Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 6, No. 3, pp. 483 – 496, September 2010

TRUNCATED CODIFFERENTIAL METHOD FOR NONSMOOTH CONVEX OPTIMIZATION

A.M. Bagirov*, A. Nazari Ganjehlou, J. Ugon and A.H. Tor

Abstract: In this paper a new algorithm to minimize convex functions is developed. This algorithm is based on the concept of codifferential. Since the computation of whole codifferential is not always possible we propose an algorithm for computation of descent directions using only a few elements from the codifferential. The convergence of the proposed minimization algorithm is proved and results of numerical experiments using a set of test problems with nonsmooth convex objective function are reported. We also compare the proposed algorithm with three different versions of bundle methods. This comparison shows that the proposed method is more robust than bundle methods.

Key words: *nonsmooth optimization, convex optimization, subdifferential, codifferential*

Mathematics Subject Classification: *65K05, 90C25, 90C56*

1 Introduction

Consider the following unconstrained minimization problem:

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \mathbb{R}^n \tag{1.1}$$

where the objective function f is assumed to be proper convex.

Numerical methods for solving Problem (1.1) have been studied extensively. Subgradient methods [23], different versions of bundle methods [3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 18, 19, 20, 24] are among them. In this paper, we propose a method, namely the truncated codifferential method, for solving Problem (1.1).

The notion of codifferential was introduced in [2]. The codifferential mapping is Hausdorff continuous for most of important classes of nonsmooth functions. Despite its good differential properties, only a few algorithms were developed based on the codifferential (see [1, 2, 26]). In these algorithms it is assumed that either the whole codifferential or its subsets can be computed at any point. However, this assumption is too restrictive for many classes of nonsmooth optimization problems.

In this paper, a new codifferential method is proposed for solving Problem (1.1). At each iteration of this method, a few elements from the codifferential are used to find descent directions. Therefore we call this method a truncated codifferential method. It is proved that a sequence generated by the truncated codifferential method converges to solutions of

*Dr. Adil Bagirov is the recipient of an Australian Research Council Australian Research Fellowship (Project number: DP 0666061).

Problem (1.1). Results of numerical experiments using a set of well-known nonsmooth optimization academic test problems are reported and used to compare the proposed algorithm with the bundle method.

The paper is structured as follows: In Section 2, we recall the definition of the codifferential and describe it for convex functions. An algorithm for finding descent directions is presented in Section 3. A truncated codifferential method is introduced and its convergence is studied in Section 4. Results of numerical experiments are reported in Section 5. Section 6 concludes the paper.

We use the following notation in this paper. $\mathbb{R}^n$ is an n-dimensional Euclidean space, $\langle u,v\rangle = \sum_{i=1}^n u_i v_i$ is an inner product in $\mathbb{R}^n$ and $\|\cdot\|$ is the associated Euclidean norm, $\partial f(x)$ is the subdifferential of the convex function f at a point x, co denotes the convex hull of a set, $S_1 = \{x \in \mathbb{R}^n : \ \|x\| = 1\}$ is the unit sphere, $B_\varepsilon(x) = \{y \in \mathbb{R}^n : \ \|y-x\| < \varepsilon\}$ is the open and $\bar{B}_\varepsilon(x) = \mathrm{cl}\, B_\varepsilon(x)$ is the closed ball centered at x with the radius $\varepsilon > 0$.

2 Codifferentials for Convex Functions

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called codifferentiable at $x \in \mathbb{R}^n$ if there exists a pair of convex compact sets $\underline{d}f(x)$ and $\overline{d}f(x)$ of $\mathbb{R}^{n+1}$ such that

$$f(x+\Delta) = f(x) + \max_{(a,v)\in \underline{d}f(x)} [a + \langle v,\Delta\rangle] + \min_{(b,u)\in \overline{d}f(x)} [b + \langle u,\Delta\rangle] + o_x(\Delta), \tag{2.1}$$

for all $\Delta \in \mathbb{R}^n$ and $\frac{o_x(\alpha\Delta)}{\alpha} \to 0$ as $\alpha \to +0$.

The pair $Df(x) = [\underline{d}f(x), \overline{d}f(x)]$ is called a codifferential of f at x, the set $\underline{d}f(x)$ is called hypodifferential and the set $\overline{d}f(x)$ is called hyperdifferential of f at x. Elements of the hypodifferential are called hypogradients and elements of the hyperdifferential are called hypergradients. It is important to note that the codifferential is not uniquely defined [2].

A function f is called hypodifferentiable if $\overline{d}f(x) = \{0_{n+1}\}$. It is called hyperdifferentiable if $\underline{d}f(x) = \{0_{n+1}\}$.

Now consider a convex function $f : \mathbb{R}^n \to \mathbb{R}$, a closed bounded set $U \subset \mathbb{R}^n$, a point $x \in \mathrm{int}\, U$ and for any $z \in U$ take one subgradient $v_z \in \partial f(z)$. At the point x, the subgradient inequality implies that

$$f(x) \ge f(z) + \langle v_z, x-z\rangle, \quad \forall z \in U$$

and

$$f(x) = \max_{z\in U}\{f(z) + \langle v_z, x-z\rangle\}.$$

At a point $x + d \in U$ we have

$$f(x+d) - f(x) = \max_{z\in U}\{f(z) - f(x) + \langle v_z, x+d-z\rangle\}$$

which implies that

$$f(x+d) - f(x) = \max_{(a,v)\in \underline{d}f(x)}\{a + \langle v,d\rangle\}, \tag{2.2}$$

where the set $\underline{d}f(x)$ is the hypodifferential of f at x given by (see [26]):

$$\underline{d}f(x) = \mathrm{cl\,co}\ \{(a,v) \in \mathbb{R}\times\mathbb{R}^n : \ a = f(z) - f(x) + \langle v, x-z\rangle,\ v \in \partial f(z),\ z \in U\}.$$

Thus, convex functions are hypodifferentiable. The condition $0_{n+1} \in \underline{d}f(x^*)$ is both necessary and sufficient for the point x^* to be a solution to Problem 1.1. The hypodifferential can be applied to find descent directions of convex functions at non-stationary points. Indeed, if $0_{n+1} \notin \underline{d}f(x)$ then one can compute

$$\|w\|^2 = \min\left\{\|v\|^2 : \quad v \in \underline{d}f(x)\right\}, \; w = (a,u) \in \mathbb{R} \times \mathbb{R}^n.$$

It is proved in [2] that in this case $u \neq 0_n$. Then we define a direction $g = -\|u\|^{-1}u$. For this direction

$$f'(x,g) \leq -\|u\|,$$

that is g is a descent direction. Unfortunately, computation of the whole hypodifferential $\underline{d}f(x)$ is not always an easy task. Therefore, our first aim in this paper is to develop an algorithm for finding descent directions using only a few elements from hypodifferentials.

3 Computation of a Descent Direction

In order to compute descent directions, we will define a subset of the hypodifferential and will show that this subset is sufficient to find such directions. We take any $\lambda \in (0,1)$ and define the following set:

$$H(x,\lambda) = \operatorname{cl}\operatorname{co}\left\{ w = (a,v) \in \mathbb{R} \times \mathbb{R}^n : \begin{array}{l} \exists y \in B_\lambda(x), \\ v \in \partial f(y), \\ a = f(y) - f(x) - \langle v, y-x \rangle \end{array} \right\}. \tag{3.1}$$

It is clear that $a \leq 0$ for all $w = (a,v) \in H(x,\lambda)$. Since $a = 0$ when $y = x$

$$\max_{w=(a,v)\in H(x,\lambda)} a = 0. \tag{3.2}$$

Let $U \subset \mathbb{R}^n$ be a closed convex set such that $\bar{B}_\lambda(x) \subset \operatorname{int}U$ for all $\lambda \in (0,1)$. In this case it follows from the definition of both the hypodifferential and the set $H(x,\lambda)$ that

$$H(x,\lambda) \subset \underline{d}f(x) \quad \forall\, \lambda \in (0,1).$$

We call the sets $H(x,\lambda)$ truncated codifferentials of the function f at the point x.

Proposition 3.1. *Assume that $0_{n+1} \notin H(x,\lambda)$ for a given $\lambda \in (0,1)$ and*

$$\|w^0\| = \min\{\|w\| : \; w \in H(x,\lambda)\} > 0, \; \textit{with} \;\; w^0 = (a_0, v^0).$$

Then $v^0 \neq 0_n$ and

$$f(x + \lambda g^0) - f(x) \leq -\lambda\|w^0\|, \tag{3.3}$$

where $g^0 = -\|w^0\|^{-1}v^0$.

Proof. The necessary condition for a minimum implies that

$$\langle w^0, w - w^0 \rangle \geq 0 \quad \forall w = (a,v) \in H(x,\lambda)$$

or

$$a_0 a + \langle v^0, v \rangle \geq \|w^0\|^2. \tag{3.4}$$

First we will show that $v^0 \neq 0_n$. Assume the contrary, that is $v^0 = 0_n$. Since $w^0 \neq 0_{n+1}$ we get that $a_0 < 0$. Then it follows from (3.4) that $a_0(a - a_0) \geq 0$ or $a \leq a_0 < 0$. In other words $a < 0$ for all $w = (a,v) \in H(x,\lambda)$ which contradicts (3.2).

Now we will prove (3.3). Dividing both sides of (3.4) by $-\|w^0\|$ we get

$$-\frac{a_0 a}{\|w^0\|} + \langle v, g^0\rangle \leq -\|w^0\|. \tag{3.5}$$

It is clear that $\|w^0\|^{-1}a_0 \in (-1, 0)$ and since $\lambda \in (0, 1)$

$$\mu = -\frac{\lambda a_0}{\|w^0\|} \in (0, 1).$$

Therefore taking into account that $a \leq 0$ and (3.5) we get

$$a + \lambda\langle v, g^0\rangle \leq \mu a + \lambda\langle v, g^0\rangle = -\frac{\lambda a_0}{\|w^0\|} a + \lambda\langle v, g^0\rangle \leq -\lambda\|w^0\|. \tag{3.6}$$

It is obvious that $x + \lambda g^0 \in B_\lambda(x)$. Then it follows from the definition of the set $H(x, \lambda)$ that

$$f(x + \lambda g^0) - f(x) = a + \lambda\langle v, g^0\rangle$$

where $w = (a, v) \in H(x, \lambda)$ and $a = f(x + \lambda g^0) - f(x) - \lambda\langle v, g^0\rangle$, with $v \in \partial f(x + \lambda g^0)$. Then the proof follows from (3.6). □

Proposition 3.1 implies that the set $H(x, \lambda)$ can be used to find descent directions of a function f. Furthermore, this can be done for any $\lambda \in (0, 1)$. Unfortunately, it is not always possible to apply Proposition 3.1 since it assumes the entire set $H(x, \lambda)$ to be known. In fact, computation of the entire set $H(x, \lambda)$ is not always possible. However, Proposition 3.1 shows how an algorithm for finding descent directions can be designed. Such an algorithm will use only a few elements from $H(x, \lambda)$ to compute descent directions.

Let the numbers $\lambda,\ c \in (0, 1)$ and a small enough number $\delta > 0$ be given.

Algoritheorem 3.2. Computation of descent directions at x.

Step 1. Select any $g^1 \in S_1$, and compute $v^1 \in \partial f(x + \lambda g^1)$ and $a_1 = f(x + \lambda g^1) - f(x) - \lambda\langle v^1, g^1\rangle$. Set $\bar{H}_1(x) = \{w^1 = (a_1, v^1)\}$ and $k = 1$.

Step 2. Compute $\bar{w}^k = (\bar{a}_k, \bar{v}^k) \in \mathbb{R} \times \mathbb{R}^n$ as a solution to the following problem:

$$\min \quad \|w\|^2 \quad \text{s.t.} \quad w \in \bar{H}_k(x). \tag{3.7}$$

Step 3. If

$$\|\bar{w}^k\| \leq \delta, \tag{3.8}$$

then **stop**. Otherwise, compute $\bar{g}^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$ and go to Step 4.

Step 4. If

$$f(x + \lambda\bar{g}^k) - f(x) \leq -c\lambda\|\bar{w}^k\|, \tag{3.9}$$

then **stop**. Otherwise, set $g^{k+1} = \bar{g}^k$ and go to Step 5.

Step 5. Compute $v^{k+1} \in \partial f(x + \lambda g^{k+1})$ and $a_{k+1} = f(x + \lambda g^{k+1}) - f(x) - \lambda\langle v^{k+1}, g^{k+1}\rangle$. Construct the set $\bar{H}_{k+1}(x) = \mathrm{co}\,\{\bar{H}_k(x) \bigcup \{w^{k+1} = (a_{k+1}, v^{k+1})\}\}$, set $k := k + 1$ and go to Step 2.

Some explanations on Algorithm 3.2 follow. In Step 1 we select any direction $g^1 \in S_1$ and compute the element of the truncated codifferential in this direction. The least distance between the convex hull of all computed elements of the truncated codifferential and the origin is found in Step 2. This is a quadratic programming problem and algorithms from [4, 11, 21, 22, 25] can be applied to solve it. In numerical experiments, we use the algorithm from [25]. If the least distance is less than a given tolerance $\delta > 0$, then the point x is an approximate stationary point; otherwise, we compute a new search direction in Step 3. If it is the descent direction satisfying (3.9) then the algorithm stops (Step 4). Otherwise, we compute a new element of the truncated codifferential in the direction g^{k+1} in Step 5.

There are some similarities between the ways descent directions are computed in the bundle-type algorithms and in Algorithm 3.2. The latter algorithm is close to the version of the bundle method proposed in [24]. However, in the new algorithm elements of the truncated codifferential are used instead of subgradients.

In the next proposition we show that Algorithm 3.2 terminates in a finite numbers of iterations. A standard technique is used to prove it.

Proposition 3.3. *Assume that f is proper convex function, $\lambda \in (0,1)$ and there exists $K \in (0,\infty)$ such that*

$$\max\{\|w\| : w \in \underline{d}f(x)\} \leq K.$$

If $c \in (0,1)$ and $\delta \in (0,K)$, then Algorithm 3.2 terminates after at most m steps, where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 2, \quad K_1 = 1 - [(1-c)(2K)^{-1}\delta]^2.$$

Proof. Since at a point x for a given $\lambda \in (0,1)$

$$\bar{H}_k(x) \subset H(x,\lambda) \subset \underline{d}f(x)$$

for any $k = 1, 2, \ldots$, it follows that

$$\max\left\{\|w\| : w \in \bar{H}_k(x)\right\} \leq K, \quad k = 1, 2, \ldots \tag{3.10}$$

First, we will show that if neither stopping criteria (3.8) and (3.9) are satisfied, then a new hypogradient w^{k+1} computed in Step 5 does not belong to the set $\bar{H}_k(x)$. Assume the contrary, that is $w^{k+1} \in \bar{H}_k(x)$. In this case $\|\bar{w}^k\| > \delta$ and

$$f(x + \lambda g^{k+1}) - f(x) > -c\lambda\|\bar{w}^k\|.$$

The definition of the hypogradient $w^{k+1} = (a_{k+1}, v^{k+1})$ implies that

$$f(x + \lambda g^{k+1}) - f(x) = a_{k+1} + \lambda\langle v^{k+1}, g^{k+1}\rangle,$$

and we have

$$-c\lambda\|\bar{w}^k\| < a_{k+1} + \lambda\langle v^{k+1}, g^{k+1}\rangle.$$

Putting $g^{k+1} = -\|\bar{w}^k\|^{-1}\bar{v}^k$ we get

$$\langle v^{k+1}, \bar{v}^k\rangle - \frac{\|\bar{w}^k\|}{\lambda} a_{k+1} < c\|\bar{w}^k\|^2. \tag{3.11}$$

Since $\bar{w}^k = \text{argmin}\,\{\|w\|^2 : w \in \bar{H}_k(x)\}$, the necessary condition for a minimum implies that

$$\langle \bar{w}^k, w\rangle \geq \|\bar{w}^k\|^2$$

for all $w \in \bar{H}_k(x)$. Since by assumption $w^{k+1} \in \bar{H}_k(x)$ we get

$$\bar{a}_k a_{k+1} + \langle \bar{v}^k, v^{k+1}\rangle \geq \|\bar{w}^k\|^2. \tag{3.12}$$

Notice that $a_{k+1} \leq 0$ and $\bar{a}_k \geq -\|\bar{w}^k\|$. Then we have $\bar{a}_k a_{k+1} \leq -\|\bar{w}^k\| a_{k+1}$. Combining this with (3.12), we obtain

$$\langle v^{k+1}, \bar{v}^k\rangle - \|\bar{w}^k\| a_{k+1} \geq \|\bar{w}^k\|^2.$$

Finally, taking into account that $\lambda \in (0,1)$ we have

$$\langle v^{k+1}, \bar{v}^k\rangle - \frac{\|\bar{w}^k\|}{\lambda} a_{k+1} \geq \|\bar{w}^k\|^2$$

which contradicts (3.11). Thus, if both (3.8) and (3.9) do not hold then the new hypogradient w^{k+1} allows one to improve the approximation of the set $H(x,\lambda)$.

It is clear that $\|\bar{w}^{k+1}\|^2 \leq \|tw^{k+1} + (1-t)\bar{w}^k\|^2$ for all $t \in [0,1]$, which means

$$\|\bar{w}^{k+1}\|^2 \leq \|\bar{w}^k\|^2 + 2t\langle \bar{w}^k, w^{k+1} - \bar{w}^k\rangle + t^2\|w^{k+1} - \bar{w}^k\|^2.$$

(3.10) implies that

$$\|w^{k+1} - \bar{w}^k\| \leq 2K.$$

It follows from (3.11) that

$$\begin{aligned}\langle \bar{w}^k, w^{k+1}\rangle &= \bar{a}_k a_{k+1} + \langle \bar{v}^k, v^{k+1}\rangle \\ &\leq -\frac{\|\bar{w}^k\|}{\lambda} a_{k+1} + \langle \bar{v}^k, v^{k+1}\rangle \\ &< c\|\bar{w}^k\|^2.\end{aligned}$$

Then we have

$$\|\bar{w}^{k+1}\|^2 < \|\bar{w}^k\|^2 - 2t(1-c)\|\bar{w}^k\|^2 + 4t^2K^2.$$

Let $t_0 = (1-c)(2K)^{-2}\|\bar{w}^k\|^2$. It is clear that $t_0 \in (0,1)$ and therefore

$$\|\bar{w}^{k+1}\|^2 < \left\{1 - [(1-c)(2K)^{-1}\|\bar{w}^k\|]^2\right\}\|\bar{w}^k\|^2. \tag{3.13}$$

Since $\|\bar{w}^k\| > \delta$ for all $k = 1,\ldots,m-1$, it follows from (3.13) that

$$\|\bar{w}^{k+1}\|^2 < \{1 - [(1-c)(2K)^{-1}\delta]^2\}\|\bar{w}^k\|^2.$$

Let $K_1 = 1 - [(1-c)(2K)^{-1}\delta]^2$. Then $K_1 \in (0,1)$ and we have

$$\|\bar{w}^m\|^2 < K_1\|\bar{w}^{m-1}\|^2 < \ldots < K_1^{m-1}\|\bar{w}^1\|^2 < K_1^{m-1}K^2.$$

Thus, the inequality $\|\overline{w}\| \leq \delta$ is satisfied if $K_1^{m-1}K^2 \leq \delta^2$. This inequality must happen after at most m steps where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 2.$$

□

Definition 3.4. A point $x \in \mathbb{R}^n$ is called a (λ, δ)-stationary point of the function f if

$$\min_{w \in H(x,\lambda)} \|w\| \le \delta$$

One can see that Algorithm 3.2 at a given point x after finite many steps either finds a direction of sufficient decrease or determines that the point x is a (λ, δ)-stationary point of the convex function f.

4 A Truncated Codifferential Method

In this section we describe the truncated codifferential method for solving problem (1.1). Let $\lambda \in (0,1),\ \delta > 0,\ c_1 \in (0,1), c_2 \in (0, c_1]$ be given numbers.

Algoritheorem 4.1. The truncated codifferential method for finding (λ, δ)-stationary points.

Step 1. Choose any starting point $x^0 \in \mathbb{R}^n$ and set $k = 0$.

Step 2. Apply Algorithm 3.2 for the computation of the descent direction at $x = x^k$ for given $\delta > 0$ and $c = c_1 \in (0,1)$. This algorithm terminates after finite many steps $m > 0$. As a result, we get the set $\bar{H}_m(x^k) \subset H(x,\lambda) \subset \underline{d}f(x)$ and an element $\bar{w}^k = (\bar{a}_k, \bar{v}^k)$ such that

$$\|\bar{w}^k\|^2 = \min\left\{\|w\|^2 :\ w \in \bar{H}_m(x^k)\right\}.$$

Furthermore, either $\|\bar{w}^k\| \le \delta$ or for the search direction $g^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$

$$f(x^k + \lambda g^k) - f(x^k) \le -c_1\lambda\|\bar{w}^k\|. \tag{4.1}$$

Step 3. If

$$\|\bar{w}^k\| \le \delta \tag{4.2}$$

then **stop**. Otherwise, go to Step 4.

Step 4. Compute $x^{k+1} = x^k + \alpha_k g^k$, where α_k is defined as follows

$$\alpha_k = \operatorname{argmax}\left\{\alpha \ge 0 :\ f(x^k + \alpha g^k) - f(x^k) \le -c_2\alpha\|\bar{w}^k\|\right\}.$$

Set $k := k + 1$ and go to Step 2.

Theorem 4.2. *Assume that the function f is bounded below, i.e.*

$$f_* = \inf\{f(x) : x \in \mathbb{R}^n\} > -\infty. \tag{4.3}$$

Then Algorithm 4.1 terminates after finite many iterations $M > 0$ and generates a (λ, δ)-stationary point x^M where

$$M \le M_0 \equiv \left\lfloor \frac{f(x^0) - f_*}{c_2\lambda\delta} \right\rfloor + 1.$$

Proof. Assume the contrary. Then the sequence $\{x^k\}$ is infinite and points x^k are not (λ, δ)-stationary points. This means that

$$\min \left\{ \|w\| : \ w \in H(x^k, \lambda) \right\} > \delta, \quad k = 1, 2, \dots .$$

Therefore, Algorithm 3.2 will find descent directions and the inequality (4.1) will be satisfied at each iteration k. Since $c_2 \in (0, c_1]$, it follows from (4.1) that $\alpha_k \geq \lambda$. Therefore, we have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &< -c_2 \alpha_k \|\bar{w}^k\| \\ &\leq -c_2 \lambda \|\bar{w}^k\|. \end{aligned}$$

Since $\|\bar{w}^k\| > \delta$ for all $k \geq 0$, we get

$$f(x^{k+1}) - f(x^k) \leq -c_2 \lambda \delta,$$

which implies

$$f(x^{k+1}) \leq f(x^0) - (k+1) c_2 \lambda \delta$$

and therefore $f(x^k) \to -\infty$ as $k \to +\infty$, which contradicts (4.3). It is obvious that the upper bound for the number of iterations M necessary to find the (λ, δ)-stationary point is M_0. □

Remark 4.3. Since $c_2 \leq c_1$, always $\alpha_k \geq \lambda$, and therefore $\lambda > 0$ is a lower bound for α_k. This leads to the following rule for the estimation of α_k. We define a sequence:

$$\theta_l = 2^l \lambda, \quad l = 1, 2, \dots,$$

and α_k is defined as the largest θ_l satisfying the inequality in Step 4 of Algorithm 4.1.

Next we will describe an algorithm for solving Problem (1.1). Let $\{\lambda_k\}, \ \{\delta_k\}$ be sequences such that $\lambda_k \to +0, \ \delta_k \to +0$ as $k \to \infty$ and $\varepsilon_{opt} > 0, \ \delta_{opt} > 0$ be tolerances.

Algoritheorem 4.4. The truncated codifferential method.

Step 1. Choose any starting point $x^0 \in \mathbb{R}^n$, and set $k = 0$.

Step 2. If $\lambda_k \leq \varepsilon_{opt}$ and $\delta_k \leq \delta_{opt}$, then stop.
Step 3. Apply Algorithm 4.1 starting from the point x^k for $\lambda = \lambda_k$ and $\delta = \delta_k$. This algorithm terminates after a finite number of iterations $M_k > 0$, and as a result, it computes a (λ_k, δ_k)-stationary point x^{k+1}.

Step 4. Set $k := k + 1$ and go to Step 2.

For the point $x^0 \in \mathbb{R}^n$, we consider the set $\mathcal{L}(x^0) = \left\{ x \in \mathbb{R}^n : f(x) \leq f(x^0) \right\}$.

Theorem 4.5. *Assume that f is a proper convex function and the set $\mathcal{L}(x^0)$ is bounded. Then every accumulation point of the sequence $\{x^k\}$ generated by Algorithm 4.4 belongs to the set $X^0 = \{x \in \mathbb{R}^n : 0_n \in \partial f(x)\}$.*

Proof. Since the function f is proper convex and the set $\mathcal{L}(x^0)$ is bounded, $f_* > -\infty$. Therefore, conditions of Theorem 4.2 are satisfied, and Algorithm 4.1 generates a sequence of (λ_k, δ_k)-stationary points for all $k \geq 0$. More specifically, the point x^{k+1} is (λ_k, δ_k)-stationary, $k > 0$. Then it follows from Definition 3.4 that

$$\min \left\{ \|w\| : w \in H(x^{k+1}, \lambda_k) \right\} \leq \delta_k. \tag{4.4}$$

It is obvious that $x^k \in \mathcal{L}(x^0)$ for all $k \ge 0$. The boundedness of the set $\mathcal{L}(x^0)$ implies that the sequence $\{x^k\}$ has at least one accumulation point. Let x^* be an accumulation point and $x^{k_i} \to x^*$ as $i \to +\infty$. The inequality in (4.4) implies that

$$\min\left\{\|w\| : w \in H(x^{k_i}, \lambda_{k_i-1})\right\} \le \delta_{k_i-1}.$$

Then there exists $\bar{w} \in H(x^{k_i}, \lambda_{k_i-1})$ such that $\|\bar{w}\| \le \delta_{k_i-1}$. Considering $\bar{w} = (\bar{a}, \bar{v})$ where $\bar{v} \in \partial f(y)$ for some $y \in B_{\lambda_{k_i-1}}(x^{k_i})$, we have $\|\bar{v}\| \le \|\bar{w}\| \le \delta_{k_i-1}$. Therefore,

$$\min\left\{\|v\| : \ v \in \partial f(B_{\lambda_{k_i-1}}(x^{k_i}))\right\} \le \delta_{k_i-1}.$$

Here

$$\partial f(B_{\lambda_{k_i-1}}(x^{k_i})) = \bigcup \left\{\partial f(y) : \ y \in B_{\lambda_{k_i-1}}(x^{k_i})\right\}.$$

The upper semicontinuity of the subdifferential mapping $\partial f(x)$ implies that for any $\varepsilon > 0$ there exists $\eta > 0$ such that

$$\partial f(y) \subset \partial f(x^*) + B_\varepsilon(0_n) \tag{4.5}$$

for all $y \in B_\eta(x^*)$. Since $x^{k_i} \to x^*$, $\delta_{k_i}, \lambda_{k_i} \to +0$ as $i \to +\infty$ there exists $i_0 > 0$ such that $\delta_{k_i} < \varepsilon$ and

$$B_{\lambda_{k_i-1}}(x^{k_i}) \subset B_\eta(x^*)$$

for all $i \ge i_0$. Then it follows from (4.5) that

$$\min\{\|v\| : \ v \in \partial f(x^*)\} \le 2\varepsilon.$$

Since $\varepsilon > 0$ is arbitrary we have $0 \in \partial f(x^*)$. □

5 Results of Numerical Experiments

The efficiency of the truncated codifferential method (TCM) was verified by applying it to some academic test problems with nonsmooth objective functions. Test problems from [16] have been used in numerical experiments. The brief description of test problems are given in Table 1, where the following notation is used:

- n - number of variables;
- f_{opt} - optimal value.

We do not include test problems Colville 1, HS78 and TR48. In problems Colville 1 and HS78 the objective functions are unbounded below ([16] reports one of local minima of these functions) and in the problem TR48 the input data is not fully available. The objective functions in test problems Rosenbrock, Crescent, Mifflin 2, El-Attar, Gill, Steiner 2 and Shell Dual are nonconvex.

In our experiments, we use three bundle algorithms for comparisons:

- Subroutine PBUN is based on the proximal bundle method [13, 18, 19];
- Subroutine PNEW is based on the bundle-Newton method [14];
- Subroutine PVAR is based on the variable metric method [15].

Table 1: The brief description of test problems

Problem	n	f_{opt}	Problem	n	f_{opt}
Rosenbrock	2	0	Shor	5	22.600162
Crescent	2	0	El-Attar	6	0.5598131
CB2	2	1.9522245	Maxquad	10	-0.8414083
CB3	2	2	Gill	10	9.7857721
DEM	2	-3	Steiner 2	12	16.703838
QL	2	7.2	Maxq	20	0
LQ	2	-1.4142136	Maxl	20	0
Mifflin 1	2	-1	Goffin	50	0
Mifflin 2	2	-1	MXHILB	50	0
Wolfe	2	-8	L1HILB	50	0
Rosen-Suzuki	4	-44	Shell Dual	15	32.348679

Brief description of these algorithms and subroutines can be found in [17].

In Algorithm 4.4 parameters were chosen as follows: $c_1 = 0.2,\ c_2 = 0.05,\ \delta_k \equiv \delta_{opt} = 10^{-7},\ \lambda_{k+1} = 0.2\lambda_k,\ k \geq 1,\ \lambda_1 = 1$ and $\varepsilon_{opt} = 10^{-10}$. We implemented all algorithms in Fortran 95 and compiled it using the Lahey Fortran compiler on a 1.83GHz Intel Pentium IV CPU with 1GB of RAM running Windows XP.

Table 2: Results of numerical experiments with given starting points

Prob.	TCM			PBUN			PNEW			PVAR		
	f	n_f	n_{sub}	f	n_{it}	n_f, n_{sub}	f	n_{it}	n_f, n_{sub}	f	n_{it}	n_f, n_{sub}
Rosenbrock	0	335	133	0	45	48	0	58	59	0	56	57
Crescent	0	152	55	0	29	31	0	7	8	0	15	16
CB2	1.95222	291	93	1.95222	31	33	1.95222	15	17	1.95222	26	26
CB3	2	239	91	2	13	15	2.00001	10	11	2	17	17
DEM	-3	244	94	-3	17	19	-3	14	15	-3	22	22
QL	7.2	224	92	7.2	13	15	7.2	3	5	7.2	22	22
LQ	-1.41421	114	44	-1.41421	18	21	-1.41421	16	17	-1.41421	14	14
Mifflin 1	-1	245	86	-1	54	56	-1	11	13	-1	58	63
Mifflin 2	-1	396	91	-1	13	15	-1	10	11	-1	62	62
Wolfe	-8	354	151	-8	44	47	-8	24	25	-8	39	39
Rosen-Suzuki	-44	337	120	-44	43	45	-44	13	15	-44	74	75
Shor	22.60016	458	257	22.60016	27	29	22.60017	7	8	22.60016	49	49
El-Attar	0.55981	858	567	0.55982	144	145	0.55981	116	123	0.55982	114	114
Maxquad	-0.84141	1089	697	-0.84141	74	75	-0.84141	12	14	-0.84140	112	112
Gill	9.78610	694	247	9.78578	201	203	9.78577	68	72	9.78586	215	215
Steiner 2	16.70384	894	656	16.70384	117	132	16.70386	40	42	16.70384	98	99
Maxq	0	894	332	0	148	149	0	36	37	0	128	129
Maxl	0	662	330	0	39	40	0	24	25	0	22	22
Goffin	0	2533	2163	0	52	53	0	111	114	0.00002	402	403
MXHILB	0	518	390	0	22	23	0	14	15	0.00001	59	60
L1HILB	0.00001	573	434	0.00001	73	74	0	16	17	0	72	72
Shell Dual	32.34885	2499	1685	32.35581	643	662	32.38115	382	434	32.99505	152	154

First we applied all algorithms using starting points from [16]. Results are presented in Table 2, where we report the value f of the objective function at the final point, the number of function and subgradient evaluations (n_f and n_{sub}, respectively) and the number of iterations n_{it} for bundle methods. Algorithm PNEW also computes a substitute for the Hessian matrix. Since the number of the Hessian matrix's substitute evaluations $n_H = n_{it} + 1$, we do not include them in this and other tables. The results presented in Table 2 demonstrate that bundle methods perform better than the TCM on all problems, except the Shell Dual problem. Bundle methods failed to solve the Shell Dual problem with high accuracy. These methods use significantly less function and subgradient evaluations than the TCM. The TCM requires significantly more function and subgradient evaluations as it approaches to a solution.

Then we applied all algorithms starting from 20 randomly generated points for each problem. Results are presented in Tables 3 and 4. In Table 3 we report n_b - the number of successful runs considering the best known solution. We say that an algorithm finds the best solution with respect to a tolerance $\varepsilon > 0$ if

$$\bar{f} - f_{opt} \leq \varepsilon(1 + |f_{opt}|),$$

where f_{opt} is the best known objective function value and $\bar{f}$ is the best value of the objective function found by an algorithm. In our experiments $\varepsilon = 10^{-4}$.

Results presented demonstrate that the truncated codifferential method is more accurate than bundle methods. This method is not sensitive to the choice of starting points. Moreover, the success of bundle methods depends on the choice of starting points even for convex problems. The truncated codifferential method is also more successful than bundle methods on nonconvex problems (El-Attar, Shell Dual). These results confirm that the truncated codifferential method is more robust and accurate than bundle methods.

It should be also noted that bundle methods contain some parameters and the adjustment of these parameters, especially the penalty parameter, is a key point. For some problems the choice of the values of these parameters is crucial to ensure convergence. On the other hand the TCM does not depend on any parameter to be adjusted.

Table 3: Results of numerical experiments with 20 starting points

Prob.	TCM	PBUN	PNEW	PVAR
Rosenbrock	20	20	20	20
Crescent	20	11	20	20
CB2	20	11	20	19
CB3	20	12	13	18
DEM	20	20	19	19
QL	20	20	20	20
LQ	20	20	20	17
Mifflin 1	20	20	20	16
Mifflin 2	20	20	20	20
Wolfe	20	13	20	3
Rosen-Suzuki	20	20	20	20
Shor	20	20	20	20
El-Attar	20	19	9	3
Maxquad	20	20	20	19
Gill	20	20	20	13
Steiner 2	20	20	20	20
Maxq	20	20	20	20
Maxl	20	20	20	19
Goffin	20	8	20	14
MXHILB	20	20	20	17
L1HILB	20	4	20	8
Shell Dual	20	9	3	6
Total	440	367	406	351

Table 4 presents the average number of iterations (n_{it}), the average number of objective function (n_f) and subgradient (n_{sub}) evaluations over 20 runs of algorithms. For bundle methods $n_f = n_{sub}$. Results presented in this table demonstrate that the TCM uses more function and subgradient evaluations than bundle algorithms.

Table 4: The average number of iterations, function and subgradient evaluations

Prob.	TCM			PBUN		PNEW		PVAR	
	n_{it}	n_f	n_{sub}	n_{it}	n_f, n_{sub}	n_{it}	n_f, n_{sub}	n_{it}	n_f, n_{sub}
Rosenbrock	53	305	133	260	263	133	135	85	85
Crescent	48	303	118	21	24	7	8	24	25
CB2	47	294	125	22	23	32	33	34	34
CB3	38	227	113	19	21	28	29	34	34
DEM	30	160	86	18	20	13	14	26	27
QL	38	207	101	25	27	4	6	22	22
LQ	48	321	124	9	12	5	6	18	18
Mifflin 1	44	256	105	9	11	17	18	262	228
Mifflin 2	47	296	113	11	13	10	11	40	40
Wolfe	51	321	137	46	50	43	44	2519	2471
Rosen-Suzuki	61	405	196	49	51	15	16	61	61
Shor	67	466	242	48	50	7	8	46	46
El-Attar	130	1001	567	285	288	438	444	151	152
Maxquad	147	1189	762	65	66	13	15	118	118
Gill	105	825	458	352	354	204	207	1117	1121
Steiner 2	87	706	421	130	147	56	57	103	103
Maxq	97	851	325	162	163	37	38	127	127
Maxl	75	689	364	37	38	24	25	139	140
Goffin	117	2343	1976	71	73	110	113	365	382
MXHILB	77	636	377	510	511	38	39	85	87
L1HILB	98	939	657	207	209	74	75	110	111
Shell Dual	205	2173	1487	999	1052	387	434	300	303

6 Conclusions

In this paper we developed the truncated codifferential method for minimizing convex functions. In this method, at each iteration only a few elements from the hypodifferential of the objective function are used to compute descent directions. It is proved that the proposed method converges to minimizers of a convex function.

We presented results of numerical experiments and compared the proposed method with three versions of bundle methods. The computational results show that the proposed method is not sensitive to the choice of starting points whereas performance of all three versions of the bundle method depends on starting points. The truncated codifferential method may locate the solution with higher accuracy than the bundle methods. Therefore the truncated codifferential method is more robust and accurate than the bundle methods. However, the proposed method uses more function and subgradient evaluations than the bundle methods.

Acknowledgements

The authors would like to thank two anonymous referees for their valuable comments which improved the quality of the paper.

References

[1] V.F. Demyanov, A.M. Bagirov and A.M. Rubinov, A method of truncated codifferential with application to some problems of cluster analysis, *J. Global Optim.* 23 (2002) 63–80.

[2] V.F. Demyanov and A.M. Rubinov, *Constructive Nonsmooth Analysis*, Peter Lang, Frankfurt am Main, 1995.

[3] A. Frangioni, Generalized bundle methods, *SIAM J. Optim.* 113 (2002) 117–156.

[4] A. Frangioni, Solving semidefinite quadratic problems within nonsmooth optimization algorithms, *Comput. Oper. Res.* 23 (1996) 1099–1118.

[5] A. Fuduli, M. Gaudioso and G. Giallombardo, Minimizing nonconvex nonsmooth functions via cutting planes and proximity control, *SIAM J. Optim.* 14 (2004) 743–756.

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

[7] M. Gaudioso and M.F. Monaco, A bundle type approach to the unconstrained minimization of convex nonsmooth functions, *Math. Program.* 23 (1982) 216–226.

[8] W. Hare and C. Sagastizabal, Computing proximal points of nonconvex functions, *Math. Program.* 116 (2008) 221–258.

[9] J.B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms,* Vol. 1 and 2, Springer Verlag, Heidelberg, 1993.

[10] K.C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization,* Lecture Notes in Mathematics, Springer-Verlag, Berlin, Vol. 1133, 1985.

[11] K.C. Kiwiel, A dual method for certain positive semidefinite quadratic programming problems, *SIAM J. Sci. Stat. Comput.* 10 (1989) 175–186.

[12] K.C. Kiwiel, Proximal control in bundle methods for convex nondifferentiable minimization, *Math. Program.* 29 (1990) 105–122.

[13] C. Lemarechal and J. Zowe, A condensed introduction to bundle methods in nonsmooth optimization. in *Algorithms for Continuous Optimization,* E. Spedicato (Ed.), Kluwer Academic Publishers, Hingham, MA, 1994, pp. 357–382.

[14] L. Lukśan and J. Vlćek, A bundle Newton method for nonsmooth unconstrained minimization, *Math. Program.* 83 (1998) 373–391.

[15] L. Lukśan and J. Vlćek, Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *J. Optim. Theory Appl.* 102 (1999) 593–613.

[16] L. Lukśan and J. Vlćek, *Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization,* Technical Report No. 78, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2000.

[17] L. Lukśan and J. Vlćek, Algorithm 811: NDA: Algorithms for nondifferentiable optimization, *ACM Trans. Math. Soft.* 27 (2001) 193–213.

[18] M.M. Makela and P. Neittaanmaki, *Nonsmooth Optimization*, World Scientific, Singapore, 1992.

[19] R. Mifflin, An algorithm for constrained optimization with semismooth functions, *Math. Oper. Res.* 2 (1977) 191–207.

[20] R. Mifflin, A quasi-second-order proximal bundle algorithm, *Math. Program.* 73 (1996) 51–72.

[21] E.A. Nurminski, Convergence of the suitable affine subspace method for finding the least distance to a simplex, *Comput. Math. Math. Phys.* 45 (2005) 1915–1922.

[22] E.A. Nurminski, Projection onto polyhedra in outer representation, *Comput. Math. Math. Phys.* 48 (2008) 367–375.

[23] N.Z. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer-Verlag, Heidelberg, 1985.

[24] P.H. Wolfe, A method of conjugate subgradients of minimizing nondifferentiable convex functions, *Math. Program. Study* 3 (1975) 145–173.

[25] P.H. Wolfe, Finding the nearest point in a polytope, *Math. Program.* 11 (1976) 128–149.

[26] A. Zaffaroni, Continuous approximations, codifferentiable functions and minimization methods, in *Nonconvex Optimization and its Applications: Quasidifferentiability and Related Topics*, V.F. Demyanov and A.M. Rubinov (eds.), Kluwer Academic Publishers, Dordrecht, 2000, pp. 361–391.

Manuscript received 25 March 2009
revised 10 August 2009, 11 September 2009, 16 September 2009
accepted for publication 18 September 2009

A.M. Bagirov
Centre for Informatics and Applied Optimization
School of Information Technology and Mathematical Sciences
University of Ballarat, Victoria 3353, Australia
E-mail address: a.bagirov@ballarat.edu.au

A. Nazari Ganjehlou
Centre for Informatics and Applied Optimization
School of Information Technology and Mathematical Sciences
University of Ballarat, Victoria 3353, Australia
E-mail address: a.nazari@ballarat.edu.au

J. Ugon
Centre for Informatics and Applied Optimization
School of Information Technology and Mathematical Sciences
University of Ballarat, Victoria 3353, Australia
E-mail address: j.ugon@ballarat.edu.au

A.H. Tor
Department of Mathematics, Middle East Technical University
Ankara and Department of Mathematics
Yüzüncü Yil, Van, Turkey
E-mail address: htor@metu.edu.tr