

$$c_i = \min_{j \in i} \|F(x_j)\|.$$ 

**Proof.** The result is obvious by using Lemma 3.1. \qed

**Algorithm 4.1.** Exact Level Function Method for Variational Inequality Problem.

Step 1. Select a starting point $x_0 \in C$; set $i := 0$;
Step 2. Calculate a level function $l_{x_i}(x)$ of (VIP) in (21), and set

$$l_i(x) = \max\{l_{x_{i-1}}(x), l_{x_i}(x)\}, \quad \text{where } \sigma_{-1}(x) \equiv -\infty.$$ 

Let

$$x_{i+1} \in \arg\min_{x \in C} l_i(x).$$

Step 3. If $\Delta(i) \leq 0$, stop; otherwise, set $i := i + 1$; go to Step 2.
Proposition 4.3. If Algorithm 4.1 terminates at the \(i\)th iteration, then \(x_i\) is a solution of (VIP).

**Proof.** The conclusion follows from Proposition 4.1. \(\Box\)

Proposition 4.4. Let \(\{x_i\}\) be an infinite sequence generated by Algorithm 4.1. Assume that \(F\) is monotone on \(C\). If

\[
\inf_{i \geq 0} \|F(x_i)\| = c > 0,
\]

and if \(\lim_{i \to \infty} \Delta(i) = 0\), then \(\{x_i\}\) converges to a solution point of (VIP).

**Proof.** Given \(\epsilon > 0\), by assumption, there exists \(i_0\) such that, for \(i \geq i_0\), \(\Delta(i) \leq \epsilon\). Since \(x_{i_0+1}\) is a global minimizer of \(l_{i_0}\) over \(C\),

\[
x_{i_0+1} \in \text{cl} \left( T_{h_0}(0) \cap C \right)
\]

and, for every solution \(x'\) of (VIP),

\[
x' \in \text{cl} \left( T_{h_0}(0) \cap C \right).
\]

By Proposition 4.2, we have that

\[
\|x_{i_0+1} - x'\| \leq 2r \leq 2\Delta(i)/c_{i_0} < 2\epsilon/c_{i_0}.
\]

Therefore, the conclusion follows, since \(c_{i_0} \geq c > 0\) and \(\epsilon\) can be made arbitrarily small. \(\Box\)

Theorem 4.1. Let \(\{x_i\}\) be an infinite sequence generated by Algorithm 4.1. Assume that \(\inf_{i \geq 0} \|F(x_i)\| > 0\) and that \(F\) is monotone on \(C\). Then, \(\{x_i\}\) converges to a solution of (VIP).

**Proof.** Under the assumptions, the sequence of level functions \(\{l_{i_0}(x)\}\) generated by Algorithm 4.1 satisfies Property (P) for \(\Omega = C\). The rest is straightforward using Lemma 3.2 and Proposition 4.4. \(\Box\)

5. Numerical Experiments

In this section, we report some results of numerical tests on Algorithms 3.1–3.3 and Algorithm 4.1 for some problems arising in fractional programming and variational inequalities. The algorithms were implemented in Matlab 4.2c.2 installed in a PC. Standard Matlab functions for solving quadratic programming and linear programming were used to solve the projection subproblem in Algorithm 3.3 and to compute \(\Delta(i)\). Comparisons are
Example 5.1. Consider the fractional programming problem (3) in Example 2.4. Let
\[ p(x) = \max \{ x_1^2 + x_2^2; (2 - x_1)^2 + (2 - x_2)^2; 2e^{-x_1 - x_2} \}, \]
\[ q(x) = c_1 x_1 + c_2 x_2 + 1, \]
where \( c_1 \) and \( c_2 \) will be specified below. Let
\[ C = \{ x \in \mathbb{R}^2; 0 \leq x_1; 0 \leq x_2; x_1 + x_2 \leq 3 \}. \]

The function \( p/q \) is nonsmooth and quasiconvex on \( C \). We tested Algorithms 3.1–3.3 for this example. In the implementation of Algorithm 3.1, we used (3) to compute the level function at each iteration. As such, the algorithm reduces to the Dinkelbach algorithm. We applied the Shor algorithm to the solution of the nonsmooth convex minimization subproblem at Step 2 of Algorithm 3.1. The stopping rule for the Shor algorithm is \( ||x_{k+1} - x_k|| \leq 0.005 \), where \( x_k \) and \( x_{k+1} \) are two consequent approximated solutions of the subproblem. In the implementation of Algorithm 3.3, we computed \( \sigma(x) \) at the \( i \)th iteration as follows.
\[ \sigma_i(x) = \langle g_i, x - y \rangle, \]
where \( g_i \) is a normalized subgradient of the convex function \( p(x) - f(x_i)q(x) \) at \( x_i \). We tested the algorithm for different values of the projection parameter \( \lambda \). When \( \lambda = 1 \), the algorithm reduces to Algorithm 3.2. We display the numerical results in Table 1. In the table, \( c = (c_1, c_2)^T \) is a vector representing the parameters in the function \( q(x) \) and \( \lambda \) is the projection parameter in Algorithm 3.3. At the column of the parameter \( \lambda \), the symbol Dk

<table>
<thead>
<tr>
<th>( c )</th>
<th>( \lambda )</th>
<th>NIT</th>
<th>( \Delta )</th>
<th>( f )</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (0, 0)^T )</td>
<td>Dk</td>
<td>12</td>
<td>1.3869E−4</td>
<td>1.9589</td>
<td>( (1.1471, 0.8892)^T )</td>
</tr>
<tr>
<td>( (0, 0)^T )</td>
<td>0.3</td>
<td>14</td>
<td>9.3985E−4</td>
<td>1.9537</td>
<td>( (1.1508, 0.8907)^T )</td>
</tr>
<tr>
<td>( (0, 0)^T )</td>
<td>1.0</td>
<td>36</td>
<td>9.3927E−4</td>
<td>1.9552</td>
<td>( (1.1669, 0.8770)^T )</td>
</tr>
<tr>
<td>( (2, 1)^T )</td>
<td>Dk</td>
<td>7</td>
<td>6.9841E−4</td>
<td>0.4641</td>
<td>( (1.2505, 0.8088)^T )</td>
</tr>
<tr>
<td>( (2, 1)^T )</td>
<td>0.3</td>
<td>11</td>
<td>8.2432E−4</td>
<td>0.4618</td>
<td>( (1.2009, 0.8497)^T )</td>
</tr>
<tr>
<td>( (2, 1)^T )</td>
<td>1.0</td>
<td>33</td>
<td>8.8342E−4</td>
<td>0.4615</td>
<td>( (1.2428, 0.8144)^T )</td>
</tr>
<tr>
<td>( (20, 10)^T )</td>
<td>Dk</td>
<td>47</td>
<td>9.0994E−4</td>
<td>0.0589</td>
<td>( (1.2665, 0.7918)^T )</td>
</tr>
<tr>
<td>( (20, 10)^T )</td>
<td>0.3</td>
<td>10</td>
<td>5.6850E−4</td>
<td>0.0583</td>
<td>( (1.2518, 0.8055)^T )</td>
</tr>
<tr>
<td>( (20, 10)^T )</td>
<td>1.0</td>
<td>31</td>
<td>6.1847E−4</td>
<td>0.0583</td>
<td>( (1.2609, 0.7967)^T )</td>
</tr>
</tbody>
</table>
denotes the case that Algorithm 3.1, which coincides here with the Dinkelbach algorithm, was used. NIT denotes the number of iterations needed to reach the specified precision of $\Delta$. $f$ denotes the objective function value $p(x)/q(x)$, and the last column presents the approximate solution as the algorithms terminate.

We have made similar numerical tests for the following problem.

**Example 5.2.** Consider the fractional programming problem (3) in Example 2.4, where

$$p(x) = \max\{x_1^4 + x_2^2, (2 - x_1)^2 + (2 - x_2)^2, 2e^{-x_1 + x_2}\},$$

$$q(x) = (c_1 x_1 + c_2 x_2 + 1),$$

$$C = \{x \in \mathbb{R}^2; 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, x_1 + x_2 \leq 3\}.$$  

The results are displayed in Table 2.

We have also tested a set of randomly generated smooth quasiconvex fractional programming problems which were suggested in Barros (Ref. 13).

**Example 5.3.** See Ref. 13. Consider the fractional programming problem (3) in Example 2.4, where

$$p(x) = (1/2)x^THx + a^Tx + b,$$

$$q(x) = c^Tx + d.$$ 

The Hessian is defined by $H = LU^T$, where $L$ is a unit lower triangular matrix with components uniformly drawn from the interval $[-2.5, 2.5]$ and $U$ is a positive diagonal matrix with elements uniformly drawn from the

<table>
<thead>
<tr>
<th>$c$</th>
<th>$\lambda$</th>
<th>NIT</th>
<th>$\Delta$</th>
<th>$f$</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>Dk</td>
<td>38</td>
<td>9.3494E-4</td>
<td>2.0036</td>
<td>(0.9992, 0.9928)</td>
</tr>
<tr>
<td>(0, 0)</td>
<td>0.3</td>
<td>34</td>
<td>6.5269E-4</td>
<td>2.0016</td>
<td>(1.0006, 0.9997)</td>
</tr>
<tr>
<td>(0, 0)</td>
<td>1.0</td>
<td>20</td>
<td>9.8776E-4</td>
<td>2.0058</td>
<td>(1.0010, 1.0008)</td>
</tr>
<tr>
<td>(2, 1)</td>
<td>Dk</td>
<td>15</td>
<td>2.5446E-4</td>
<td>0.5002</td>
<td>(1.0048, 1.0015)</td>
</tr>
<tr>
<td>(2, 1)</td>
<td>0.3</td>
<td>15</td>
<td>9.7107E-4</td>
<td>0.5018</td>
<td>(0.9986, 1.0020)</td>
</tr>
<tr>
<td>(2, 1)</td>
<td>1.0</td>
<td>22</td>
<td>6.6151E-4</td>
<td>0.5003</td>
<td>(1.0001, 1.0005)</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>Dk</td>
<td>231</td>
<td>3.3851E-4</td>
<td>0.0653</td>
<td>(1.0011, 0.9977)</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>0.3</td>
<td>18</td>
<td>5.8651E-4</td>
<td>0.0645</td>
<td>(0.9999, 1.0003)</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>1.0</td>
<td>20</td>
<td>8.6725E-4</td>
<td>0.0645</td>
<td>(1.0001, 0.9998)</td>
</tr>
</tbody>
</table>
Table 3. Computational results.

<table>
<thead>
<tr>
<th>$n$</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_k$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$\lambda = 0.3$</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>$\lambda = 1.0$</td>
<td>11</td>
<td>7</td>
<td>4</td>
<td>14</td>
<td>17</td>
<td>16</td>
</tr>
</tbody>
</table>

interval $[0.1, 1.6]$. The elements of the vectors $a$ and $c$ are uniformly drawn from the interval $[-15.0, 45.0]$ and the interval $[0.0, 30.0]$, respectively. Similarly, $b$ and $d$ are drawn from the intervals $[-30.0, 0]$ and $[5.0, 35.0]$, respectively. Finally,

$$c = \left\{ x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1.0, \; 0 = x_i \right\}.$$  

The starting point is the origin. Numerical results are displayed in Table 3. Here, $n$ denotes the number of variables in this problem.

The initial numerical results show that, when applied to smooth quasiconvex fractional programming problems, the Dinkelbach algorithm, as an example of the level function method, takes less outer iterations to reduce $\Delta(i)$ to the required precision than other versions of the level function algorithms. In contrast, when employed to solve nonsmooth quasiconvex fractional programming problems, the linearized projected level function algorithm is more promising. Note also that, in Example 5.1 and Example 5.2, the Dinkelbach algorithm requires a nonsmooth convex minimization process at each outer iteration; this may be costly, and therefore Algorithms 3.2 and 3.3 may be preferable in this setting.

**Example 5.4.** Consider the 5-dimensional variational inequality problem (21), where the components of $F(x)$ are given as

$$f_1(x) = 0.726x_1 - 0.949x_2 + 0.266x_3 - 1.193x_4 - 0.504x_5 + \rho \arctan(x_1 - 2) + 5.308,$$

$$f_2(x) = 1.645x_1 + 0.678x_2 + 0.333x_3 - 0.217x_4 - 1.443x_5 + \rho \arctan(x_2 - 2) + 0.008,$$

$$f_3(x) = -1.016x_1 - 0.225x_2 + 0.769x_3 + 0.934x_4 + 1.007x_5 + \rho \arctan(x_3 - 2) - 0.938,$$

$$f_4(x) = 1.063x_1 + 0.567x_2 - 1.144x_4 + 0.550x_4 - 0.548x_5 + \rho \arctan(x_4 - 2) + 1.024,$$
\[ f_5(x) = -0.259x_1 + 1.453x_2 - 1.073x_3 + 0.509x_4 + 1.026x_5 + \rho \arctan(x_5 - 2) - 1.312. \]

We considered two constraint sets:

\[ C_1 = \left\{ x \in \mathbb{R}^5 : x_i \geq 0, i = 1, \ldots, 5, \sum_{i=1}^{5} x_i \leq 50 \right\} \]

and

\[ C_2 = \left\{ x \in \mathbb{R}^5 : x_i \geq 2, i = 1, \ldots, 5, \sum_{i=1}^{5} x_i \leq 50 \right\}. \]

When \( C = C_1 \), the approximate solution is \((1.770, 1.825, 1.820, 1.812, 1.820)^T\); and when \( C = C_2 \), the solution is \((2, 2, 2, 2, 2)^T\). This example is a modification of an example considered by Taji, Fukushima, and Ibaraki in Ref. 14. We tested Algorithm 4.1 for this example. The algorithm terminates when \( \Delta(i) = 10^{-4} \). The results are displayed in Table 4. In the table, \( C \) denotes that constraint set of (VIP). The last column presents the approximate solution at which the algorithm terminates at the prescribed precision of \( \Delta(i) \) and \( \|F(x_i)\| \) is the value of the 2-norm of \( F \) at the terminated approximate solution.

The initial results show that Algorithm 4.1 converged very slowly when the solution of (VIP) is at the interior of the constraint set, but it converged relatively quickly when the solution is at the boundary of the constraint set.

<table>
<thead>
<tr>
<th>( C )</th>
<th>Starting point</th>
<th>( \Delta )</th>
<th>NIT</th>
<th>( |F(x_i)| )</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 )</td>
<td>(0, 0, 0, 0, 0)( ^T )</td>
<td>8.8346E-5</td>
<td>151</td>
<td>0.0015</td>
<td>(1.7697, 1.8248, 1.8198, 1.8123, 1.8259)( ^T )</td>
</tr>
<tr>
<td>( C_1 )</td>
<td>(50, 0, 0, 0, 0)( ^T )</td>
<td>9.7800E-5</td>
<td>153</td>
<td>0.0012</td>
<td>(1.7697, 1.8248, 1.8196, 1.8124, 1.8258)( ^T )</td>
</tr>
<tr>
<td>( C_1 )</td>
<td>(0, 50, 0, 0, 0)( ^T )</td>
<td>9.9515E-5</td>
<td>183</td>
<td>0.0015</td>
<td>(1.7698, 1.8247, 1.8197, 1.8123, 1.8259)( ^T )</td>
</tr>
<tr>
<td>( C_1 )</td>
<td>(0, 0, 50, 0, 0)( ^T )</td>
<td>9.8964E-5</td>
<td>154</td>
<td>0.0018</td>
<td>(1.7699, 1.8249, 1.8197, 1.8123, 1.8259)( ^T )</td>
</tr>
<tr>
<td>( C_1 )</td>
<td>(0, 0, 0, 50, 0)( ^T )</td>
<td>9.1039E-5</td>
<td>144</td>
<td>0.0010</td>
<td>(1.7698, 1.8249, 1.8197, 1.8124, 1.8258)( ^T )</td>
</tr>
<tr>
<td>( C_1 )</td>
<td>(0, 0, 0, 0, 50)( ^T )</td>
<td>9.4649E-5</td>
<td>155</td>
<td>0.0020</td>
<td>(1.7698, 1.8247, 1.8196, 1.8123, 1.8259)( ^T )</td>
</tr>
<tr>
<td>( C_2 )</td>
<td>(42, 2, 2, 2, 2)( ^T )</td>
<td>3.3245E-16</td>
<td>4</td>
<td>4.4721</td>
<td>(2.0000, 2.0000, 2.0000, 2.0000, 2.0000)( ^T )</td>
</tr>
<tr>
<td>( C_2 )</td>
<td>(2, 42, 2, 2, 2)( ^T )</td>
<td>6.6529E-15</td>
<td>17</td>
<td>4.4721</td>
<td>(2.0000, 2.0000, 2.0000, 2.0000, 2.0000)( ^T )</td>
</tr>
<tr>
<td>( C_2 )</td>
<td>(2, 2, 42, 2, 2)( ^T )</td>
<td>1.5491E-14</td>
<td>13</td>
<td>4.4721</td>
<td>(2.0000, 2.0000, 2.0000, 2.0000, 2.0000)( ^T )</td>
</tr>
<tr>
<td>( C_2 )</td>
<td>(2, 2, 2, 42, 2)( ^T )</td>
<td>9.6048E-16</td>
<td>16</td>
<td>4.4721</td>
<td>(2.0000, 2.0000, 2.0000, 2.0000, 2.0000)( ^T )</td>
</tr>
<tr>
<td>( C_2 )</td>
<td>(2, 2, 2, 2, 42)( ^T )</td>
<td>9.6048E-16</td>
<td>7</td>
<td>4.4721</td>
<td>(2.0000, 2.0000, 2.0000, 2.0000, 2.0000)( ^T )</td>
</tr>
</tbody>
</table>
References