A Non-monotone Conjugate Subgradient Type Method for Minimization of Convex Functions

Igor Konnov

Received: 20 April 2019 / Accepted: 13 September 2019 / Published online: 26 September 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract
We suggest a conjugate subgradient type method without any line search for minimization of convex non-differentiable functions. Unlike the custom methods of this class, it does not require monotone decrease in the goal function and reduces the implementation cost of each iteration essentially. At the same time, its step-size procedure takes into account behavior of the method along the iteration points. The preliminary results of computational experiments confirm the efficiency of the proposed modification.

Keywords Convex minimization problems · Non-differentiable functions · Conjugate subgradient method · Simple step-size choice · Convergence properties

Mathematics Subject Classification 90C25 · 90C30

1 Introduction

Given a convex and continuous, but not necessarily differentiable function defined on a finite-dimensional Euclidean space, we can consider the well-known general problem of minimizing this function over the whole space. There exist a great number of significant applications of convex minimization problems having just non-differentiable goal functions; see, e.g., [1–3] and the references therein. For this reason, their theory and methods were developed rather well. In particular, many different iterative methods were proposed for finding solutions of convex non-differentiable (non-smooth) minimization problems; see, e.g., [1–6] and the references therein. We recall that most
applications admit calculation of only one arbitrary taken element from the subdifferential of the cost function at any point.

During a rather long time, most efforts were concentrated on developing more powerful and rapidly convergent methods within this setting, such as space dilation and bundle type ones, which admit complex transformations at each iteration. However, significant areas of applications related to decision-making in industrial, transportation, information and communication systems, having large dimensionality and inexact data together with scattered necessary information force one to avoid complex transformations and even line search procedures and to apply mostly simple methods, whose iteration computation expenses and accuracy requirements are rather low. We observe that the problem of creation of efficient low-cost non-smooth optimization methods is more difficult in comparison with that in the smooth case.

For instance, the choice of the step size in the simplest subgradient method, which provides convergence to a solution, can conform to various rules. The most popular is the so-called divergent series rule. However, this rule yields rather slow convergence; see, e.g., [1, Chapter II, §§1–2] and [2, Chapter V, §3]. We notice that the step-size choice is then non-adaptive to iterates. There are several ways to speed up convergence of the subgradient methods with the same storage and computation expenses per iteration.

One of them consists in calculating the step size via utilization of a priori information such as the optimal value or some condition numbers; see, e.g., [1,2]. However, it is usually difficult to calculate these values exactly, whereas utilization of their inexact estimates may again lead to slow convergence. The second way consists in creating simple descent or relaxation subgradient methods; see [7–12] and also [4,6,13,14]. These methods usually demonstrate more stable convergence in comparison with the custom subgradient methods. It should be noticed that they involve adaptive step-size rules. In the case of a strongly convex and smooth cost function, they attain a linear rate of convergence, see [6,7,10,14], whereas this is not the case for the custom (sub)gradient method with the divergent series step-size rule. However, the requirement of monotone decrease in the goal function at each iteration leads to rather small step size and may also result in low convergence. The third way consists in utilization of averaging procedures for subgradients from several previous iteration points in order to find the current direction; see [13,15,16]. This property is also used in most descent subgradient methods, but here no descent is required for iteration points. As a result, one can obtain very flexible methods, which, however, require mostly divergent series-type rules for providing convergence, so that their convergence rate estimate is usually the same as that of the custom subgradient method.

Therefore, one is interested in further development of these directions in order to create more efficient versions of subgradient methods. Recently, a simple adaptive step-size procedure for smooth optimization methods was proposed in [17–19]. In this paper, we follow this approach, but due to the non-smooth goal function we combine it with some procedures and rules from conjugate subgradient methods, see [7,20], and non-monotone averaging subgradient methods (see [13, Chapter IV]). Our method admits different changes of the step size and a wide variety of implementation rules. It does not utilize a priori information, but takes into account behavior of the iteration sequence. The preliminary results of computational experiments confirm its efficiency.
2 Basic Preliminaries

Let \( f : \mathbb{R}^n \to \mathbb{R} \) be a convex, but not necessarily differentiable function defined on the real \( n \)-dimensional Euclidean space \( \mathbb{R}^n \). Then, we can define the problem of minimizing \( f \) over \( \mathbb{R}^n \), or briefly,

\[
\min_{x \in \mathbb{R}^n} f(x). \tag{1}
\]

We denote by \( X^* \) and \( f^* \) the set of solutions of problem (1) and the optimal value in (1), respectively.

We now recall two well-known auxiliary properties of vector sequences; see [7,20] and [14, Section 4.3]. Given a set \( X \), we denote by \( \text{conv} X \) the convex hull of \( X \) and by \( \text{Nr} X \) the element of \( X \) nearest to origin. Also, we denote by \( B(x, \varepsilon) \) the closed ball of a radius \( \varepsilon \) around \( x \), i.e.,

\[
B(x, \varepsilon) = \{ y \in \mathbb{R}^n : \| y - x \| \leq \varepsilon \}.
\]

**Lemma 2.1** Let \( \{p^i\} \) and \( \{g^i\} \) be sequences in \( \mathbb{R}^n \) such that

\[
p^0 = g^0, \quad p^{i+1} = \text{Nr} \, \text{conv} \{p^i, g^{i+1}\}, \quad i = 0, 1, \ldots
\]

Then,

\[
p^i \in \text{conv} \{g^0, \ldots, g^i\}, \quad i = 0, 1, \ldots
\]

**Lemma 2.2** Let \( \{p^i\} \) and \( \{g^i\} \) be sequences in \( \mathbb{R}^n \) such that (2) holds and

\[
\|g^i\| \leq C < \infty, \quad \langle g^{i+1}, p^i \rangle \leq \theta \|p^i\|^2, \quad i = 0, 1, \ldots, \theta \in ]0, 1[.
\]

Then,

\[
\|p^i\| \leq C/((1 - \theta)\sqrt{i + 1}) \quad \text{for } i = 0, 1, \ldots
\]

We recall some concepts and properties from convex analysis. If a function \( f : \mathbb{R}^n \to \mathbb{R} \) is convex, one can define its subdifferential:

\[
\partial f(x) = \{ g \in \mathbb{R}^n : f(y) - f(x) \geq \langle g, y - x \rangle \, \forall y \in \mathbb{R}^n \},
\]

which is non-empty, convex and compact at any point \( x \); see [21, Section 23]. Moreover, the subdifferential mapping \( x \mapsto \partial f(x) \) is upper semicontinuous at any point \( x \); see [21, Theorems 24.4 and 24.5].

We need also a simple deviation estimate for convex non-smooth functions; see [6, Lemma 4.3].
Lemma 2.3 Let
\[ p = \sum_{j=1}^{m} \mu_j g^j, \quad g^j \in \partial f(y^j), \quad y^j \in B(x, \delta), \quad \sum_{j=1}^{m} \mu_j = 1, \quad \mu_j \geq 0, \quad j = 1, \ldots, m. \]

Then,
\[ f(y) - \sum_{j=1}^{m} \mu_j f(y^j) \geq \langle p, y - x \rangle - \delta L_{x,\delta} \]
for any \( y \in \mathbb{R}^n \) where
\[ L_{x,\delta} = \max_{j=1,\ldots,m} \|g^j\|. \]

Proof For any \( y \), we have
\[ f(y) - f(y^j) \geq \langle g^j, y - y^j \rangle \geq \langle g^j, y - x \rangle - \delta \|g^j\| \]
\[ \geq \langle g^j, y - x \rangle - \delta L_{x,\delta} \]
for \( j = 1, \ldots, m \). Multiplying these inequalities by \( \mu_j \) and summing over \( j = 1, \ldots, m \), gives (3).

3 The Basic Method and its Convergence

We will use the following set of basic assumptions.

(A1) The function \( f : \mathbb{R}^n \to \mathbb{R} \) is convex.

(A2) There exists a number \( \alpha \) such that the set
\[ E_\alpha = \{ x \in \mathbb{R}^n : f(x) \leq \alpha \} \]
is non-empty and bounded.

Clearly, (A2) is a general coercivity condition, which implies that the set \( E_\alpha \) is bounded for any \( \alpha \) if non-empty. If (A1) and (A2) hold, problem (1) has a solution.

We now describe the non-monotone conjugate subgradient type method for problem (1), which involves a simple adaptive step-size procedure without line search.

Method (CSGM)

Step 0 Choose a point \( x^0 \), numbers \( \mu \) and \( \theta \in ]0, 1[ \), and sequences \( \{\alpha_s\}, \{\beta_m\}, \{\eta_t\}, \{d_t\} \) such that
\[ \alpha_s \in ]0, 1[ , \{\alpha_s\} \to 0; \quad \beta_m > 0, \{\beta_m\} \to 0, \sum_{m=0}^{\infty} \beta_m = \infty; \quad (4) \]
\[ \eta_t > 0, \{\eta_t\} \to 0; \quad d_t > 0, \{d_t\} \to 0. \quad (5) \]
Set $k := 0$, $l := 0$, $m := 0$, $s := 0$, $t := 0$, $b := 0$, $\lambda_0 := \beta_0$, $y^0 := x^0$. Compute $g^0 \in \partial f(x^0)$, set $u^0 := x^0$, $p^0 := g^0$.

**Step 1** If $\|p^k\| \leq \eta_l$, set $p^k := g^k$, $l := l + 1$, $t := t + 1$, $b := 0$. (norm restart)

**Step 2** Set $y^{k+1} := x^k - \lambda_k p^k$, $b := b + \lambda_k \|p^k\|$, compute $g^{k+1} \in \partial f(y^{k+1})$. If

$$f(y^{k+1}) \leq f(x^k) - \theta \lambda_k \|p^k\|^2,$$

set $x^{k+1} := y^{k+1}$, $\lambda_{k+1} := \lambda_k$ and go to Step 4. (descent step)

**Step 3** Set $\lambda_{k+1} := \alpha_s \beta_m$, $s := s + 1$. If $f(y^{k+1}) \leq \mu_s$ set $x^{k+1} := y^{k+1}$ and go to Step 4. Otherwise take $p^{k+1} := g^{k+1} \in \partial f(u^k)$, set $x^{k+1} := u^k$, $u^{k+1} := u^k$, $m := m + 1$, $\lambda_{k+1} := \beta_m$, $k := k + 1$, $t := t + 1$, $s := 0$, $b := 0$ and go to Step 1. (function value restart)

**Step 4** If $f(x^{k+1}) < f(u^k)$, set $u^{k+1} := x^{k+1}$. If $b > d_l$, take $p^{k+1} := g^{k+1} \in \partial f(x^{k+1})$, set $m := m + 1$, $\lambda_{k+1} := \beta_m$, $k := k + 1$, $t := t + 1$, $s := 0$, $b := 0$ and go to Step 1. (distance restart)

**Step 5** Set

$$p^{k+1} := \text{Nr conv}(p^k, g^{k+1}),$$

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

According to the description, at each iteration, the current direction $p^k$ belongs to the convex hull of the subgradients from several previous iteration points. In the restart cases, we set $p^k := g^k \in \partial f(x^k)$. The norm restart implies the decrease in the parameters $\eta_l$ and $d_l$ since the current point approximates a solution. The variable $l$ is a counter for the norm restarts. The distance and function value restarts imply the decrease in the parameters $\eta_l$ and $d_l$ and new starting step-size value $\beta_m$ since the current approximation appeared not so precise.

The strategy of descent steps and norm restarts follows the conjugate subgradient and adaptive gradient methods from [7,12,18]. Observe that

$$\lambda_k \langle g^{k+1}, p^k \rangle = \langle g^{k+1}, x^k - y^{k+1} \rangle \leq f(x^k) - f(y^{k+1}) \leq \theta \lambda_k \|p^k\|^2,$$

if (6) does not hold. Then, the conditions of Lemma 2.2 are satisfied. The strategy of function value restarts follows the adaptive gradient method from [18]. The distance restart strategy follows in part the non-monotone averaging subgradient methods from [13, Chapter IV]. Therefore, (CSGM) is an intermediate method since it admits steps without descent, but descent and non-descent steps imply different choices of the next step-size values. Observe that the sequence $\{u^k\}$ simply contains the best current points of the sequence $\{x^k\}$, i.e., $f(u^k) = \varphi_k$ where $\varphi_k = \min_{0 \leq i \leq k} f(x^i)$.

In order to guarantee convergence of (CSGM), we have to specialize the choice of the parameter $\mu$. Take any $x^* \in X^*$, and then, there exists a ball $B(x^*, \rho) \supseteq E_{\alpha'}$ where $\alpha' = f(x^0)$. Let $\rho'$ denote the radius of the smallest ball $B(x^*, \rho')$ containing $E_{\alpha'}$ and let $E_{\mu'} \supseteq B(x^*, \rho')$.  

@ Springer
Theorem 3.1 Let the assumptions (A1)–(A2) be fulfilled and \( \mu \geq \mu' \). Then,

(i) The sequence \( \{x^k\} \) has a limit point, which belongs to the set \( X^* \).
(ii) It holds that

\[
\lim_{k \to \infty} \varphi_k = f^*.
\] (7)

Proof First, we observe that the sequence \( \{x^k\} \) belongs to the bounded set \( E_\mu \) and must have limit points.

Let us consider several possible cases.

Case 1 The number of changes of the index \( l \) is infinite. Then, we have \( t \to \infty \) and \( \{d_l\} \to 0 \) because of (5). Besides, in accordance with Lemma 2.1 there exist subsequences of indices \( \{k_l\} \) and \( \{t_l\} \) such that

\[
\|p_{k_l}\| \leq \eta_l, \quad p_{k_l} \in \text{conv} \bigcup_{\|y-x_{k_l}\| \leq d_{t_l}} \partial f(y), \quad \{d_{t_l}\} \to 0,
\]

for \( l = 0, 1, \ldots \). Take an arbitrary limit point \( x^* \) of the subsequence \( \{x^{k_l}\} \). Without loss of generality, we suppose that

\[
\lim_{l \to \infty} x^{k_l} = x^*.
\]

Since the mapping \( x \mapsto \partial f(x) \) is upper semicontinuous, these properties together with (5) yield \( x^* \in X^* \). It follows that (7) is also true.

Case 2 The number of changes of the index \( l \) is finite. This situation admits two possible cases.

Case 2a The number of changes of the index \( m \) is finite. Then, we have \( t \leq t' < \infty \), \( d_l \geq d > 0 \) and \( \eta_l \geq \bar{\eta} > 0 \), and hence,

\[
\|p_k\| \geq \bar{\eta} \quad \text{for} \quad k = 0, 1, \ldots
\] (8)

If we suppose that the number of changes of the index \( s \) is finite, from (6) we have

\[
f(x^{k+1}) \leq f(x^k) - \theta \lambda_k (\bar{\eta})^2
\]

for \( k \) large enough. But now (4) gives

\[
\sum_{k=0}^{\infty} \lambda_k = \infty,
\]

and hence, \( f(x^k) \to -\infty \) as \( k \to \infty \), which is a contradiction. Otherwise, the number of changes of the index \( s \) is infinite. Then, Lemma 2.2 gives \( \|p_k\| \to 0 \) as \( k \to \infty \), which contradicts (8). That is, this case is impossible.
Case 2b The number of changes of the index \( m \) is infinite. Then, we have \( t \to \infty \) and \( \{d_t\} \to 0 \). For a number \( \varepsilon > 0 \), we set

\[
T_\varepsilon = \{ x \in \mathbb{R}^n : f(x) < f^* + \varepsilon \}
\]

and

\[
B(T_\varepsilon, \varepsilon) = \{ x \in \mathbb{R}^n : \inf_{u \in T_\varepsilon} \|x - u\| \leq \varepsilon \}.
\]

Let us suppose that there exists a number \( \varepsilon > 0 \) such that \( x^k \notin B(T_\varepsilon, \varepsilon) \) for any \( k = 0, 1, \ldots \). Take a point \( x^* \in X^* \) related to the choice of \( \rho' \), and then,

\[
\|x^{k+1} - x^*\|^2 = \|x^k - x^*\|^2 - 2\lambda_k \langle p^k, x^k - x^* \rangle + \lambda_k^2 \|p^k\|^2.
\]

Next, at each point \( x^k \) we have

\[
p^k \in \text{conv}\{g^j \}_{j \in J_k}, \quad g^j \in \partial f(y^j), \quad y^j \in B(x^k, d_t),
\]

where \( t \) is the current index value at the \( k \)th iteration. Take an index \( k' \) large enough such that \( d_t < \varepsilon \) if \( k \geq k' \). This means that \( y^j \notin T_\varepsilon \). Now from Lemma 2.3, we have

\[
-\varepsilon \geq \langle p^k, x^* - x^k \rangle - d_t C_k,
\]

where

\[
C_k = \max_{j \in J_k} \|g^j\| \leq C < \infty.
\]

It follows that

\[
\|x^{k+1} - x^*\|^2 \leq \|x^k - x^*\|^2 - \lambda_k \varepsilon'
\]

for some \( \varepsilon' \in ]0, \varepsilon[ \) and \( k \) large enough. Due the choice of the parameter \( \mu \), this means that all the points \( x^k \) will be contained in the ball \( B(x^*, \rho') \) and (9) holds for \( k \) large enough. It follows that the function value restart does not occur if \( k \geq k'' \) where \( k'' \geq k' \) is chosen large enough. Since the number of changes of the index \( m \) is infinite, (4) gives

\[
\sum_{k=0}^{\infty} \lambda_k = \infty,
\]

and hence, \( \|x^k - x^*\| < 0 \) for \( k \) large enough, which is a contradiction.

Therefore, for each \( \varepsilon > 0 \) there exists a number \( k(\varepsilon) \) such that \( x^{k(\varepsilon)} \in B(T_\varepsilon, \varepsilon) \). Taking any sequence \( \{\varepsilon_h\} \to 0 \), we obtain that any limit point of the corresponding sequence \( \{x^{k(\varepsilon_h)}\} \) belongs to \( X^* \) and that (7) holds true. \( \square \)
4 Modifications and Implementation

The above descent method admits various modifications and extensions. For instance, we can determine the sequence \( \{\eta_l\} \) to be dependent only of the index \( l \) and take

\[ \eta_l > 0, \{\eta_l\} \to 0 \]

instead of that in (5). Besides, in Step 1 we should check now the inequality \( \|p^k\| \leq \eta_l \). Then, all the assertions of Theorem 3.1 remain true with small modifications in the proof. Then, \( \eta_l \) is not changed in the cases of the distance and function value restarts when \( d_l \) decreases. The preferences of both the strategies need further investigations. Next, it is not necessary in fact to evaluate \( \mu' \) directly. We can first take any \( \mu \geq f(x^0) \) and fix a number \( \alpha'' > 0 \). Afterward, we should set \( \mu := \mu + \alpha'' \) if the function value restart occurs.

We now illustrate opportunities of (CSGM) with its behavior in the smooth case. Let us suppose that the gradient of \( f \) is Lipschitz continuous with constant \( L \). We first fix temporarily the values \( \eta_t = +\infty \) and \( d_t = +\infty \). Then, (CSGM) has norm restarts at each iteration and behaves as the adaptive gradient method from [18]. Next, then by (6) we have

\[ f(x^{k+1}) \leq f(x^k) - \theta \lambda_k \|f'(x^k)\|^2 \]

for \( k \) large enough since this inequality holds if \( \lambda_k \leq \bar{\lambda} = 2(1 - \theta)/L \); see [22, Chapter II, §1]. This means that \( \lambda_k = \lambda' > 0 \) for \( k \) large enough. If we suppose in addition that \( f \) is strongly convex with constant \( \kappa \), then

\[ f(x^k) - f^* \leq (2/\kappa)\|f'(x^k)\|^2, \]

and the above method converges linearly to the optimal value \( f^* \); see [22, Chapter II, §1]. Therefore, all these properties will hold if we choose \( \eta_0 \) and \( d_0 \) large enough and take these parameters in conformity with (5) but tending to 0 slower than linearly. It follows that divergent series rules for parameters do not prevent in general from a linear rate of convergence if they involve adaptive step-size procedures. At the same time, averaging gradients may enhance performance after proper norm restart regulation with evaluation of the goal function properties along current points. For this reason, we should avoid the situation where \( d_t \approx \beta_m \|p^k\| \) in the method in order to guarantee averaging gradient iterations.

We now intend to somewhat specialize the choice of parameters in (CSGM) and describe its implementation, which involves two different rules after norm and distance restarts. That is, the rule after the norm restart corresponds to those in descent subgradient and adaptive gradient methods, whereas the rule after the distance restart follows that in non-descent subgradient methods.

Method (CSGI)
Step 0 Choose a point $x^0$, numbers $\mu$ and $\theta \in ]0, 1[$, and sequences $\{\alpha'_s\}, \{\alpha'_t\}, \{\beta'_m\}, \{\beta''_m\}$ such that

\begin{align}
\alpha'_s \in ]0, 1[, \{\alpha'_t\} & \to 0; \quad \alpha'_t \in ]0, 1[, \{\alpha''_t\} \to 0; \\
\beta'_m > 0, \{\beta'_m\} & \to 0, \sum_{m=0}^{\infty} \beta'_m = \infty; \quad \beta''_m > 0, \{\beta''_m\} \to 0; \\
\beta''_m > 0, \{\beta''_m\} & \to 0. \tag{10}
\end{align}

Set $k := 0, l := 0, m := 0, s := 0, t := 0, b := 0, \lambda_0 := \beta'_t, \eta_0 := \beta''_0, d_0 := \beta''_0$, $y^0 := x^0$. Compute $g^0 \in \partial f(x^0)$, set $u^0 := x^0, p^0 := g^0$.

Step 1 If $\|p^k\| \leq \eta_t$, set $p^k := g^k, \eta_{t+1} := \alpha'_t \beta''_m, d_{t+1} := \alpha'_t \beta''_m, l := l + 1, t := t + 1, b := 0$. (norm restart)

Step 2 Set $y^{k+1} := x^k - \lambda_k p^k, b := b + \lambda_k \|p^k\|$, compute $g^{k+1} \in \partial f(y^{k+1})$. If

\[ f(y^{k+1}) \leq f(x^k) - \theta \lambda_k \|p^k\|^2, \]

set $x^{k+1} := y^{k+1}, \lambda_{k+1} := \lambda_k$ and go to Step 4. (descent step)

Step 3 Set $\lambda_{k+1} := \alpha'_s \beta''_m, s := s + 1$. If $f(y^{k+1}) \leq \mu$, set $x^{k+1} := y^{k+1}$ and go to Step 4. Otherwise take $p^{k+1} := g^{k+1} \in \partial f(u^k)$, set $x^{k+1} := u^k, u^{k+1} := u^k, m := m + 1, \lambda_{k+1} := \beta'_m, \eta_{t+1} := \beta''_m, d_{t+1} := \beta''_m, k := k + 1, t := t + 1, s := 0, l := 0, b := 0$ and go to Step 1. (function value restart)

Step 4 If $f(x^{k+1}) < f(u^k)$, set $u^{k+1} := x^{k+1}$. If $b > d_t$, take $p^{k+1} := g^{k+1} \in \partial f(x^{k+1})$, set $m := m + 1, \lambda_{k+1} := \beta'_m, \eta_{t+1} := \beta''_m, d_{t+1} := \beta''_m, k := k + 1, t := t + 1, s := 0, l := 0, b := 0$ and go to Step 1. (distance restart)

Step 5 Set

\[ p^{k+1} := \text{Nr conv}\{p^k, g^{k+1}\}, \]

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

Following the lines of the proof of Theorem 3.1, we can conclude that the same assertions are true for (CSGI).

**Theorem 4.1** Let the assumptions (A1)–(A2) be fulfilled and $\mu \geq \mu'$. Then, the sequence $\{x^k\}$ generated by (CSGI) has a limit point, which belongs to the set $X^*$, besides, relation (7) holds.

Although (CSGI) involves several sequences of parameters, it seems reasonable to apply the same rule for $\{\alpha'_s\}$ and $\{\alpha'_t\}$, as well as the same rule for $\{\beta'_m\}, \{\beta''_m\}$ and $\{\beta'''_m\}$. This in particular means that

\[ \sum_{m=0}^{\infty} \beta''_m = \infty, \sum_{m=0}^{\infty} \beta'''_m = \infty. \]

However, these additional rules are not obligatory for providing convergence.
5 Computational Experiments

In order to check the performance of the proposed method, we carried out computational experiments. The main goal was to compare it with subgradient methods without line search, which have similar storage and computation expenses per iteration.

We first took the simplest subgradient method:

\[ x^{k+1} := x^k - \lambda_k g^k, \quad g^k \in \partial f(x^k), \quad \lambda_k > 0, \]

where the step size \( \lambda_k \) was subordinated to the divergent series rule:

\[ \lim_{k \to \infty} \lambda_k = 0, \quad \sum_{k=0}^{\infty} \lambda_k = \infty. \]

It was abbreviated as (SGM). We also took the non-monotone averaging subgradient method from [16] abbreviated as (NASGM), the simple descent subgradient method from [10,12] abbreviated as (DSGM) and the proposed method (CSGI). We took for comparison the well-known non-smooth test problem from [23], where

\[ f(x) = \max_{i=1,...,m} h_i(x), \quad h_i(x) = b^i \sum_{j=1}^{n} (x_j - a_{ij})^2, \quad i = 1, \ldots, m, \]

with \( n = 5 \) and \( m = 10 \). The coefficients of the quadratic functions are given by the vector \( b = (1, 5, 10, 2, 4, 3, 1.7, 2.5, 6, 3.5)^T \) and by the transposed matrix

\[ A^T = \begin{pmatrix} 0 & 2 & 1 & 1 & 3 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 2 & 4 & 2 & 2 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 1 & 2 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 3 & 2 & 2 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}; \]

see also [1,2]. The problem has the optimal value \( f^* = 22.60016 \). We chose the starting point \( x^0 = (0, 0, 0, 0, 1)^T \).

We compared all the methods for different accuracy \( \varepsilon \) with respect to the goal function deviation

\[ \Delta(x) = f(x) - f^*. \]

They were implemented in Delphi with double precision arithmetic. Namely, we indicate the total number of iterations (it) (or the total number of subgradient calculations) for attaining the desired accuracy \( \varepsilon \).

We took the rule \( \lambda_k = \lambda/(k + 1) \) with \( \lambda = 0.1 \) for (SGM). Since (NASGM) involves several sequences of parameters tending to 0, we decreased them linearly with the same ratio \( \sigma = 0.9 \). For (DSGM), we also used the same rule with the ratio \( \sigma = 0.6 \) for two sequences of parameters tending to 0. For (CSGI), we set \( \theta = 0.3 \) and
Table 1  Comparison of subgradient methods

| SGM | NASGM | DSGM | CSGI |
|-----|-------|------|------|
| $\varepsilon$ | $\varepsilon$ | $\varepsilon$ | $\varepsilon$ |
| $\text{it}$ | $\text{it}$ | $\text{it}$ | $\text{it}$ |
| 0.1  | 81  | 0.1 | 30 | 0.1 | 92 | 0.1 | 141 |
| 0.01 | 320 | 0.01 | 63 | 0.01 | 352 | 0.01 | 253 |
| 0.001 | 1645 | 0.004 | 10000 | 0.001 | 1058 | 0.001 | 466 |
| 0.0001 | 8243 | – | – | 0.0001 | 2809 | 0.0001 | 640 |
| 0.00002 | 35000 | – | – | 0.00001 | 5909 | 0.00001 | 860 |

the rules $\beta_m' = \beta'/(m + 1)$, $\beta_m'' = \beta''/(m + 1)$, $\beta_m''' = \beta'''/(m + 1)$ with $\beta' = 0.05$, $\beta'' = 0.4\|g_0\|$ and $\beta''' = \beta'\|g_0\|/0.7$. Also, we set $\alpha_0' = \alpha_0'' = \sigma = 0.8$ and the rule

$$\alpha_{s+1}' = \sigma \alpha_s', \quad \alpha_{l+1}'' = \sigma \alpha_l''.$$  

For simplicity, we implemented (CSGI) with $\mu = +\infty$, i.e., without function value restarts. The results are given in Table 1.

The implementation of (CSGI) showed rather rapid convergence in comparison with the other methods. Convergence of (NASGM) appeared rather unstable. At the same time, convergence of (DSGM) appeared not so rapid, but stable.

Somewhat different strategies for determining step sizes in subgradient methods with averaging procedures were proposed in [24,25]. They were deduced from theoretical convergence rates for these methods on the class of convex minimization problems. Besides, comparative computational experiments with three methods were also described in [25]. They involved the same subgradient method (13)–(14) with the best theoretical rule:

$$\lambda_k := \lambda/\sqrt{(k + 1)}, \quad \lambda = \|x^0 - x^*\|/L,$$

where $L$ is an upper bound for the norm of subgradients, $x^* \in X^*$. In what follows, we abbreviate this method as (SGMT). In addition, we took the normed version of this subgradient method with the rule:

$$\lambda_k := 1/(\|g^k\|\sqrt{(k + 1)}).$$

In what follows, we abbreviate this method as (SGMN). Next, the so-called simple dual averaging method from [24] was taken. It can be written as follows:
Table 2  Comparison of subgradient methods

|                   | SGMT |            | SGMN |            | ASG |            | DASG |            |
|-------------------|------|------------|------|------------|-----|------------|------|------------|
| $\varepsilon$     | it   | $\varepsilon$ | it   |            | it  |            | it   |            |
| 0.1               | 116  | 0.1        | 317  | 2.038      | 10000 | 0.1       | 324  |            |
| 0.01              | 4510 | 0.01       | 3426 | –          | –   | 0.01       | 3254 |            |
| 0.0013            | 35000 | 0.003      | 35000 | –          | –   | 0.001      | 34169 |            |

\[ x^{k+1} := x^0 - \lambda_k p^k, \quad p^k := \sum_{i=0}^{k} g^i, \quad g^i \in \partial f(x^i), \]

where $\lambda_k$ was chosen as in (15). We abbreviate this method as (ASG). In addition, the so-called method of simple double averaging from [25] was used. It can be written as follows:

\[ x^{k+1} := \mu_k x^k + (1 - \mu_k) y^k, \quad \mu_k := (k + 1)/(k + 2), \]

\[ y^k := x^0 - \lambda_k p^k, \quad p^k := \sum_{i=0}^{k} g^i, \quad g^i \in \partial f(x^i), \]

where $\lambda_k$ was chosen as in (15). We abbreviate this method as (DASG).

We wrote programs for these four methods and took for comparison the above test problem with the same starting point $x^0 = (0, 0, 0, 0, 1)^T$. Also, we took $L = \|g^0\|$. The results are given in Table 2.

From the calculations, we can conclude that (SGMT) and (SGMN) are slow in comparison with (SGM), but (ASG) appeared very slow. At the same time, convergence of (DASG) appeared better than (SGMT) and (SGMN), but was not so rapid even in comparison with (SGM). Therefore, these variants of the methods with averaging subgradients demonstrated rather slow convergence. In fact, their step-size strategies stem from the worst-case analysis of convergence for the whole class of minimization problems. At the same time, any nonlinear function may behave in a different manner on different sets containing iteration points. Hence, an iterative solution method should utilize some adaptive parameter strategies for attaining better convergence properties.

Acknowledgements  The results of this work were obtained within the state assignment of the Ministry of Science and Education of Russia, Project No. 1.460.2016/1.4. This work was supported by Russian Foundation for Basic Research, Project No. 19-01-00431.

References

1. Shor, N.Z.: Minimization Methods for Non-differentiable Functions. Springer, Berlin (1985)
2. Polyak, B.T.: Introduction to Optimization. Optimization Software, New York (1987)
3. Mäkela, M.M., Neittaanmäki, P.: Nonsmooth Optimization. World Scientific, Singapore (1992)
4. Kiwiel, K.C.: Methods of Descent for Nondifferentiable Optimization. Springer, Berlin (1985)
5. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms. Springer, Berlin (1993)

6. Konnov, I.V.: Nonlinear Optimization and Variational Inequalities. Kazan University Press, Kazan (2013). [in Russian]

7. Wolfe, P.: A method of conjugate subgradients for minimizing nondifferentiable functions. In: Balinski, M.L., Wolfe, P. (eds.) Nondifferentiable Optimization. Mathematical Programming Study 3, 145–173 (1975)

8. Dem’yyanov, V.F., Vasil’yev, L.V.: Nondifferentiable Optimization. Optimization Software, New York (1985)

9. Chepurnoi, N.D.: Relaxation method of minimization of convex functions. Dokl. Akad. Nauk Ukrainy. Ser. A. 3, 68–69 (1982). [in Russian]

10. Konnov, I.V.: A subgradient method of successive relaxation for solving optimization problems. Preprint VINITI No. 531-83, Faculty of Computational Mathematics and Cybernetics, Kazan University, Kazan, 14 pp. (1982). [in Russian]

11. Kiwiel, K.C.: An aggregate subgradient method for nonsmooth convex minimization. Math. Program. 27, 320–341 (1983)

12. Konnov, I.V.: A method of the conjugate subgradient type for minimization of functionals. Issled. Prikl. Matem. 12, 59–62 (1984); Engl. transl.: J. Soviet Math. 45, 1026–1029 (1989)

13. Mikhalevich, V.S., Gupal, A.M., Norkin, V.I.: Methods of Nonconvex Optimization. Nauka, Moscow (1987). [in Russian]

14. Konnov, I.V.: Combined Relaxation Methods for Variational Inequalities. Springer, Berlin (2001)

15. Gupal, A.M.: Methods of minimizing functions that satisfy a Lipschitz condition, averaging the directions of descent. Cybernetics 14, 695–698 (1978)

16. Chepurnoi, N.D.: A method of average quasigradients with stepwise step control for minimization of weakly convex functions. Kibernetika 6, 131–132 (1981). [in Russian]

17. Konnov, I.V.: Conditional gradient method without line-search. Russ. Mathem. (Izv. VUZ.) 62, 82–85 (2018)

18. Konnov, I.V.: A simple adaptive step-size choice for iterative optimization methods. Adv. Model. Optim. 20, 353–369 (2018)

19. Konnov, I.V.: Simplified versions of the conditional gradient method. Optimization 67, 2275–2290 (2018)

20. Lemaréchal, C.: An extension of Davidon methods to nondifferentiable problems. In: Balinski, M.L., Wolfe, P. (eds.) Nondifferentiable Optimization. Mathematical Programming Study 3, pp. 95–109 (1975)

21. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)

22. Pshenichnyi, B.N., Danilin, Y.M.: Numerical Methods in Extremal Problems. MIR, Moscow (1978)

23. Shor, N.Z., Shabashova, L.P.: Solution of minimax problems by the method of generalized gradient descent with dilatation of the space. Cybernetics 8, 88–94 (1972)

24. Nesterov, Y.: Primal–dual subgradient methods for convex problems. Math. Program. 120, 261–283 (2009)

25. Nesterov, Y., Shikhman, V.: Quasi-monotone subgradient methods for nonsmooth convex minimization. J. Optim. Theory Appl. 165, 917–940 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.