Journal of Computational Mathematics, Vol.21, No.4, 2003, 451–462.

PROXIMAL POINT ALGORITHM FOR MINIMIZATION OF DC FUNCTION *1)

Wen-yu Sun
(*School of Mathematics and Computer Science, Nanjing Normal University, Nanjing 210097, China*)

Raimundo.J.B. Sampaio
(*Programa de Pos-Graduacao em Informatica Aplicada, Pontificia Universidade Catolica do Parana (PUCPR), CEP: 80215-901, Curitiba, PR, Brazil*)

M.A.B. Candido
(*Programa de Pos-Graduacao em Informatica Aplicada, Pontificia Universidade Catolica do Parana (PUCPR), CEP: 80215-901, Curitiba, PR, Brazil*)

Abstract

In this paper we present some algorithms for minimization of DC function (difference of two convex functions). They are descent methods of the proximal-type which use the convex properties of the two convex functions separately. We also consider an approximate proximal point algorithm. Some properties of the ϵ-subdifferential and the ϵ-directional derivative are discussed. The convergence properties of the algorithms are established in both exact and approximate forms. Finally, we give some applications to the concave programming and maximum eigenvalue problems.

Key words: Nonconvex optimization, Nonsmooth optimization, DC function, Proximal point algorithm, ϵ-subgradient.

1. Introduction

In this paper we consider solving a special class of nonconvex optimization problems:

$$\min_{x\in R^n} \quad f(x), \tag{1.1}$$

where $f : R^n \to R$ is a nonconvex function. In many cases, for example, in optimal control and engineering design, the nonconvex function f can be dealt with as a difference of two convex functions

$$f(x) = g(x) - h(x), \quad \forall x \in R^n, \tag{1.2}$$

where $g : R^n \to R$ and $h : R^n \to R$ are proper, convex, and lower semi-continuous (l.s.c.). In this case, the function f is called DC function.

The interest for studying DC function (i.e. difference of two convex functions) is motivated by the possibility of using twice the underlying convex structure of such representation when dealing with nonconvex problems. This is especially attractive when one of these convex functions or both is nonsmooth. Although there is a lot of papers devoted to the theory of DC functions in the literature (see for example, [6] [7] [8]), only a few have proposed some specific algorithms and reported some numerical experiments. Here we quote some methods which use the regularization approach [3] [19], the dual approach [1] and the subgradient method [13], respectively.

* Received February 13, 2001; final revised July 16, 2001.

1) This work was supported by the National Natural Science Foundation of China, the Oversea Exchange Fund of Nanjing Normal University, and CNPq of Brazil.

It is well-known that proximal point algorithm (PPA) is an effective method for solving nonsmooth convex optimization problems. Its remarkable feature is that a nonsmooth convex optimization problem can be converted to a continuously differentiable convex optimization problem. Consequently, we can use some methods for smooth optimization to deal with it. This paper aims to study using proximal point algorithm to minimize a DC function.

Let $\langle \cdot, \cdot \rangle$ denote the inner product in R^n, Γ_0 the set of convex proper and l.s.c functions on R^n. Let $f : R^n \to R$ be a DC function on R^n, i.e. there exist g and h both in Γ_0 such that

$$f(x) = g(x) - h(x),\ \forall x \in R^n. \tag{1.3}$$

Moreover, suppose that $\mathrm{Dom}(g) \cap \mathrm{Dom}(h) \neq \phi$, where $\mathrm{Dom}(g)$ denotes the domain of g

$$\mathrm{Dom}(g) := \{x \in R^n \mid g(x) < \infty\}.$$

The functions g and h can be chosen as strongly convex since one can always add a strongly convex function to each function, for example,

$$f(x) = [g(x) + \omega(x)] - [h(x) + \omega(x)],$$

where $\omega : R^n \to R$ is a strongly convex function. The corresponding conjugate function of g and h are denoted by g^* and h^*, and their respective subdifferentials by ∂g, $\partial h, \partial g^*$ and ∂h^*.

Proposition 1.1. (see [23] [8])

1.

$$\inf_{x \in R^n} \{g(x) - h(x)\} = \inf_{y \in R^n} \{h^*(y) - g^*(y)\}. \tag{1.4}$$

2. A necessary condition for $x \in Dom(f)$ to be a local minimizer of f is

$$\partial h(x) \subset \partial g(x). \tag{1.5}$$

In general, the condition 2 above is hard to be reached and one may relax it to

$$\partial g(x) \cap \partial h(x) \neq \phi. \tag{1.6}$$

We say that x^* is a critical point of f if it satisfies (1.6).

The method presented in this paper is closely related to the proximal point algorithm (see e.g. [17]). This class of algorithms finds a zero of a maximal monotone operator T by means of the following iteration:

$$x_{k+1} = (I + c_k T)^{-1} x_k, \tag{1.7}$$

where $c_k > c > 0$, where c is a suitably small positive number such that $I + cT$ is nonsingular. The operator $P_k = (I + c_k T)^{-1}$ which is the resolvent of T, is nonexpansive, single-valued on the whole space, and Lipschitz continuous. When T is the subdifferential of a convex l.s.c. function g, i.e., $T = \partial g$, the iteration (1.7) becomes

$$x_{k+1} = \arg\min\{g(x) + \frac{1}{2c_k} \parallel x - x_k \parallel^2\}. \tag{1.8}$$

Rockafellar [17] [18] has developed a detailed study of the convergence on proximal point algorithm. In particular, the algorithm converges linearly at least. If $c_k \to \infty$, the convergence is superlinear. In addition, the attractive approximate versions of proximal point algorithm are established by [17] [10].

With this strategy, we propose a new descent algorithm for finding a critical point of a DC function which satisfies necessary optimality conditions. Each iteration combines an ascent subgradient step on the second function with a proximal step on the first function. In addition, the approximate version of our algorithm is also discussed.

The organization of this paper is as follows. In Section 2 we describe our method for minimizing DC function which combines proximal point algorithm with subgradient method. In Section 3 we establish the convergence properties of our algorithm. In Section 4 we study some further properties of ϵ-subdifferential and ϵ-directional derivative. In Section 5, by means of the above theory, we present an approximate version of proximal point algorithm for solving DC optimization which is interesting and important from the practical point of view. Finally, in Section 6, we discuss some applications to concave programming and maximum eigenvalue problems.

2. A Descent Method for Minimization of DC Function

In this section we demonstrate our method via establishing two lemmas, describing the algorithm and giving several remarks. The key of this method is that each iteration will be decomposed into two distinct steps and each one takes into account the convexity of the function. Roughly speaking, this method consists of increasing the function h along the direction of the subgradient and then decreasing the function g by a proximal step. Indeed, any subgradient direction of a convex function is an ascent direction of the function.

Lemma 2.1. *Let $h \in \Gamma_0$ and $x \in R^n$. Then $\forall w \in \partial h(x)$ with $w \neq 0$ and $\forall c_k > c > 0$, we have $h(x + c_k w) > h(x)$.*

Proof. It is an immediate consequence of the subgradient inequality:

$$h(x + c_k w) \geq h(x) + \langle w, c_k w \rangle, \ \forall w \in \partial h(x).$$

The following lemma gives a necessary and sufficient condition for x to be a critical point of DC function f. It is not difficult to see that any critical point of DC function f can be viewed as a fixed point of a certain operator.

Lemma 2.2. *A necessary and sufficient condition for x to be a critical point of f is that*

$$x = (I + c_k \partial g)^{-1}(x + c_k w) \tag{2.1}$$

for any $c_k > c > 0$ and $w \in \partial h(x)$.

Proof. Let x be a critical point of f. From (1.6), there exists $w \neq 0$ such that $w \in \partial g(x) \cap \partial h(x)$. Obviously, $w \in \partial g(x)$, which is equivalent to $x + c_k w \in x + c_k \partial g(x)$. Since ∂g is a maximal monotone operator and $(I + c_k \partial g)^{-1}$ is single-valued, we get (2.1). Vice versa.

Set

$$P_k = (I + c_k \partial g)^{-1}.$$

The above Lemma 2.2 implies that $x = P_k(x + c_k w)$ if and only if x is a critical point of f. Note that P_k is a proximal mapping with $T = \partial g$, therefore one immediately gets a proximal point type iteration:

$$x_{k+1} = P_k(x_k + c_k w_k), \text{ where } P_k = (I + c_k \partial g)^{-1}. \tag{2.2}$$

In the following, we give our proximal point algorithm for minimization of dc functions.

Algorithm 2.3. *(Proximal Point Algorithm)*

Step 1. Given an initial point x_0 and $c_0 > c > 0$. Set $k = 0$.

Step 2. Compute $w_k \in \partial h(x_k)$ and set $y_k = x_k + c_k w_k$.

Step 3. Compute $x_{k+1} = (I + c_k \partial g)^{-1}(y_k)$ by proximal point algorithm.

Step 4. If $x_{k+1} = x_k$, stop. Otherwise $k := k + 1$ and return to Step 2.

Remarks.

1. In the convex case ($h \equiv 0$), it is just the proximal point algorithm (see [17]).

2. When h is differentiable, this algorithm is closely related to one proposed by Mine and Fukushima [11].

3. It is well-known from Lemma 2.2 that the method is a regularization approach. More precisely, it is clear that the problem of minimizing $f(x) = g(x) - h(x)$ is equivalent to minimizing $\hat{f}(x) = \hat{g}(x) - \hat{h}(x)$ with $\hat{g}(x) = cg + \frac{1}{2} \parallel \cdot \parallel^2$ and $\hat{h}(x) = ch + \frac{1}{2} \parallel \cdot \parallel^2$ for $c > 0$. The method can then be written as
$$y_k \in \partial\hat{h}(x_k) \text{ and } x_{k+1} \in \partial(\hat{g})^*(y_k), \tag{2.3}$$
where $(\hat{g})^*$ is the conjugate of $\hat{g}$ (indeed, it is easy to show that $\partial(\hat{g})^* = (I + c\partial g)^{-1}$ from the definition of conjugate function). The alternative steps with the subgradients of $\hat{h}$ and $(\hat{g})^*$ showed the relation with the subgradient methods introduced by Pham and Souad [13]. By the way, it is important to observe that any algorithm designed for dc functions depends on the decomposition of dc functions where the decomposition is not unique.

4. Just like any proximal type method, its numerical behaviour depends on the relative complexity of the proximal point computation. Here, Step 3 means that we must solve
$$\min_x \{g(x) + \frac{1}{2c_k} \parallel x - y_k \parallel^2\},$$
which can be solved, for example, by a cutting plane algorithm or bundle method (see [4] [5] [9] [20] [24]). Here c_k is changed in each iteration. About the choice of this important parameter, we'll discuss some concrete schemes in a separate paper.

3. Convergence of the Algorithm

In this section we shall establish the convergence of the algorithm. We begin by showing that Algorithm 2.3 is a descent algorithm.

Theorem 3.1. *The sequence $\{x_k\}$ generated by Algorithm 2.3 satisfies*

- *either the algorithm stops at a critical point of f;*
- *or f decreases strictly, that is, $f(x_{k+1}) < f(x_k)$.*

Proof. If $x_{k+1} = x_k$, then, from Lemma 2.2, x_k is a critical point of f. Suppose that $x_{k+1} \neq x_k$. Using the subgradient inequality, we can rewrite the iteration (2.2) in the following way:
$$\begin{aligned} & x_k + c_k w_k \in x_{k+1} + c_k \partial g(x_{k+1}) \\ \Longleftrightarrow\quad & c_k^{-1}(x_k - x_{k+1}) + w_k \in \partial g(x_{k+1}) \\ \Longleftrightarrow\quad & g(x_k) \geq g(x_{k+1}) + \langle c_k^{-1}(x_k - x_{k+1}) + w_k, x_k - x_{k+1}\rangle. \end{aligned} \tag{3.1}$$
On the other hand, w_k is a subgradient of h at x_k, then we have
$$h(x_{k+1}) \geq h(x_k) + \langle w_k, x_{k+1} - x_k\rangle. \tag{3.2}$$
If we subtract (3.2) from (3.1), we obtain
$$f(x_{k+1}) \leq f(x_k) - c_k^{-1} \parallel x_k - x_{k+1} \parallel^2 . \tag{3.3}$$
Hence we conclude that $f(x_{k+1}) < f(x_k)$.

In the following, we use Zangwill's convergence theorem [25] to prove the convergence of our algorithm.

Theorem 3.2. *Assume that the sequence* $\{x_k\}$ *and* $\{y_k\}$ *generated by Algorithm 2.3 are bounded. Then, any convergent subsequence of* $\{x_k\}$ *converges to a critical point of* f.

Proof. According to Zangwill [25], the global convergence of an algorithm depends on three properties of iterative sequence: descent, closedness and boundedness. Now we enter the details to these three properties. Let S be the set of critical points of f.

1. f is a descent function out of S. Indeed, Theorem 3.1 guarantees that $f(x_{k+1}) < f(x_k)$, $\forall x$ *such that* $x_k \neq x_{k+1}$. Hence, from Lemma 2.2, $x_k \notin S$. Obviously, if $x_k \in S$, then $f(x_k) = f(x_{k+1})$.
2. The algorithm map is closed. In fact, the algorithm can be written as $x_{k+1} \in B \circ C(x_k)$ with $B = (I + c_k \partial g)^{-1}$ and $C = I + c_k \partial h$. Note that B is the resolvent operator of ∂g, hence its graph is closed. Moreover, since h is a proper convex l.s.c. function, the graph of C is also closed. Therefore, the sequence $\{y_k\}$ with $y_k \in C(x_k)$ being bounded by the hypothesis, possesses a convergent subsequence $\{y'_k\}$, and then the map $B \circ C$ is closed (see the Theorem on the composition of closed point–to–set maps in Zangwill [25]).
3. The sequence $\{x_k\}$ is bounded by the assumption.

Therefore, using Zangwill's theorem [25], any convergent subsequence of $\{x_k\}$ converges to a critical point of f in S.

4. The ϵ-Directional Derivative and the ϵ-Subdifferential

In the above, we have discussed proximal point algorithm in exact form. But from a practical point of view it is more important to replace the exact form of PPA by an approximate version which is based on the theory of the ϵ-subgradient of convex function. In fact, relating to the exact subgradient, the ϵ-subgradient is relaxed by ϵ. This relaxedness brings us importance on the computational side and the theoretical side (for example see [9] [10]). In this section, we plan to introduce and explore some basic properties of ϵ-directional derivative and ϵ-subdifferential of convex function. Please note that, in order to agree with custom, in this section the function f does not stand for DC function, but for a convex function.

For $x \in \text{dom } f$ and $\epsilon \geq 0$,

$$f'_\epsilon(x; d) = \inf_{t>0} \frac{f(x+td) - f(x) + \epsilon}{t} \tag{4.1}$$

which is called the ϵ-directional derivative of f at x. If

$$f(y) \geq f(x) + \langle s, y - x \rangle - \epsilon, \ \forall y \in R^n, \tag{4.2}$$

the vector $s \in R^n$ is called an ϵ-subgradient of f at x. The set of all ϵ-subgradients of f at x is called the ϵ-subdifferential of f at x, denoted by $\partial_\epsilon f(x)$ which is a nonempty, convex, bounded and closed set. In addition, $s \in R^n$ is an ϵ-subgradient of f at x if and only if

$$f^*(s) + f(x) - \langle s, x \rangle \leq \epsilon, \tag{4.3}$$

where f^* is the conjugate of the function f. From the above definitions, we have

$$f'_\epsilon(x; d) = \sup_{s \in \partial_\epsilon f(x)} \langle s, d \rangle \tag{4.4}$$

which also means that

$$\partial_\epsilon f(x) = \{ s \in R^n \mid \langle s, d \rangle \leq f'_\epsilon(x; d), \ \forall d \in R^n \}.$$

It is easy to see that for convex function $f(x)$ the following relation

$$0 \in \partial_\epsilon f(x) \ \Leftrightarrow \ f(x) \leq f(y) + \epsilon, \forall y \in R^n$$

is true.

In the following, we will explore some properties of ϵ-directional derivative and ϵ-subdifferential. These properties are generalization of the corresponding properties of directional derivative and subdifferential of convex function.

(1) The conjugate function of ϵ-directional derivative.

Since

$$f'_\epsilon(x;d) = \inf_{t>0} \frac{f(x+td)-f(x)+\epsilon}{t},$$

then setting

$$f_t(d) = \frac{f(x+td)-f(x)+\epsilon}{t}$$

and using the operation rules (ii), (iii) and (v) of conjugate function in Proposition 1.3.1 of [9], we get

$$f_t^*(s) = \frac{f^*(s)+f(x)-\langle s,x\rangle-\epsilon}{t}.$$

Since $f'_\epsilon(x;\cdot)$ is an inf-function, its conjugate function is a sup-function. Consequently,

$$\begin{aligned}
[f'_\epsilon(x;\cdot)]^*(s) &= \sup_{t>0} f_t^*(s) \\
&= \sup_{t>0} \frac{f^*(s)+f(x)-\langle s,x\rangle-\epsilon}{t}.
\end{aligned} \tag{4.5}$$

Note that the supremum in (4.5) is always nonnegative and that it is zero if and only if $s \in \partial_\epsilon f(x)$.

(2) $\partial_\epsilon f(x) \subset \partial f(B(x,\delta))$.

As to this property, Theorem XI 4.2.1 in [9] indicates that: for any $\eta > 0$ and $s \in \partial_\epsilon f(x)$, there exist $x_\eta \in B(x,\eta)$ and $s_\eta \in \partial f(x_\eta)$ such that $\|s_\eta - s\| \le \epsilon/\eta$. Our result is similar, but the proof is different, more simple and intuitive.

Theorem 4.1. *For all $\delta > 0$ there exists $\epsilon \ge 0$ such that*

$$\partial_\epsilon f(x) \subset \partial f(B(x;\delta)). \tag{4.6}$$

Proof. For any $s \in \partial_\epsilon f(x)$, let $\bar{s} = P_{\partial f(x)}(s)$ be an orthogonal projection of s onto $\partial f(x)$. Note that $\partial f(x_\delta) \in \partial f(B(x;\delta)), \partial f(x) \in \partial f(B(x;\delta))$ and $\partial f(B(x;\delta))$ is a nonempty compact convex set. Then, to prove (4.6), it is enough to prove that there is $\sigma \ge 0$ such that

$$\operatorname{dist}(s, \partial f(x)) \le \sigma, \tag{4.7}$$

i.e.,

$$\|s-\bar{s}\| \le \sigma. \tag{4.8}$$

If $s = \bar{s}$, the result is trivial. Now we consider the case of $s \ne \bar{s}$.

Take $\epsilon \le \frac{1}{2}\sigma\delta$. Let $s = \bar{s} + \frac{\|s-\bar{s}\|}{\delta} d, d \in R^n$. Also let $x_\delta \in B(x,\delta)$, $x_\delta = x+d$ and $s_\delta \in \partial f(x_\delta)$. For $s \in \partial_\epsilon f(x)$, we have

$$f(x_\delta) - f(x) - \langle s, x_\delta - x\rangle \ge -\epsilon,\ \forall x_\delta \in B(x,\delta). \tag{4.9}$$

For $s_\delta \in \partial f(x_\delta)$, we have

$$f(x) - f(x_\delta) - \langle s_\delta, x - x_\delta\rangle \ge 0,\ \forall x \in \operatorname{dom} f. \tag{4.10}$$

It follows from (4.9) and (4.10) that

$$\langle s - s_\delta, x - x_\delta\rangle \ge -\epsilon. \tag{4.11}$$

From the outer semi-continuity of ∂f, for any $s_\delta \in \partial f(x_\delta)$, there exists $\sigma > 0$ such that

$$\|s_\delta - \tilde{s}\| \leq \frac{\sigma}{2}, \tag{4.12}$$

where $\tilde{s} \in \partial f(x)$.

Obviously, the left hand side of (4.11) can be written as follows:

$$\langle s - s_\delta, x - x_\delta \rangle = \langle s - \bar{s}, x - x_\delta \rangle + \langle \bar{s} - \tilde{s}, x - x_\delta \rangle + \langle \tilde{s} - s_\delta, x - x_\delta \rangle. \tag{4.13}$$

Using (4.12), the third term of right hand side in (4.13) becomes

$$\langle \tilde{s} - s_\delta, x - x_\delta \rangle \leq \frac{\sigma\delta}{2}. \tag{4.14}$$

Also, note that

$$x - x_\delta = -d = \delta \frac{\bar{s} - s}{\|s - \bar{s}\|}$$

and that $\bar{s}$ is the projection of s on $\partial f(x)$ which means that

$$\langle s - \bar{s}, \tilde{s} - \bar{s} \rangle \leq 0, \ \tilde{s} \in \partial f(x)$$

from the projection theorem (see [24] Th.1.3.17). Therefore, the second term of right hand side in (4.13) becomes

$$\langle \bar{s} - \tilde{s}, x - x_\delta \rangle = \langle \bar{s} - \tilde{s}, \delta \frac{\bar{s} - s}{\|s - \bar{s}\|} \rangle \leq 0, \forall \tilde{s} \in \partial f(x). \tag{4.15}$$

In addition, for the first term of right hand side in (4.13), we have obviously from $x_\delta - x = d = \delta \frac{s - \bar{s}}{\|s - \bar{s}\|}$ that

$$\langle s - \bar{s}, x - x_\delta \rangle = -\delta \|s - \bar{s}\|. \tag{4.16}$$

Combining (4.11)–(4.16) gives

$$-\delta\|s - \bar{s}\| + \frac{\sigma\delta}{2} \geq -\epsilon.$$

Noting that $\epsilon \leq \frac{1}{2}\sigma\delta$, hence

$$\|s - \bar{s}\| \leq \frac{\epsilon}{\delta} + \frac{\sigma}{2} \leq \sigma.$$

We complete the proof.

At the end of this section, we want to point out that all these theories on ϵ–subdifferential can go a step further if instead of (4.2) we consider a function that satisfies the rule: $\forall y \in R^n, \forall s \in \partial_\epsilon f(x)$,

$$f(y) \geq f(x) + s^T \phi(x, y) - \epsilon,$$

where $\phi(x, y)$ is a function with some properties. It is immediate to see that when $\phi(x, y) = y - x$ we get (4.2). So, this class of functions is a generalization of ϵ-subdifferentiable function. For the sake of limit of space, we won't enter this topic.

5. Approximate Version of Proximal Point Algorithm

The research on ϵ-subgradient method is an attractive direction in nonsmooth optimization. With applications of ϵ-subgradient, ϵ-bundle method, ϵ-feasible direction method and other proximal nonsmooth method are successively presented. In this section, by means of ϵ-subgradient, we establish an approximate version of proximal point algorithm for minimization of DC function and analyze its convergence property. Our analysis resembles the exact case in Section 2 and Section 3, but with the obvious distinction of dealing with ϵ-subgradient. It means, the method consists of increasing the function h in the direction of ϵ-subgradient of h, and then decreasing the function g by an ϵ-proximal step.

We first give a definition of ϵ-critical point of DC function f. (Note that in this section the function f denotes DC function as in Section 1-3).

Let $\epsilon \geq 0$, the point x is said to be an ϵ-critical point of f if it satisfies

$$\partial_\epsilon g(x) \cap \partial_\epsilon h(x) \neq \phi. \tag{5.1}$$

Obviously, when $\epsilon = 0$, (5.1) reduces to (1.6).

Lemma 5.1. *Let $h \in \Gamma_0$ and $x \in R^n$. Then $\forall w \in \partial_\epsilon h(x)$ with $w \neq 0$ and $\forall c_k > c > 0$, we have*

$$h(x + c_k w) > h(x) - \epsilon.$$

Proof. It follows directly from the ϵ-subgradient inequality:

$$h(x + c_k w) \geq h(x) + \langle w, c_k w\rangle - \epsilon, \ \forall w \in \partial_\epsilon h(x).$$

The following result gives a necessary and sufficient condition for x to be an ϵ-critical point of f.

Lemma 5.2. *A necessary and sufficient condition for x to be an ϵ-critical point of f is that*

$$x \in (I + c_k \partial_\epsilon g)^{-1}(x + c_k w) \tag{5.2}$$

for any $c_k > c > 0$ and $w \in \partial_\epsilon h(x)$.

Proof. Let x be an ϵ-critical point of f. From (5.1), there is $w \neq 0$ such that $w \in \partial_\epsilon g(x) \cap \partial_\epsilon h(x)$. Then $w \in \partial_\epsilon g(x)$ which is equivalent to

$$x + c_k w \in x + c_k \partial_\epsilon g(x).$$

Therefore

$$x \in (I + c_k \partial_\epsilon g)^{-1}(x + c_k w).$$

Vice versa.

From the above lemma we obtain an ϵ-proximal point type iteration:

$$x_{k+1} \in P_{k,\epsilon}(x_k + c_k w_k), \text{ where } P_{k,\epsilon} = (I + c_k \partial_\epsilon g)^{-1}. \tag{5.3}$$

Now we summarize these ideas and state the following modification of Algorithm 2.3.

Algorithm 5.3. *(ϵ-Proximal Point Algorithm)*

Step 1. *Given an initial point x_0 and $c_0 > c > 0$. Set $k = 0$.*

Step 2. *Compute $w_k \in \partial_\epsilon h(x_k)$ and set $y_k = x_k + c_k w_k$.*

Step 3. *Compute $x_{k+1} = (I + c_k \partial_\epsilon g)^{-1}(y_k)$ from an approximate proximal point algorithm.*

Step 4. *If $x_{k+1} = x_k$, stop. Otherwise $k := k + 1$ and return to Step 2.*

In fact, in Algorithm 5.3, finding the ϵ-subgradient of h is an inner iteration; the ϵ-proximal step for g is an outer iteration. The following theorem indicates that the ϵ-proximal point algorithm provides a 2ϵ-descent of the objective function for each iteration.

Theorem 5.4. *If x_k is not an ϵ-critical point of f, then $\{f(x_k)\}$ is 2ϵ-descent, i.e., $\{f(x_k)\}$ satisfies*

$$f(x_{k+1}) < f(x_k) + 2\epsilon. \tag{5.4}$$

Proof. From ϵ-PPA iteration (5.3), we have

$$c_k^{-1}(x_k - x_{k+1}) + w_k \in \partial_\epsilon g(x_{k+1})$$

which implies

$$g(x_k) \geq g(x_{k+1}) + \langle c_k^{-1}(x_k - x_{k+1}) + w_k, x_k - x_{k+1}\rangle - \epsilon. \tag{5.5}$$

Also, since $w_k \in \partial_\epsilon h(x_k)$, we have

$$h(x_{k+1}) \geq h(x_k) + \langle w_k, x_{k+1} - x_k\rangle - \epsilon. \tag{5.6}$$

Subtracting (5.6) from (5.5) gives

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - c_k^{-1}\|x_k - x_{k+1}\|^2 + 2\epsilon \\ &< f(x_k) + 2\epsilon. \end{aligned}$$

Note that the exact PPA method in our present situation is

$$x_{k+1} = P_k(x_k + c_k w_k),\ P_k = (I + c_k \partial g)^{-1} \tag{5.7}$$

which is equivalent to

$$0 \in \partial g(x_{k+1}) + \frac{1}{c_k}(x_{k+1} - y_k), \tag{5.8}$$

where $y_k = x_k + c_k w_k$. This implies that x_{k+1} is the solution of the subproblem

$$\min\{g(z) + \frac{1}{2c_k}\|z - y_k\|^2\}. \tag{5.9}$$

For ϵ-PPA method,

$$x_{k+1} = P_{k,\epsilon}(x_k + c_k w_k),\ P_{k,\epsilon} = (I + c_k \partial_\epsilon g)^{-1} \tag{5.10}$$

which is equivalent to finding x_{k+1} such that

$$x_{k+1} \approx \arg\min\{g(z) + \frac{1}{2c_k}\|z - y_k\|^2\}. \tag{5.11}$$

In fact, let

$$\tilde{g}(y_k) := \min\{g(z) + \frac{1}{2c_k}\|z - y_k\|^2\}. \tag{5.12}$$

This is the Moreau-Yosida regularization and the inf-convolution of g with the quadratic function $z \to \frac{1}{2c_k}\|z - y_k\|^2$. Taking ϵ-subgradient and setting $0 \in \partial_\epsilon \tilde{g}(y_k)$ yield

$$0 \in \partial_{\epsilon_1} g(x_{k+1}) \cap B\left(\frac{1}{c_k}(y_k - x_{k+1}), \sqrt{2\epsilon_2/c_k}\right), \tag{5.13}$$

where $B(\cdot,\cdot)$ is a closed ball, $\epsilon_1, \epsilon_2 \geq 0$ and $\epsilon_1 + \epsilon_2 = \epsilon$. In particular, setting $\epsilon_1 = \epsilon$ and $\epsilon_2 = 0$, we obtain

$$0 \in \partial_\epsilon g(x_{k+1}) + \frac{1}{c_k}(x_{k+1} - y_k) \tag{5.14}$$

which is just (5.10).

On the other hand, from (5.14) we have

$$\frac{1}{c_k}(y_k - x_{k+1}) \in \partial_\epsilon g(x_{k+1}).$$

Using Theorem 4.1, we deduce that

$$\begin{aligned} x_{k+1} &\in y_k - c_k \partial_\epsilon g(x_{k+1}) \\ &\subset y_k - c_k \partial g(B(x_{k+1}, \delta_k)) \end{aligned} \tag{5.15}$$

for some $\delta_k > 0$. The above expression indicates that, in the approximate version, $\partial_\epsilon g(x_{k+1})$ is an enlargement of $\partial g(x_{k+1})$, and therefore the value scope of x_{k+1} is also enlarged. This shows that x_{k+1} lies in an enlarged σ-neighborhood of the exact candidate and satisfies (5.11). Since, from Theorem 4.1, the enlargement is bounded by σ_k in each iteration, then if $\sum_{k=0}^{\infty} \sigma_k < \infty$, the Rockafellar's criteria (A) holds (see [17]). Thus we can use Theorem 1 in [17] to obtain the convergence: the sequence $\{x_k\}$ generated from Algorithm 5.3 converges to x^∞ satisfying

$$0 = w^\infty \in \partial g(x^\infty) \subset \partial_\epsilon g(x^\infty)$$

and

$$0 = w^\infty \in \partial_\epsilon h(x^\infty),$$

where $w^\infty \leftarrow w_k$. So, we have

$$0 \in \partial_\epsilon g(x^\infty) \cap \partial_\epsilon h(x^\infty).$$

To sum up, the ϵ-subgradient method is used in inner iteration, and in outer iteration just an approximate version of PPA is used which satisfies the conditions of Theorem 1 of [17].

According to the above derivation, we now state the convergence theorem.

Theorem 5.5. *Let $\{x_k\}$ be any sequence generated by ϵ-proximal point algorithm with $c_k > c > 0, \forall k$. Assume that $\{x_k\}$ is bounded. Then $\{x_k\}$ converges weakly to x^∞ satisfying $0 \in \partial_\epsilon g(x^\infty) \cap \partial_\epsilon h(x^\infty)$, i.e., $\{x_k\}$ converges weakly to an ϵ-critical point of f.*

6. Some Applications

6.1. Application to Concave Programming

Let's consider the application of Algorithm 2.3 to the following problem

$$\max_{x \in C} h(x), \tag{6.1}$$

where h is a convex l.s.c. function on a closed convex set C on R^n.

The nonconvex programming (6.1) has received a great attention during the last two decades, from the viewpoint of global optimization or even aiming at some efficient local search of critical points ([13] etc.). We do not intend here to analyze the numerical performance of Algorithm 2.3 in this context but rather to illustrate its bahaviour.

The problem (6.1) can be written easily as a DC programming problem by introducing an indicator function δ_C of the set C:

$$\min_{x \in R^n} \{\delta_C(x) - h(x)\}. \tag{6.2}$$

Algorithm 2.3 now takes the following form:

step 1. Initial step: $x_0 \in C,\ c_0 > c > 0, k := 0.$

step 2. k-th step: $x_{k+1} = Proj_C(x_k + c_k w_k),\ where\ w_k \in \partial h(x_k).$

The algorithm progress looks like the projected gradient algorithm.

6.2 Application to Maximum Eigenvalue

Let Q be a symmetric positive definite matrix. Each eigenvalue of Q is a positive real number satisfying the following condition:

$$\exists x \in R^n,\ x \neq 0,\ and\ a\ positive\ real\ number\ \lambda,\ such\ that\ Qx = \lambda x.$$

Then we have

$$\langle x, \lambda x\rangle = \langle x, Qx\rangle \iff \lambda = \frac{x^T}{\| x \|}\ Q\ \frac{x}{\| x \|},$$

and so, the maximum eigenvalue of Q can be reached by solving the problem

$$\max_{\|x\|\leq 1} \langle x, Qx\rangle.$$

Using the same reason and strategy we did in Subsection 6.1, we get how Algorithm 2.3 applies and

$$x_{k+1} = \frac{(I+cQ)x_k}{\| (I+cQ)x_k \|},$$

which points out that in this case Algorithm 2.3 behaves like the power iterative method for determining the maximum eigenvalue of the matrix Q, where c is a suitably small positive number.

7. Conclusion

This paper is aimed at proposing the proximal point algorithm for minimizing DC function in both exact and approximate forms, and making some theoretical investigation of the algorithms. In this paper we don't discuss the actual computation pattern. Our algorithms above are general and do not rely on the concrete computational scheme for proximal point. In general, to compute a proximal point, one can use cutting plane strategy, bundle method and quasi-Newton method. In any way, this kind of methods, PPA, will be more advantageous and more robust than existing algorithms (e.g. [1] [13] etc.). Especially, if one uses the quasi-Newton-PPA method, the superlinear convergence will be obtained under mild conditions. We will continue this topic in this line, including numerical experiments.

Acknowledgements. Many thanks to two anonymous referees for their various valuable comments and suggestions. Their suggestions make clear the proofs of related theorems and greatly improve the presentation of this paper.

References

[1] G. Auchmuty, Duality algorithm for nonconvex variational principle, Research Report UH/MD-41, University of Houston, (1988).

[2] F.H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, (1983).

[3] E. Fernandez-Cara and C. Moreno, Critical points approximation through exact regularization, *Mathematics of Computation*, **50** (1988) 139–153.

[4] M. Fukushima, A descent algorithm for nonsmooth convex optimization, *Mathematical Programming,* **30** (1984), 163-175.

[5] M. Fukushima and L. Qi, A globally and superlinearly convergent algorithm for nonsmooth convex minimization, *SIAM J. Optimization,* **6** (1996), 1106-1120.

[6] J.B. Hiriart-Urruty, From convex optimization to nonconvex optimization. necessary and sufficient conditions for global optimization, *Nonsmooth Optimization and Related Topics*, Plenum Press, (1989), 219-240.

[7] J.B. Hiriart-Urruty, Generalized differenciability, duality and optimization for problems dealing with difference of convex functions, *Convexity and Duality in Optimization*, Lecture Notes in Economics And Mathematical Systems 256, (1986), 37-70.

[8] J.B. Hiriart-Urruty and H. Tuy, Essays on nonconvex optimization, *Mthematical Programming,* **41** (1988).

[9] J.B. Hiriart-Urruty and C. Lemarechal, Convex Analysis and Minimization Algorithms II, Sringer-Verlag, Berlin, (1993).

[10] K.C. Kiwiel, Proximity control in bundle methods for convex non-differentiable minimization, *Mathematical Programming,* **46** (1990), 105-122.

[11] H. Mine and M. Fukushima, A minimization method for the sum of a convex function and a continuously differentiable function, *J.Optimization Theory and Applications,* **33** (1981), 9-23.

[12] J.S. Pang and L. Qi, Nonsmooth equations: motivation and algorithms, *SIAM J. Optimization,* **3** (1993), 443-465.

[13] D.T. Pham and E.B. Souad, Algorithms for solving a class of nonconvex optimizations problems: methods of subgradient, *Fermat Days 85: Mathematics for Optimization*, North Holland, (1986).

[14] L. Qi, Convergence analysis of some algorithms for solving nonsmooth equations, *Mathematics of Operations Research,* **18** (1993), 227-244.

[15] L. Qi and W. Sun, An iterative method for minimax problem, in D.Z.Du and P.M.Pardalos eds., *Minimax and Applications*, (Kluwer Academic Publisher, New York, 1995), 55-67.

[16] R.T. Rockafellar, Convex Analysis, Princeton University Press, (1970).

[17] R.T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM J. Control and Optimization*, **14** (1976), 877-898.

[18] R.T. Rockafellar and R.J.-B. Wets, Variational Analysis, Springer-Verlag, New York, (1997).

[19] R.J.B. Sampaio, Contribution to study of dc programming, Doctoral Thesis, Catholic University of Rio de Janeiro, (1990).

[20] R.J.B. Sampaio, W.Sun and J.Yuan, On trust region algorithm for nonsmooth optimization, *Applied Mathematics and Computation,* **85** (1997), 109-116.

[21] W. Sun, On convergence of an iterative method for minimax problem, *J. Australian Mathematics Society, Series B*, **39** (1997), 280-292.

[22] W. Sun, Newton's method and quasi-Newton-SQP method for general LC^1 constrained optimization, *Applied Mathematics and Computation,* **92** (1998), 69-84.

[23] J.F. Toland, Duality in nonconvex optimization, *J. of Mathematical Analysis and Applications,* **66** (1978), 399–415.

[24] Y. Yuan and W. Sun, Optimization Theory and Methods, Science Press, Beijing, 1997.

[25] W.I. Zangwill, Nonlinear Programming : A Unified Approach, Prentice-Hall, 1969.