Journal of Optimization Theory and Applications (2019) 183:1077–1098
https://doi.org/10.1007/s10957-019-01585-w

# Adaptive Conditional Gradient Method

**Z. R. Gabidullina**[1]

Received: 25 May 2018 / Accepted: 31 August 2019 / Published online: 27 September 2019


**Abstract**
We present a novel fully adaptive conditional gradient method with the step length regulation for solving pseudo-convex constrained optimization problems. We propose some deterministic rules of the step length regulation in a normalized direction. These rules guarantee to find the step length by utilizing the finite procedures and provide the strict relaxation of the objective function at each iteration. We prove that the sequence of the function values for the iterates generated by the algorithm converges globally to the objective function optimal value with sublinear rate.

**Keywords** Optimization problems · Pseudo-convex function · Adaptation · Descent direction · Normalization · Step length · Regulation · Rate of convergence

**Mathematics Subject Classification** 90C30 · 65K05

## 1 Introduction

As is widely known, the development of the first variant of the conditional gradient method (or, briefly, CGM) was pioneered by Frank and Wolfe for the case of quadratic programming problems. For mathematical programming problems in the other settings, a broad range of different variants of CGM was explored by many researchers over the years. A helpful systematic survey of the existing literature related to the various schemes of CGM is contained, for instance, in [1] (see also the references therein). For an up-to-date survey of the subject, see also [2–4].

Rather than detailing all the different versions of CGM, that researchers have been developing over the years in the search for the right ideas, we seek to occupy our attention only in some related work.

Communicated by Alexanre Cabot.

✉ Z. R. Gabidullina
zgabid@mail.ru; Zulfiya.Gabidullina@kpfu.ru

[1] Department of Data Analysis and Operations Research, Kazan Federal University, 18, Kremljovskaja Street, Kazan, Russia 420008

In [5], there is provided Frank–Wolfe (conditional gradients) method with a convergence analysis allowing one to approach a primal-dual solution of the convex optimization problem with composite objective functions. In [2], a randomized block-coordinate variant of the classic Frank–Wolfe algorithm (or, briefly, FWA) was presented for the specific optimization problem. Namely, there was considered the problem of minimizing the convex quadratic function under block-separable constraints. In applications to the dual structural support vector machine (SVM) objective, this algorithm provides $O(1/\vartheta)$ convergence rate. The parameter $\vartheta$ is used at each iteration for monitoring the convergence by evaluating the current duality gap as a certificate for the current approximation quality. More precisely, the proposed algorithm, after $O(1/\vartheta)$ many iterations, guarantees an $\vartheta$-approximate solution to the structural SVM dual problem when the duality gap is less or equal to $\vartheta$. Exact minimizers of the linear subproblems were used for determining the descent directions. Since the objective function of the structural SVM dual is simply a quadratic function, the step size for any given candidate search point can be calculated analytically by an explicit formula with further clipping to [0, 1] or optimized by a line search.

In [3], there is considered a method for lazifying the vanilla Frank–Wolfe algorithm as well as the conditional gradient methods. For the descent direction search, they used a weak separation oracle instead of a linear optimization oracle. This allowed them to reapply feasible solutions from the preceding oracle calls, so that in many events there might be skipped the oracle call. In [3], for a lazification of the different variants of CGM, there is required to be fulfilled many supplementary conditions such as: (1) the objective function is, for instance, smooth convex function with curvature C (or $\beta$-smooth and $S$-strongly convex function), (2) the step size is selected with the help of the curvature constant and/or the other hard-to-estimate parameters like $\beta$, $S$, and so forth, (3) in some versions of CGM, the feasible region is only a polytope (moreover, it is given in the special form). For the sake of fairness, it should be noted that the authors also provided a parameter-free variant of the lazy Frank–Wolfe algorithm. But in this case, the parameters are adaptively adjusted using the exact (computationally expensive) line search for the step-size selection.

For any convex and compact feasible domain, there was proved in [4] that FWA gets an approximate stationary point at the rate of $O(1/\sqrt{k})$ on potentially non-convex objectives with a Lipschitz continuous gradient. The adaptive step size is calculated by means of the finite curvature constant $C_{\mathrm{f}}$. The assumption of the bounded curvature corresponds closely to a Lipschitz assumption on the gradient of the function. By the way, we note that $C_{\mathrm{f}}$ is related to the hard-to-estimate parameters. It thus takes at most $O(1/\vartheta^2)$ iterations to find an approximate stationary point with the Frank–Wolfe gap smaller than $\vartheta$. A discussion of the related work in [4] contains the extensive representation of the main results as concerned with Frank–Wolfe methods for non-convex settings (see also the references therein). For the general setting, there was shown in [6] that any limit point of the sequence of iterates for the standard FWA converges to a stationary point (though without the presentation of convergence rates). The proof of convergence only requires the objective function to be smooth. For non-convex objectives, there is studied in [7] a Frank–Wolfe-type algorithm for which the convergence rate is slower than $O(1/\sqrt{k})$.

The main contributions in this paper are as follows. We present a novel fully adaptive conditional gradient method (or, briefly, ACGM) with the step length regulation for solving pseudo-convex constrained optimization problems. This relaxation algorithm allows one to generate the sequence of iterates $\{x_k\}, k = 0, 1, \ldots$ such that the sequence of its function values $\{f(x_k)\}, k = 0, 1, \ldots$ converges globally to the objective function optimal value with the following rate $O(1/k)$ which is usually called the sublinear one. To the extent of our knowledge, a convergence rate of $O(1/k)$ is now known to be the best for CGM in a non-convex objectives context.

It should be pointed out that the sublinear convergence rate takes place under the following assumptions regarding the problem: 1) a feasible region is any convex and closed set; 2) an objective function is required to be satisfied with the so-called Condition A introduced in [8]. Let us note that this condition will be defined explicitly in Sect. 2 (see Definition 2.2). Here, we focus attention on the fact that, for the implementation of ACGM, there is not required any priori information relatively the auxiliary constant defined in Condition A. With respect to this fact, the presented version of ACGM compares favorably with the other versions of CGM discussed above. The fulfillment of Condition A allows us further to adaptively regulate the step length without using any complicated line search techniques. Namely, there is no need to utilize, for instance, the limited minimization rule required for the one-dimensional exact minimization of the objective function on the line segment [0, 1] in the chosen direction of descent. We propose the two deterministic rules of the step length adjustment. They guarantee to find the step size by making use for this purpose the finite procedures of diminishing an original value of a specific parameter. The latter is a user-chosen parameter which is decreased until a moment when the condition applied for the step length regulation becomes fulfilled. We note that the step length calculating rules provide the strict relaxation of the objective function at each iteration. Notice that the concept of the objective function relaxation helps to interpret the relaxation properties of optimization methods. Due to this interpretation, one can evaluate the speed of the objective function value decreasing at each iteration. The case of this value being decreased on the positive magnitude corresponds to the concept of the strict relaxation which is determined by the relation: $f(x_k) > f(x_{k+1}), k = 0, 1, \ldots$

It should be noted that the fully adaptive character of the introduced variant of CGM is determined namely by the combination of simultaneous controlling the adaptation of an $\varepsilon$-normalization parameter of the descent direction as well as the iteration step-size regulation in tandem. We justify rigorously the finiteness of all the procedures for both the step length regulation and the adaptation of the $\varepsilon$-normalization parameter.

Let us note that the ACGM relates to the so-called lazy type of methods due to its lazification by replacing the exact solution of the linear optimization subproblem for finding the descent direction to inexact one. This means that the above-mentioned auxiliary subproblem should be solved with some prescribed by user accuracy. As a result, this strategy leads to considerable reduction in the computational costs for solving the descent direction problem. In the case when the set of the feasible solutions is given by the linear constraints, CGM is essentially popular due to its simplicity of only requiring to solve a linear programming subproblem for determining the descent direction. In ACGM, this subprogram solving is still more simplified, since it allows one to limit oneself to several iterations (using linear optimization methods) toward

the direction of minimizing a linear function. Besides, the feasible domain can be potentially unbounded. In this event, despite the solvability of the original problem, the linear optimization subproblem can have no solutions. Fortunately, the special setting of the descent direction problem allows one to construct the descent direction for this case, too.

Thanks to all the mentioned properties of ACGM, it seems that the results of the paper can be potentially applicable in both theoretical and practical aspects in numerous applied areas [in pseudo-convex programming, sets separation, and many others (in particular, data classification and identification techniques)]. Besides, it is well known from the optimization literature that Frank–Wolfe-type methods are very useful for solving the problem of projecting the origin of the Euclidean space onto a convex polyhedron (see, for instance, [9]). Therefore, it will be especially interesting to study in the future the application of ACGM to solve the problems of projecting onto the convex polyhedron and computing the distance between the convex polyhedra learned, for example, in [10–12].

The rest of this paper is organized as follows. In Sect. 2, we provide some preliminaries for our convergence rate analysis of ACGM. In Sect. 3, we formulate ACGM and justify its convergence rate. In Sect. 4, we present the finite algorithms for the refinement of an $\varepsilon$-normalization parameter of the descent direction. Section 5 contains some results of our experimental study of ACGM. In Sect. 6, there are drawn some conclusions.

## 2 Definitions and Preliminaries

Our goal is to study the following problem:

$$\min_{x \in D} f(x), \tag{1}$$

where $f(x)$ is a continuously differentiable pseudo-convex function satisfying the so-called Condition $A$ (introduced in [8]) on a convex and closed subset $D$ of Euclidean space $\mathbb{R}^n$. For solving this problem, we present a new efficient algorithm, which has the estimates of the rate of its convergence and allows one to adaptively control both the parameter of an $\varepsilon$-normalization of a descent direction and the step length.

We start with some notations: $g(x)$ is the gradient of the function $f(x)$ at the point $x$, $x_0$ stands for the starting point of the iterative consequence constructed by minimizing the objective function. Let $\|\cdot\|$ stand for the Euclidean norm of a vector in $\mathbb{R}^n$, $f^* := \min\limits_{x \in D} f(x)$, $D^* := \{x \in D : f(x) = f^*\}$, $\mathbb{N} = \{0, 1, \ldots\}$, and $p_k^*$ corresponds to a projection of the iterative point $x_k$ onto the set $D^*$, $k \in \mathbb{N}$.

To the best of our knowledge, the class of continuously differentiable pseudo-convex functions was pioneered by Mangasarian in [13]. It is well known that the above-mentioned class generalizes the family of all smooth convex functions.

**Definition 2.1** (*pseudo-convexity*) A continuously differentiable function $f(x)$ given on an open and convex set $G$ from $\mathbb{R}^n$ is called pseudo-convex, if for all $x, y \in G$ there holds the following implication:

$$\langle g(x), y - x\rangle \geq 0 \Rightarrow f(x) \leq f(y),$$

or equivalently,

$$f(y) < f(x) \Rightarrow \langle g(x), y - x\rangle < 0.$$

For the class of pseudo-convex functions, the necessary and sufficient conditions of optimality are established in the following theorem:

**Theorem 2.1** (Basic first-order conditions for optimality) ([13], p. 282) *For the point $x^* \in G$ to furnish the minimum of $f(x)$ over $G$, it is necessary and sufficient for all $x \in G$ to hold*

$$\langle g(x^*), x - x^*\rangle \geq 0.$$

**Definition 2.2** (*Condition A*) We say that a continuous function $f(x)$ satisfies Condition $A$ on the convex set $D \subseteq \mathbb{R}^n$ if there exists a nonnegative symmetric function $\tau(x, y)$ and $\mu > 0$ such that

$$f(\alpha x + (1-\alpha)y) \geq \alpha f(x) + (1-\alpha) f(y) - \alpha(1-\alpha)\mu\tau(x, y),$$
$$\forall x, y \in D,\ \alpha \in [0, 1].$$

For $x, y \in D \subseteq \mathbb{R}^n$, some function $\tau(x, y)$ is called symmetric if $\tau(x, y) = \tau(y, x)$, $\tau(x, x) = 0$. Condition $A$ describes a sufficiently broad class of functions $A(\mu, \tau(x, y))$ . It was shown in [8,14,15] that the class $A(\mu, \|x - y\|^2)$, in particular, is wider than $C^{1,1}(D)$—the well-known class of functions whose gradients satisfy the Lipschitz condition on the convex set $D \subseteq \mathbb{R}^n$. By the way, we note that Lipschitzian properties of gradients for this class of functions have been sought as the favorable assumptions in the justification of the theoretical estimates of the convergence rate for the various modern differentiable optimization algorithms. In [15], there is given a variety of examples of functions that satisfy Condition A. For functions from $A(\mu, \tau(x, y))$, we also investigated their principle properties and criteria which allow one to categorize some function as belonging to the treated class or not. In particular, it was proved in [15] that, for a continuously differentiable function satisfying Condition A on a convex set D, the following extremely important differential inequality holds:

$$f(x) - f(y) \geq \langle g(x), x - y\rangle - \mu\tau(x, y). \tag{2}$$

**Theorem 2.2** (Relation between two classes of functions) [8] *If $D$ is convex subset of $\mathbb{R}^n$, $f(x) \in C^{1,1}(D)$, then $f(x)$ satisfies Condition A on $D$ with coefficient $\mu = L/2$ and function $\tau(x, y) = \|x - y\|^2$, where $L$ is a Lipschitz constant for the gradient of $f(x)$.*

***Proof*** In optimization theory, it is certainly well known that the function $f(x)$ satisfies the following differential inequality:

$$f(x) - f(y) \geq \langle g(x), x - y\rangle - \frac{L}{2}\|x - y\|^2,\ \forall x, y \in D. \tag{3}$$

Let $x_\alpha = \alpha x + (1-\alpha)y, \forall \alpha \in [0,1]\, \forall x, y \in D$; from (3), we then have the following inequalities:

$$f(x_\alpha) - f(x) \geq \langle g(x_\alpha), x_\alpha - x\rangle - \frac{L}{2}\|x_\alpha - x\|^2. \tag{4}$$

$$f(x_\alpha) - f(y) \geq \langle g(x_\alpha), x_\alpha - y\rangle - \frac{L}{2}\|x_\alpha - y\|^2. \tag{5}$$

Summing these inequalities, having previously multiplied (4) by $\alpha$ and (5) by $(1-\alpha)$, taking into account the symmetricity of $\tau(x, y) = \|x - y\|^2$, we get

$$\begin{aligned} f(x_\alpha) \geq\ & \alpha f(x) + (1-\alpha) f(y) + \langle g(x_\alpha), x_\alpha - (\alpha x + (1-\alpha)y)\rangle \\ & - \alpha(1-\alpha)\frac{L}{2}\|x - y\|^2 = \alpha f(x) + (1-\alpha) f(y) - \alpha(1-\alpha)\frac{L}{2}\|x - y\|^2. \end{aligned}$$

□

**Definition 2.3** ($\varepsilon$-*normalized descent direction*) For functions from the class $A(\mu, \|x - y\|^v)$, $v \geq 2$, a vector $s \neq \mathbf{0}$ is called an $\varepsilon$-normalized descent direction ($\varepsilon > 0$) of the function $f$ at the point $x \in D$ if the following inequality holds:

$$\langle g(x), s\rangle + \varepsilon\|s\|^v \leq 0.$$

**Lemma 2.1** ($\varepsilon$-normalization) *If some descent direction $s$ is not $\varepsilon$-normalized, then the vector constructed in such a way that $\bar{s} = \dfrac{ts}{\varepsilon\|s\|^v}$ is $\varepsilon$-normalized under the condition* $0 < t \leq |\langle g(x), s\rangle|$.

***Proof*** By construction, we have the following relation:

$$\begin{aligned} \langle g(x), \bar{s}\rangle + \varepsilon\|\bar{s}\|^v &= \frac{t}{\varepsilon\|s\|^v}\langle g(x), s\rangle + \frac{t^v}{\varepsilon^{v-1}\|s\|^{v(v-1)}} \\ &= \frac{t}{\varepsilon\|s\|^v}\left[\langle g(x), s\rangle + t\left[\frac{t}{\varepsilon\|s\|^v}\right]^{v-2}\right] \leq 0, \end{aligned}$$

because $\left[\dfrac{t}{\varepsilon\|s\|^v}\right]^{v-2} \leq 1$. □

Under the condition $v = 2$, we fix some point $x \in \mathbb{R}^n$; then, it is not hard to see that all the points $z \in \mathbb{R}^n$, for which the vectors $z - x$ are $\varepsilon$-normalized directions of descent at the point $x$, belong to the $n$-dimensional ball of radius $R = \|g(x)\|/2\varepsilon$ with center at the point $u = x - g(x)/2\varepsilon$.

Let

$$\zeta = \begin{cases} (\varepsilon \cdot \mu^{-1})^{1/(v-1)}, & \text{if } \varepsilon < \mu, \\ 1, & \text{if } \varepsilon \geq \mu. \end{cases}$$

For the $\varepsilon$-normalized descent directions, we further present their very useful new properties providing a strict relaxation of the objective function.

**Lemma 2.2** (Basic properties of $\varepsilon$-normalized descent directions) *Let $s$ be some $\varepsilon$-normalized descent direction for the function $f$ at the point $x$ where $v \geq 2$, $f(x) \in A(\mu, \|x - y\|^v)$, then for all $\beta \in ]0, 1[$ there exists a constant $\hat{\lambda} = \hat{\lambda}(\beta) > 0$ ($\hat{\lambda} = (1-\beta)^{1/(v-1)}\zeta$) such that for all $\lambda \in \left]0, \hat{\lambda}\right]$ it holds*

$$f(x) - f(x + \lambda s) \geq -\lambda\beta \cdot \langle g(x), s\rangle, \tag{6}$$

$$f(x) - f(x + \lambda s) \geq \lambda\beta \cdot \varepsilon\|s\|^v. \tag{7}$$

***Proof*** Using (2), we get

$$\begin{aligned} f(x) - f(x + \lambda s) &\geq -\lambda \cdot \langle g(x), s\rangle - \mu\lambda^v\|s\|^v \\ &= \lambda\beta\omega - \lambda\left(\langle g(x), s\rangle + \mu\lambda^{v-1}\|s\|^v + \beta\omega\right). \end{aligned} \tag{8}$$

We estimate further the function $\alpha(\omega) = \langle g(x), s\rangle + \mu\lambda^{v-1}\|s\|^v + \beta\omega$ for the two cases: $\omega = -\langle g(x), s\rangle$ and $\omega = \varepsilon\|s\|^v$. In the first case, by definition of the $\varepsilon$-normalized descent direction $s$, it holds

$$\begin{aligned} \alpha(\omega) &= (1-\beta) \cdot \langle g(x), s\rangle + \mu\lambda^{v-1}\|s\|^v \leq -(1-\beta)\varepsilon\|s\|^v + \mu\lambda^{v-1}\|s\|^v \\ &= \mu\|s\|^v \cdot \left((\beta - 1)\varepsilon/\mu + \lambda^{v-1}\right). \end{aligned}$$

In the second event, we have

$$\alpha(\omega) = \langle g(x), s\rangle + \mu\lambda^{v-1}\|s\|^v + \beta\varepsilon\|s\|^v \leq \mu\|s\|^v \cdot \left((\beta - 1)\varepsilon/\mu + \lambda^{v-1}\right).$$

Thus, there is fulfilled the following implication:

$$0 \leq \lambda \leq (1-\beta)^{1/v-1}\zeta \Rightarrow \alpha(\omega) \leq 0, \text{ where } \omega = -\langle g(x), s\rangle \text{ or } \omega = \varepsilon\|s\|^v.$$

This implies that the inequalities (6)–(7) then follows from (8). □

The inequalities (6)–(7) are needed for justifying the convergence of the adaptive algorithm which will be presented below. In particular, the above-mentioned expressions imply that

$$f(x + \lambda s) < f(x) \text{ for all } 0 < \lambda \leq (1-\beta)^{1/v-1}\zeta, \beta \in (0, 1).$$

According to Lemma 2.2, we describe the strategies that can be utilized for calculating the step size satisfying (6)–(7). Let $s$ be some $\varepsilon$-normalized direction of descent for $f$ at the point $x$. Besides, let the following conditions be fulfilled: $\beta \in ]0, 1[$, $\eta = (1-\beta)^{1/v-1}$, $\hat{i} = 1$, $J(\hat{i}) = \{\hat{i}, \hat{i}+1, \hat{i}+2, \ldots\}$. We further determine $i^*$ as the least index $i \in J(\hat{i})$ for which there holds the following condition:

$$f(x) - f(x + \eta^i s) \geq -\eta^i \beta \cdot \langle g(x), s \rangle, \tag{9}$$

or the more weak condition:

$$f(x) - f(x + \eta^i s) \geq \eta^i \beta \cdot \varepsilon \|s\|^v. \tag{10}$$

Next, we set $\lambda = \eta^{i^*}$. In what follows, we mean that there is used Rule 1 (or Rule 2) when we follow the first (or the second) of the above described strategies for determining the value of the step length. The step size calculated in accordance with these rules satisfies (6) or (7), respectively.

We further investigate the case when $s$ is the $\varepsilon$-normalized descent direction, but it is not $\mu$-normalized. (This situation is possible only for $\varepsilon < \mu$.) Under the assumption that $0 < \varepsilon < \mu$, for the case of finding $\lambda$ according to Rule 1 or Rule 2, we prove that the step size is bounded from below. This obviously implies that the represented procedures of diminishing the step length are finite.

**Lemma 2.3** (Finite lower bound for the step length) *If*

*(b)* $f(x) \in A(\mu, \|x - y\|^v)$, $v \geq 2$,
*(c)* $0 < \varepsilon < \mu$, $\beta \in ]0, 1[$,
*(d)* $s$- *is an* $\varepsilon$*-normalized descent direction of the function* $f$ *at the point* $x$*, but it is not* $\mu$*-normalized,*
*(e)* $i^*$ *is the smallest index* $i = 1, 2, \ldots,$ *for which there is fulfilled the condition of Rule 1 or Rule 2,* $\lambda = \eta^{i^*}$*;*

*Then, the following estimate holds:*

$$\lambda > \left(\varepsilon \mu^{-1} \cdot (1-\beta)^2\right)^{1/(v-1)} > 0.$$

***Proof*** If $i^* = 1$, then in this event there is nothing to prove since it holds $\lambda = (1-\beta)^{1/(v-1)}$. Now, let $i^* \neq 1$. This case corresponds to the fact that the condition (9) [or (10)] (applied in the rule of calculating the step length) was not fulfilled for $\eta^{i^*-1}$. Due to Lemma 2.2, we then obtain

$$\eta^{i^*-1} > \left((1-\beta)\varepsilon\mu^{-1}\right)^{1/(v-1)}.$$

This yields $\lambda = \eta^{i^*} > \left((1-\beta)^2 \varepsilon \mu^{-1}\right)^{1/(v-1)} > 0$. □

***Remark 2.1*** (Finite lower bound for the constant $\mu$) From Lemma 2.3, under its conditions, there comes immediately the following estimate:

$$\mu > \varepsilon \cdot (1-\beta)^2 \lambda^{1-v}. \tag{11}$$

Later, the estimate (11) will be applied in the algorithm for adapting the parameter for the $\varepsilon$-normalization of the descent direction.

## 3 Adaptive Algorithm and Its Convergence

This section is devoted to the principles of choosing the $\varepsilon$-normalization parameter for the descent direction. The algorithm convergence for the fixed parameter $\varepsilon$ (in the case of an arbitrary ratio of the parameter $\varepsilon$ and the value of $\mu$ in Condition A) follows from the convergence of the adaptive variant of CGM. We note that generally speaking the constant $\mu$ is unknown beforehand. Consequently, in practice, the choice of the $\varepsilon$ values close to the $\mu$ value is decisive for the algorithm convergence. If one selects the too small parameter $\varepsilon$, then, in accordance with Rule 1 and Rule 2, there can be obtained significant diminishing of the step length. In the choice of the unjustifiably large value of $\varepsilon$, the convergence of the adaptive algorithm can be slowed down. Therefore, it is expedient to estimate the parameter $\varepsilon$ in the process of working the algorithm. The inequalities (9)–(11) allow us to make an adjustment to the value $\varepsilon$, increasing it if the previous choice was unsuccessful. Now, we describe further details of a procedure for pointwise adaptation of the parameter $\varepsilon$ during the iterative process of the algorithm.

For the $k$th iteration of the adaptive algorithm, let $\varepsilon_k > 0$ be the value of a parameter for an $\varepsilon$-normalization of descent direction. Let $s_k$ be an $\varepsilon$-normalized descent direction of the function $f$ at $x_k$; the iterative step size $\lambda_k$ is selected according to one of the Rules 1–2.

If $i_k$ is the least index $i \in J(\hat{i})$, for which there holds the condition of choosing the iteration step size for $x = x_k$, $\varepsilon = \varepsilon_k$, $s = s_k$. Due to Lemma 2.2, if $i_k = \hat{i}$, then for going to the next—$(k+1)$th—iteration it is expedient to leave the unchanged value of the normalization parameter, i.e., to put $\varepsilon_{k+1} = \varepsilon_k$. During the process of dropping the step size, let the checked condition (9) [or condition (10)] be fulfilled for $i_k > \hat{i}$. Then, in accordance with (11), there should be increased current value of the parameter for the $\varepsilon$-normalization of the descent direction, for instance, as follows: $\varepsilon_{k+1} = \varepsilon_k \cdot \zeta_k$. Regardless of what rule is selected for calculating the step length, here one has

$$\zeta_k = (1-\beta)^{1-i_k}. \tag{12}$$

From the fact that $\mu < +\infty$, it follows that after the finite number of increases, the value of the parameter $\varepsilon$ can exceed $\mu$ and cease to vary. Let $j > 0$ be the number of iterations, on which it holds $\varepsilon_j \geq \mu$. We then have $\varepsilon_k \geq \varepsilon_j \geq \mu$, $\forall k \geq j$. In this case, from some iteration $j \geq 0$ the adaptive algorithm begins to work with the fixed constant for the $\varepsilon$-normalization of a descent direction. We underline that beginning from the $j$th iteration, the step length becomes constant: $\lambda_k = \eta, \forall k \geq j$. At that time,

for calculating the iterative step size, we need only one calculation of the objective function value at the point $x_k + \eta s_k$ (for checking the fulfillment of the condition used for choosing the step size).

**Algorithm**

Step 0 Initialization. Select $x_0 \in D$, $\beta \in ]0, 1[$, $\varepsilon_0 > 0$, $0 < \sigma_0 \leq 1$, $0 < \alpha \leq \alpha_0$. Set the iteration counter $k$ to 0.

Step 1 Under the conditions $0 < \sigma \leq \sigma_k \leq 1$, $0 < \alpha \leq \alpha_k$, choose a point $y_k$, $k = 0, 1, \ldots$ in such a way that it holds

$$\langle g(x_k), y_k - x_k\rangle \leq \max\{\sigma_k \min_{x\in D}\langle g(x_k), x - x_k\rangle, -\alpha_k\}. \tag{13}$$

If $\langle g(x_k), y_k - x_k\rangle = 0$, then terminate the algorithm implementation [since $x_k$ is a solution of the problem (1)]. Otherwise, set

$$s_k = \begin{cases} y_k - x_k, & \text{if} \langle g(x_k), y_k - x_k\rangle + \varepsilon_k \|y_k - x_k\|^v \leq 0, \\ \dfrac{t_k(y_k - x_k)}{\varepsilon_k \|y_k - x_k\|^v}, & \text{else.} \end{cases}$$

Here $t_k = |\langle g(x_k), y_k - x_k\rangle|$.

Step 2 Let $i_k$ be the least index $i \in J(\hat{i})$ for which there holds the condition from Rule 1 or Rule 2 when $x = x_k$, $s = s_k$, $\varepsilon = \varepsilon_k$. Set $\lambda_k = \eta^{i_k}$.

Step 3 Compute the next iterate $x_{k+1} = x_k + \lambda_k s_k$.

Step 4 Update $\varepsilon_{k+1} = \zeta_k \varepsilon_k$. Set $k = k + 1$ and go to Step 1.

Clearly, we apply the same rule for selecting the step length at each iteration point.

***Remark 3.1*** (Characterization of the descent direction) Let $\bar{s}_k = \|y_k - x_k\|$. The vector $s_k$ constructed by the algorithm is $\varepsilon$-normalized. This is evident when $s_k = \bar{s}_k$. For the case where $s_k = \dfrac{t_k \bar{s}_k}{\varepsilon_k \|\bar{s}_k\|^v}$, Lemma 2.1 justifies that $s_k$ is $\varepsilon$-normalized, too.

Moreover, it obviously holds $\|s_k\| \leq \|\bar{s}_k\|$. Indeed,

$$\|s_k\| = \begin{cases} \|\bar{s}_k\|, & \text{if} \langle g(x_k), \bar{s}_k\rangle + \varepsilon_k \|\bar{s}_k\|^v \leq 0, \\ \dfrac{t_k}{\varepsilon_k \|\bar{s}_k\|^{v-1}} < \|\bar{s}_k\|, & \text{otherwise,} \end{cases}$$

since $\dfrac{t_k}{\varepsilon_k \|\bar{s}_k\|^v} = \dfrac{-\langle g(x_k), \bar{s}_k\rangle}{\varepsilon_k \|\bar{s}_k\|^v} < 1$. Therefore, the point $x_{k+1}$, which is obtained by moving toward the search direction $s_k$ using some step size $\lambda_k \in ]0, 1[$, belongs to the feasible set.

To discuss the rate of convergence for ACGM in the pseudo-convex setting, we need to determine a measure of optimality for our iterates. The minimum value of the objective function is usually not known beforehand, so it is important to formulate the stopping criterion directly in terms of the optimum.

**Theorem 3.1** *(Constructive measure of optimality for ACGM) Let $f(x)$ be a continuously differentiable pseudo-convex function on a convex set $D \subseteq \mathbb{R}^n$. Then, for the function $f(x)$ to attain its minimum value on $D$ at the point $x_k \in D$, it is necessary and sufficient to hold*

$$\langle g(x_k), y_k - x_k \rangle = 0. \tag{14}$$

***Proof*** Necessity. Suppose that $f(x)$ achieves its minimum over $D$ at $x_k$. According to Theorem 2.1, we have $\langle g(x_k), x - x_k \rangle \geq 0$, $\forall x \in D$. Therefore, $\langle g(x_k), y_k - x_k \rangle \geq 0$, since $y_k \in D$. Then, in (13) there does not take place the following relation: $\sigma_k \min\limits_{x \in D} \langle g(x_k), x - x_k \rangle \leq -\alpha_k$. Consequently, it is fulfilled $\sigma_k \min\limits_{x \in D} \langle g(x_k), x - x_k \rangle > -\alpha_k$. Then, for all $x \in D$ it holds

$$0 \leq \langle g(x_k), y_k - x_k \rangle \leq \sigma_k \min_{x \in D} \langle g(x_k), x - x_k \rangle \leq \langle g(x_k), x - x_k \rangle.$$

Therefore, for $x = x_k$ we obtain $0 \leq \langle g(x_k), y_k - x_k \rangle \leq 0$. From this, it obviously follows that $\langle g(x_k), y_k - x_k \rangle = 0$. That is what we want to prove.

Sufficiency. We assume now that (14) holds. By force of choosing the descent direction by (13), it is not hard to see that, under the condition (14), the situation where $\max\{\sigma_k \min\limits_{x \in D} \langle g(x_k), x - x_k \rangle, -\alpha_k\} = -\alpha_k$ can never be, since $-\alpha_k \leq -\alpha < 0$ and $y_k \in D$. Consequently,

$$0 = \langle g(x_k), y_k - x_k \rangle \leq \sigma_k \min_{x \in D} \langle g(x_k), x - x_k \rangle \leq \langle g(x_k), x - x_k \rangle, \forall x \in D,$$

i.e., it holds $\langle g(x_k), x - x_k \rangle \geq 0$, $\forall x \in D$. Due to Theorem 2.1, we then conclude that $f(x)$ furnishes its minimum at the point $x_k$. □

Let $\{x_k\}$ be the sequence constructed by the algorithm. Furthermore, let $x_k \notin D^*$, $\forall k = 0, 1, \ldots$ For the purpose of exploring the convergence of numerical methods in the case of pseudo-convex functions, there is usually defined in the literature on optimization an auxiliary numeric sequence $\{\theta_k\}$ in the following way:

$$\theta_k > 0, 0 < \theta_k \cdot (f(x_k) - f(x^*)) \leq \langle g(x_k), x_k - x^* \rangle, x^* \in D^*, k \in \mathbb{N}. \tag{15}$$

From the definition of pseudo-convexity, it follows that for pseudo-convex functions such values $\theta_k$ must exist. In particular, if $f(x)$ is a smooth convex function, then $\theta_k = 1, k = 0, 1, \ldots$ The properties of elements of the sequence $\{\theta_k\}$ were investigated, for instance, in [8,15].

Before considering the theorem on the convergence of the sequence $\{x_k\}$ generated by the algorithm to a solution of problem (1), we remind the following familiar fact related to convergence of some numeric sequence.

**Lemma 3.1** (Sublinear rate of convergence for numeric sequences) ([16], p. 102) *If a numeric sequence $\{a_k\}$ is such that*

$$a_k \geq 0, a_k - a_{k+1} \geq q \cdot a_k^2, k = 1, 2, \ldots,$$

*where $q$ is some positive constant, then the following estimate holds:*

$$a_k \sim O(1/k),$$

*i.e., there will be found a constant*

$$q_1 > 0 \, \textit{such that} \, 0 \le a_k \le q_1 \cdot k^{-1}, k = 1, 2, \ldots$$

For the proof of the convergence theorem, there is a need to consider the following auxiliary lemma.

**Lemma 3.2** (Boundedness of adapted values of the normalization parameter) *If*

*(b1)* $f(x) \in A(\mu, \|x - y\|^v)$, $v \ge 2$,
*(c1)* $s_k$ *is the* $\varepsilon_k$*-normalized descent direction,*
*(d1)* $i_k$ *is the least index* $i \in J(\hat{i})$ *for which there holds one of the conditions* (9) *or* (10) *under the assumptions that* $s = s_k$, $x = x_k$, $\varepsilon = \varepsilon_k$, $\lambda_k = \eta^{i_k}$,
*(e1)* $\{x_k\}$ *is some iterative sequence constructed by the rule:*

$$x_{k+1} = x_k + \lambda_k s_k, \quad , k \in \mathbb{N},$$

*(f1)* $\varepsilon_0 > 0$, $\varepsilon_{k+1} = \varepsilon_k \cdot (1 - \beta)^{1 - i_k}$, $k \in \mathbb{N}$.

*Then, it is fulfilled* $\varepsilon_k \le \bar{\varepsilon}$, $\forall k \in \mathbb{N}$, *where* $\bar{\varepsilon} = \max\left\{\varepsilon_0, \dfrac{\mu}{1 - \beta}\right\} > 0$.

***Proof*** Part 1. First let us consider the case when among $k \in \mathbb{N}$ there exist such indices that it holds $\varepsilon_k \ge \mu$. Let the index $m$ be the smallest of them. According to Lemma 2.2, the condition (9) or (10) is then fulfilled for $i_m = \hat{i}$. Due to the condition (e1) of the lemma, $\varepsilon_{m+1} = \varepsilon_m$. Consequently, $i_{m+1} = \hat{i}$. This means that for all $k \ge m$ one has $\varepsilon_k = \varepsilon_m \ge \mu$, $i_k = \hat{i}$. If $m = 0$, then $\varepsilon_k = \varepsilon_0$, $\forall k \in \mathbb{N}$. Otherwise, by assumption relatively the index $m$: $\varepsilon_{m-1} < \mu$. From Lemma 2.3, it immediately follows that

$$\begin{aligned} \lambda_{m-1} = \eta^{i_{m-1}} &> \left(\varepsilon_{m-1}\mu^{-1}(1 - \beta)^2\right)^{1/(v-1)} \\ &\Rightarrow (1 - \beta)^{i_{m-1}/(v-1)} > \left(\varepsilon_{m-1}\mu^{-1}(1 - \beta)^2\right)^{1/(v-1)} \\ &\Rightarrow (1 - \beta)^{i_{m-1}} > \varepsilon_{m-1}\mu^{-1}(1 - \beta)^2. \end{aligned}$$

From this, taking into account the condition (f1) of the lemma, we obtain

$$\varepsilon_m = \varepsilon_{m-1}(1 - \beta)^{1 - i_{m-1}} < \frac{\mu}{1 - \beta}.$$

Part 2. If for all $k \in L$ we have $\varepsilon_k < \mu$, then there takes place the inequality $\varepsilon_k < \dfrac{\mu}{1 - \beta}$, $\forall k \in \mathbb{N}$ since $\beta \in ]0, 1[$. □

In the following theorem, there is evaluated the expected decrease in the objective function value in the $\varepsilon_k$-normalized descent direction per step size selected according to Rule 1.

**Theorem 3.2** (Estimate of the magnitude of decreasing the objective function value when the step length is chosen according to Rule 1) *If*

*(b2) the conditions (b1), (c1), (e1) and (f1) of Lemma 3.2 are fulfilled,*
*(c2) $i_k$ is the smallest index $i \in J(\hat{i})$ for which there is fulfilled the condition (9) with $x = x_k$, $s = s_k$, $\varepsilon = \varepsilon_k$, $\lambda_k = \eta^{i_k}$, $\eta = (1-\beta)^{1/(v-1)}$, $\beta \in ]0, 1[$.*

*Then, there will be found a constant $\bar{C} > 0$ such that for all $k \in \mathbb{N}$ the following relation holds:*

$$f(x_k) - f(x_{k+1}) \geq -\bar{C} \cdot \langle g(x_k), s_k \rangle \geq -\bar{C} \cdot (\langle g(x_k), s_k \rangle + \varepsilon_k \|s_k\|^v). \tag{16}$$

***Proof*** For the values $\varepsilon_k$, $k \in \mathbb{N}$ and coefficient $\mu$ from Condition A, there can be fulfilled the following estimates:

I. $0 < \varepsilon_k < \mu$, $\forall k \in \mathbb{N}$.
II. Among $k \in \mathbb{N}$, there will be found the indices such that $\varepsilon_k \geq \mu$ (let $m \in \mathbb{N}$ be one of them).

Next, we will successively analyze these enumerated cases.
I. Let $k \in \mathbb{N}$ be such that $s_k$ are $\mu$-normalized descent directions. According to Lemma 2.2, for all $k \in \mathbb{N}$, we then have

$$f(x_k) - f(x_{k+1}) \geq -\beta(1-\beta)^{1/(v-1)} \cdot \langle g(x_k), s_k \rangle. \tag{17}$$

For all $k \in \mathbb{N}$ such that $s_k$ are not $\mu$-normalized descent directions, taking into account the assertion of Lemma 2.3, one obtains

$$f(x_k) - f(x_{k+1}) \geq -\beta\lambda_k \langle g(x_k), s_k \rangle > -C_2 \langle g(x_k), s_k \rangle, \tag{18}$$

where $C_2 = \left((1-\beta)^2 \varepsilon_0/\mu\right)^{1/(v-1)} \beta$, since it holds $\varepsilon_k \geq \varepsilon_0$, $\forall k \in \mathbb{N}$.
II. In the case where $s_k$ are $\mu$-normalized descent directions for all $k < m$, $m \neq 0$, it takes place (17). When the vector $s_k$ is not $\mu$-normalized for $k < m$, $m \neq 0$, the inequality (18) is true (see the proof of case I). For all $k \geq m$, $m \neq 0$, $s_k$ are $\mu$-normalized descent directions. Consequently, for those $k$ there is fulfilled the inequality (17). Let $C_1 = \beta(1-\beta)^{1/(v-1))}$.

Now, we obtain that for either of the two relations between $\varepsilon_k$ and $\mu$ ($k \in \mathbb{N}$), there will be found a constant $\bar{C} = \min\{C_1, C_2\} > 0$ such that for all $k \in \mathbb{N}$ there is fulfilled the inequality (16). □

**Theorem 3.3** (Estimate of the magnitude of decreasing the objective function value when the step length is chosen according to Rule 2) *Let*

*(b3) the conditions (b1), (c1), (e1) and (f1) of Lemma 3.2 be fulfilled,*
*(c3) the values of the iterative step size $\lambda_k$, $\forall k \in \mathbb{N}$ be determined using (10). Then, the assertion of Theorem 3.2 is true.*

***Proof*** In complete analogy with the proof of Theorem 3.2, we explore separately the same possible ratios of parameter values $\varepsilon_k, k \in \mathbb{N}$ and coefficient $\mu$ from Condition A.
I. Let $k \in \mathbb{N}$ be such that $s_k$ is the $\mu$-normalized descent direction. Due to Lemma 2.2, for this $k \in \mathbb{N}$ one has (17). In the case when for all $k \in \mathbb{N}$, $s_k$ are not $\mu$-normalized, the formulas (2) and (10) yield

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq -\lambda_k \cdot \langle g(x_k), s_k \rangle - \mu \lambda_k^v \|s_k\|^v \\ &\geq -\lambda_k \cdot \langle g(x_k), s_k \rangle - \mu \varepsilon_k^{-1} \lambda_k^{v-1} \beta^{-1} \left( f(x_k) - f(x_{k+1}) \right). \end{aligned}$$

From this, according to Lemma 2.3, we obtain:

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq -\lambda_k \cdot \left( 1 + \mu \varepsilon_k^{-1} \lambda_k^{v-1} \beta^{-1} \right)^{-1} \cdot \langle g(x_k), s_k \rangle \\ &> -C_2 \langle g(x_k), s_k \rangle, \end{aligned} \tag{19}$$

where $C_2 = \left( \varepsilon_0 \mu^{-1} (1-\beta)^2 \right)^{1/(v-1)} \left( 1 + \varepsilon_0 \mu^{-1} (1-\beta) \beta^{-1} \right)^{-1}$, since for all $k \in \mathbb{N}$ one has $\varepsilon_k = \varepsilon_0$.
II. For all $k < m$ ($m \neq 0$), such that $s_k$ are $\mu$-normalized descent directions, the inequality (17) is true. When the descent directions $s_k$ are not $\mu$-normalized for $k < m$ ($m \neq 0$), there comes in play (19) (see the proof of case I). In the case of $k \geq m$, the vectors $s_k$ are $\mu$-normalized. In consequence, the inequality (17) for those $k$ is true.

Let $C_1 = \beta(1-\beta)^{1/(v-1))}$. Then, the estimates (17) and (19) give that the assertion of the theorem is true. Namely, for all the indices $k \in \mathbb{N}$ and the positive constant $\bar{C} = \min\{C_1, C_2\}$, the inequality (16) holds. □

**Theorem 3.4** (Sublinear rate of convergence of ACGM) *If*

*(b4)* $f(x)$ *is a continuously differentiable pseudo-convex function on the convex and closed set* $D \subseteq \mathbb{R}^n$ *satisfying Condition A with a function* $\tau(x, y) = \|x - y\|^v$, $v \geq 2$, *and some constant* $\mu$,
*(c4)* *a numeric sequence* $\{\theta_k\}$, *which is defined by* (15), *satisfies the condition:* $\exists \theta > 0$ *such that* $\theta_k \geq \theta$, $\forall k$,
*(d4)* *there exists a constant* $\gamma > 0$ *such that* $\|g(x)\| \leq \gamma < \infty$, $\forall x \in D$,
*(e4)* *the Lebesgue set of the function* $f(x)$ *at the point* $x_0 \in D$, *which is denoted by* $M_D(f, x_0) := \{x \in D : f(x) \leq f(x_0)\}$, *is bounded,*
*(f4)* $\{\alpha_k\}$, $\{\sigma_k\}$ *are such that* $\exists \bar{\eta} > 0 : \|x_k - y_k\| \leq \bar{\eta}$, $\forall k$,
*(g4)* *a step size* $\lambda_k, k \in \mathbb{N}$ *is chosen according to one of the rules (Rule 1 or Rule 2).*

*Then, the sequence* $\{x_k\}$, $k \in \mathbb{N}$ *is weakly convergent, i.e.,*

$$f(x_k) - f^* \sim O(1/k),$$

*or equivalently, there exists a constant* $C > 0$ *such that it holds*

$$f(x_k) - f^* \leq C \cdot k^{-1}.$$

***Proof*** Obviously, $f(p_k^*) = f^*$, $f(x_k) > f(p_k^*)$, $\forall k \in \mathbb{N}$. By definition of pseudo-convex functions, it holds $\langle g(x_k), p_k^* - x_k\rangle < 0$. By virtue of the assertions of Theorems 3.2–3.3, regardless of the choice of any rule for calculating the step length, there will be found a constant $\bar{C} > 0$ such that there is fulfilled the inequality (16) for all $k \in \mathbb{N}$. Select the subset of indices $\mathbb{N}_1 \subset \mathbb{N}$ such that $s_k = y_k - x_k$, $k \in \mathbb{N}_1$. We then have $s_k = \dfrac{t_k(y_k - x_k)}{\varepsilon_k \|y_k - x_k\|^v}$ for all $k \in \mathbb{N}_2 = \mathbb{N}\backslash\mathbb{N}_1$. For all $k \in \mathbb{N}_1$, we obtain the estimate:

$$f(x_k) - f(x_{k+1}) \geq -\bar{C} \cdot \langle g(x_k), y_k - x_k\rangle$$
$$\geq \frac{\bar{C}}{\gamma\bar{\eta}}\langle g(x_k), x_k - y_k\rangle \|g(x_k)\| \|x_k - y_k\| \geq \frac{\bar{C}}{\gamma\bar{\eta}}\langle g(x_k), x_k - y_k\rangle^2.$$

From Lemma 3.2, due to (16), for all $k \in \mathbb{N}_2$, one obtains the relation

$$f(x_k) - f(x_{k+1}) \geq \frac{-\bar{C}t_k}{\varepsilon_k \|y_k - x_k\|^v}\langle g(x_k), y_k - x_k\rangle \geq \frac{\bar{C}}{\bar{\varepsilon}\bar{\eta}^v}\langle g(x_k), x_k - y_k\rangle^2.$$

Thus, for all $k \in \mathbb{N}$, we have arrived at the inequality

$$f(x_k) - f(x_{k+1}) \geq \tilde{C} \cdot \langle g(x_k), x_k - y_k\rangle^2, \tag{20}$$

where $\tilde{C} = \dfrac{\bar{C}}{\bar{\eta}} \min\left\{\dfrac{1}{\gamma}, \dfrac{1}{\bar{\varepsilon}\bar{\eta}^{v-1}}\right\}$. If there takes place the following relation

$$\sigma_k \min_{x\in D}\langle g(x_k), x - x_k\rangle \leq -\alpha_k, \text{ then we observe}$$
$$\langle g(x_k), x_k - y_k\rangle \geq \alpha_k \geq \frac{\alpha_k}{\gamma\xi}\langle g(x_k), x_k - p_k^*\rangle \geq \frac{\alpha}{\gamma\xi}\langle g(x_k), x_k - p_k^*\rangle,$$

where $\xi = \sup\{\|x - y\|, x, y \in M_D(f, x_0)\} < \infty$. When there is true the inequality $\sigma_k \min\limits_{x\in D}\langle g(x_k), x - x_k\rangle > -\alpha_k$, we obtain

$$\langle g(x_k), x_k - y_k\rangle \geq -\sigma_k \min_{x\in D}\langle g(x_k), x - x_k\rangle$$
$$\geq \sigma_k\langle g(x_k), x_k - p_k^*\rangle \geq \sigma\langle g(x_k), x_k - p_k^*\rangle.$$

Thus, there holds the estimate $\langle g(x_k), x_k - y_k\rangle \geq C_1\langle g(x_k), x_k - p_k^*\rangle$, where $C_1 = \min\{\sigma, \dfrac{\alpha}{\gamma\xi}\}$. Taking into account the latter, using (20), one can quite easily obtain

$$f(x_k) - f(x_{k+1}) \geq C_2 \cdot \langle g(x_k), x_k - p_k^*\rangle^2, C_2 = \tilde{C}C_1^2.$$

Set $C_3 = \theta^2 C_2$ and evaluate for all $k \in \mathbb{N}$

$$f(x_k) - f(x_{k+1}) \geq C_3 \cdot (f(x_k) - f(p_k^*))^2.$$

Due to Lemma 3.1, the latter implies that the sequence $\{x_k\}$, $k \in \mathbb{N}$ is weakly convergent to a solution of (1), since there holds the following estimate for the convergence rate:

$$f(x_k) - f^* \leq C_3^{-1} k^{-1}.$$

□

Notice that in the case of convexity of the function $f(x)$ being minimized, to estimate the convergence rate, the condition (e4) of Theorem 3.4 can be changed to the claim on boundedness of $D^*$. Without evaluating the rate of convergence, there can be proved that the sequence $\{x_k\}$, $k \in \mathbb{N}$ converges to a solution of problem (1) under the more weak conditions. Indeed, the following theorem is true.

**Theorem 3.5** (Convergence to the set of optimal solutions) *Let the conditions (b4), (e4), and (g4) of Theorem 3.4 be fulfilled, then for the sequence* $\{x_k\}$, $k \in \mathbb{N}$ *generated by the algorithm it holds:*

*(b5)* $\lim\limits_{k\to\infty} \langle g(x_k), y_k - x_k \rangle = 0$,
*(c5) Any limit point of* $\{x_k\}$, $k \in \mathbb{N}$ *belongs to* $D^*$, *i.e.,* $\lim\limits_{k\to\infty} \|x_k - p_k^*\| = 0$.

***Proof*** By construction, $f(x_k) \geq f(x_k)$, $\forall k \in \mathbb{N}$. Since the set $M_D(f, x_0)$ is bounded and $x_k \in M_D(f, x_0)$, $\forall k \in \mathbb{N}$, the sequence $\{x_k\}$ is bounded as well. Then, $\{f(x_k)\}$, $k \in \mathbb{N}$ converges and there holds the equality

$$\lim_{k\to\infty} f(x_k) - f(x_{k+1}) = 0.$$

From Theorems 3.2–3.3, it follows that, regardless of the choice of any rule for calculating the step length, there will be found a constant $\bar{C} > 0$ such that for all $k \in \mathbb{N}$ there holds the inequality (16). From the latter, we obtain the following estimate:

$$\bar{C}^{-1}(f(x_0) - f^*) \geq \bar{C}^{-1}(f(x_k) - f(x_{k+1}) \geq -\langle g(x_k), y_k - x_k \rangle > 0.$$

This means that the sequence $\{|\langle g(x_k), y_k - x_k \rangle|\}$ is bounded. Consequently, we can select some of its convergent subsequence. Furthermore,

$$0 \geq \overline{\lim_{k\to\infty}} |\langle g(x_k), y_k - x_k \rangle| \geq \underline{\lim_{k\to\infty}} |\langle g(x_k), y_k - x_k \rangle| \geq 0,$$

i.e., we have

$$\lim_{k\to\infty} |\langle g(x_k), y_k - x_k \rangle| = 0. \tag{21}$$

Since the sequence $\{x_k\}$ is bounded, then it has at least one limit point. Let $x^*$ be an arbitrary limit point of $\{x_k\}$. Suppose that $\{x_{k_m}\} \to x^*$, $k_m \to +\infty$. Taking into account the setting of the descent direction finding problem, we can easily obtain that for all $x \in D$, $k = 0, 1, 2, \ldots$ it holds

$$\begin{aligned} \langle g(x_k), y_k - x_k \rangle &\leq \max\{\sigma_k \min_{x\in D} \langle g(x_k), x - x_k \rangle, -\alpha_k\} \leq \\ &\leq \max\{\bar{\sigma} \langle g(x_k), x - x_k \rangle, -\alpha\}. \end{aligned}$$

Here, for the purpose of estimating from above, we just formally put

$$\bar{\sigma} = \begin{cases} \sigma, & \text{if} \min\limits_{x \in D} \langle g(x_k), x - x_k \rangle < 0, \\ 1, & \text{otherwise.} \end{cases}$$

Since $-\alpha < 0$, the equality (21) implies

$$\lim_{k_m \to +\infty} \max\{\bar{\sigma} \langle g(x_k), x - x_k \rangle, -\alpha\} = \bar{\sigma} \lim_{k_m \to +\infty} \langle g(x_k), x - x_k \rangle.$$

According to (21), for $k = k_m \to +\infty$, we then have $\langle g(x^*), x - x^* \rangle \geq 0$, $\forall x \in D$. In this case, Theorem 2.1 asserts that any limit point of the sequence $\{x_k\}$ belongs to $D^*$, i.e., $\lim\limits_{k \to \infty} \|x_k - p_k^*\| = 0$. □

## 4 Algorithms for Refinement of $\varepsilon$-Normalization Parameter

In dealing with the adaptation of $\varepsilon$-normalization parameter, several algorithms for refinement of the parameter values can be useful. By definition of the $\varepsilon$-normalized descent direction, the fulfillment of the ratio $\varepsilon \gg \mu$ implies that the convergence of the adaptive conditional gradient algorithm can be slowed down. Consequently, if at the $k$th iteration of the algorithm, an increase in the value of the $\varepsilon$-normalization parameter occurs, then it is expedient to attempt to refine the value of $\varepsilon_{k+1}$ which was computed for the next iteration by the formula $\varepsilon_{k+1} = \varepsilon_k \cdot \zeta_k$, $k = 0, 1, 2 \ldots$. Let us remind that as the rule for the calculation of $\zeta_k$ there can be utilized, for instance, (12). The construction scheme of ACGM allows one to smartly use the adaptive restart strategy when the $\varepsilon$-parameter refinement occurs. One can ask what happens when at some iteration the $\varepsilon$-normalization parameter takes the refined value. In this case, a user simply should implement the next iteration with the refined value of $\varepsilon$ for normalizing the descent direction, so nothing more would really happen. In general, there is no any need in changing the scheme of ACGM.

**The First Algorithm for Refinement of the $\varepsilon$ Parameter**
Let $k \geq 0$, $a = 0$, $b = \varepsilon_k$, $\beta \in ]0, 1[$, $\eta = (1 - \beta)^{1/(v-1)}$, $\rho > 0$.
Standard Step. Determine $c = (a + b)/2$, $t = |\langle g(x_k), s_k \rangle|$, $\bar{s} = \dfrac{t s_k}{c \|s_k\|^2}$. Check the fulfillment of the following inequality:

$$f(x_k) - f(x_k + \eta \bar{s}) \geq -\beta \eta \langle g(x_k), \bar{s} \rangle. \tag{22}$$

If (22) holds, then set $\bar{i} = 1$. Otherwise, we take to be $\bar{i} = -1$.

Step 0. Implement the Standard Step
Step 1.

$$\text{If } \bar{i} = \begin{cases} 1, & \text{then set } b = c, \\ -1, & \text{then set } a = c. \end{cases}$$

Step 2. If $b-a > \rho$, then go to Step 0. Otherwise, exit from a procedure of refinement with the value $\varepsilon_k = (a+b)/2$.

**The Second Algorithm for Refinement of the $\varepsilon$ Parameter**
Let $k \geq 0$, $a = 0$, $b = \varepsilon_k$, $\beta \in ]0, 1[$, $\eta = (1-\beta)^{1/(v-1)}$, $\rho > 0$.

Step 0. Implement the Standard Step.
Step 1. If $\bar{i} = 1$, then go to Step 2. If $\bar{i} = -1$, then go to Step 4.
Step 2. If $b-a \leq \rho$, then set $\varepsilon_k = (a+b)/2$ and terminate the procedure. Otherwise, set $b = c$ and implement the Standard Step.
Step 3. If $\bar{i} = 1$, then go to Step 2. Otherwise, set $\varepsilon_k = b$ and stop.
Step 4. If $b-a \leq \rho$, then set $\varepsilon_k = (a+b)/2$ and terminate the procedure. Otherwise, set $a = c$ and implement the Standard Step.
Step 5. If $\bar{i} = -1$, then go to Step 4. Otherwise, set $\varepsilon_k = c$ and stop.

Let $\Delta_0 = b - a = \varepsilon_{k+1}$ be the length of segment of uncertainty at the beginning of the procedure for refinement of the $\varepsilon$ parameter. Analogously,

$$\Delta_1 = \frac{\Delta_0}{2},\ \Delta_2 = \frac{\Delta_1}{2} = \frac{\Delta_0}{2^2}, \ldots, \Delta_r = \frac{\Delta_0}{2^r}.$$

Obviously, $\Delta_r \to 0$ when $r \to \infty$. For the given constant $\rho > 0$, we determine further the number $r$ (the number of dividing the line segment $[a, b]$ in half) which is necessary to guarantee the fulfillment of the inequality $\Delta_r \leq \rho$. Thus, there should be fulfilled the relation

$$\Delta_r = \frac{\Delta_0}{2^r} = \frac{\varepsilon_{k+1}}{2^r} \leq \rho.$$

Consequently,

$$2^r \geq \frac{\varepsilon_{k+1}}{\rho} \Leftrightarrow r \geq \log_2 \frac{\varepsilon_{k+1}}{\rho}.$$

The latter means that the procedures for refinement of the normalization parameter are finite for any fixed accuracy $\rho$.

The exit from the first procedure is carried out when the length of the uncertainty interval is insignificantly different from zero, i.e., when the given accuracy during the process of refining the $\varepsilon$-normalization parameter is reached. While the second algorithm, in addition to this stopping criterion, makes it possible to complete the refinement as soon as after the standard step, the variable $\bar{i}$ changes its value from 1 to $-1$ or vice versa. Due to Lemma 2.2, these conditions for the termination of the procedure logically follow from (22).

## 5 Some Numerical Experiments

We performed some computational tests on different problems of nonlinear programming. The computational results are presented in tables. In the first and second columns

**Table 1** Results of the experiments for Example 5.1

| $\varepsilon_{\text{start}}$ | $\varepsilon_{\text{end}}$ | $x^*$ | $f(x^*)$ | $k_{it}$ |
|---|---|---|---|---|
| 5.21 | 5.21 | (0.33338; 1.00000) | *− 3.5185188* | 6 |
| 5.2 | 10.4 | (0.33360; 1.50000) | − 3.5185186 | 13 |

**Table 2** Results of the experiments for Example 5.2

| $\varepsilon_{\text{start}}$ | $\varepsilon_{\text{end}}$ | $x^*$ | $f(x^*)$ | $k_{it}$ |
|---|---|---|---|---|
| 30 | 60 | (0.866736; 1.031273) | − 22.461803 | 14 |
| 30.5 | 30.5 | (0.866447; 1.031226) | *− 22.461807* | 11 |

of these tables, the notations $\varepsilon_{\text{start}}$ and $\varepsilon_{\text{end}}$ correspond to the initial and last values of the $\varepsilon$- parameter, respectively. In the last column of tables can be found the number of iterations. In this section, the notations $f(x^*)$ and $x^*$ stand for the experimentally obtained minimal value of the objective function and the point at which this value is furnished. For each table, the best value of the objective function is italicized.

***Example 5.1*** [18]

$$\min_{x\in D} f(x) = \xi_1^3 + \xi_2^6 + 4\xi_1^2\xi_1^2 - 3\xi_1 - 4\xi_2,$$
$$D = \{x \in \mathbb{R}^2 : 0 \le \xi_1 \le 7;\ 1 \le \xi_2 \le 4\}.$$

Here, there is chosen the following initial iteration point: $x_0 = (7.1)$ with $f(x_0) = 515$. In [18], the best objective function approximation $-3.518$ is reached at $x^* = (0.33;\ 1)$. For the step-size selection in ACGM, we utilize Rule 1 with $\beta = 0.5$. Table 1 contains the results of the numerical experiments for Example 5.1. In this case, we do not apply the procedure for refining the $\varepsilon$-normalization parameter.

***Example 5.2*** [18]

$$\min_{x\in D} f(x) = 5\xi_1^4 + \xi_2^6 - 13\xi_1 - 7\xi_2 - 8,$$
$$D = \{x \in \mathbb{R}^2 : -1 \le \xi_1 \le 3;\ 1 \le \xi_2 \le 5\}.$$

As a point of the first approximation, we choose $x_0 = (1.1)$ for which it holds $f(x_0) = -22$. When the method presented in [18] is applied for this problem solving, it requires 2–3 iterations. However, let us note that depending on the mesh of nodes for the method parameter, at each iteration, 11 to 101 subproblems of one-dimensional exact minimization are solved by dividing a line segment in half (for details, see [18]). The results of solving Example 5.2 (using Rule 1 with $\beta = 0.5$) are presented in Table 2.

***Example 5.3*** [18]

$$\min_{x\in D} f(x) = \xi_1^4 + \xi_2^3 + 2\xi_1^2\xi_1^2 + \xi_1 - \xi_2,$$

**Table 3** Results of the experiments from [18] for Example 5.3

| | Method from [18] | Newton's Method | FWA |
|---|---|---|---|
| $x_0$ | (5; 3) | (5; 3) | (5; 3) |
| Number of iterations | 8 | 8 | 19 |
| $x_0$ | (0;3) | (0;3) | (0;3) |
| Number of iterations | 2 | 4 | 22 |

**Table 4** Results of the implementation of ACGM for Example 5.3

| $\varepsilon_{\text{start}}$ | $\varepsilon_{\text{end}}$ | $x^*$ | $f(x^*)$ | $k_{it}$ | $t$ (s) |
|---|---|---|---|---|---|
| 0.001 | 1.773 | (0.0000000; 0.5744414) | −0.38488528 | 15 | 0 |
| 1.78 | 3.56 | (0.0000000; 0.5789270) | −0.38489586 | 9 | 0 |
| 0.001 | 2.048 | (0.0000000; 0.5779053) | *−0.38489965* | 5 | 0 |
| 0.001 | 1.97 | (0.0016876; 0.5792689) | −0.38320434 | 120 | 1 |
| 1.97 | 1.97 | (0.00168865; 0.5792696) | −0.38320541 | 164 | 1 |
| 1.78 | 1.78 | (0.0012225; 0.5802417) | −0.38366213 | 116 | 1 |

$$D = \{x \in \mathbb{R}^2 : 0 \le \xi_1 \le 5; 0.1 \le \xi_2 \le 3\}.$$

This test problem was used to illustrate how the algorithm described in [18] works in comparison with FWA and Newton's method. Computations were implemented with the different starting points $x_0 \in D$. But unfortunately, what problem solutions were obtained at that time is not represented in [18]. Table 3 contains only the information from [18] related to the number of iterations for the above-mentioned methods.

Table 4 includes the results of our experiments for Example 5.3. The last table column indicates the program execution time in seconds. In the first three cases, we use $x_0 = (0; 3)$, while in the others $x_0 = (5; 3)$. For the second and third experiments, the step size is selected by means of Rule 1 with $\beta = 0.5$. For the others, there is utilized the same step-size choosing rule with $\beta = 0.9$. As one can easily see, the runtime of ACGM is very short, so there is no any need in the use of the refinement procedure for the $\varepsilon$-normalization parameter.

***Example 5.4*** (A multistage compressor optimization) [17]

$$\min_{x \in D} f(x) = \xi_1^{1/4} + (\xi_1/\xi_2)^{1/4} + (64/\xi_2)^{1/4}.$$
$$D = \{x \in \mathbb{R}^2 : 1 \le \xi_1;\ \xi_1 \le \xi_2;\ \xi_2 \le 14\}.$$

The specificity of this test problem consists in that the objective function attains its minimum on the boundary of the feasible domain $D$. Beginning from the initial point $x_0 = (14; 14)$ with $f(x_0) = 4.306557$, there was obtained the following solution: $x^* = (3.7417; 14)$ with $f(x^*) = 4.2438291$ at the next iteration of FWA. We refer the interested reader to [17], for more details and graphical interpretation. The best solution

for Example 5.4 was obtained in the case of applying ACGM with Rule 1 when $\beta = 0.5$ (without refining the $\varepsilon$-normalization parameter). After eight iterations, there was reached the objective function minimal value $f(x^*) = 4.243829$ at the following point: $x^* = (3.742954; 14.000000)$. For this experiment, $\varepsilon_{\text{start}} = \varepsilon_{\text{end}} = 0.00497$.

The numerical implementation of ACGM confirms the expediency of using the step-by-step adaptation procedure for the $\varepsilon$-normalization parameter (see also [15]). Indeed, our experiments showed that the parameter of normalizing the descent direction is stabilized, beginning from a certain iteration. Note that from the same iteration the value of the step becomes constant. From that moment, for finding the step size, only one calculation of the objective function value is performed (to verify the fulfillment of the step selection condition). Experiments have also indicated that most of the best objective function values correspond to Rule 1 with $\beta = 0.5$ (or $\beta = 0.9$) and the first refinement algorithm (in the case when the parameter refinement procedure was used). For all test problems, there were found the points at which the value of the objective function was less than or equal to known values. A carried out experimental study has demonstrated the efficiency of the adaptive variant of CGM and ability to lead to the minimum neighborhood fairly quickly and at low computational costs.

## 6 Conclusions

Finally, we note that the presented fully adaptive conditional gradient algorithm as compared with the classical Frank–Wolfe algorithm has the advantage consisting in a possibility of inexact solving of the direction finding subproblem and handling the accuracy of its solution. Moreover, the adaptive method does not require any exact line search for computing the length of the iteration step size. We proposed some novel rules for the calculation of the step length in which the iteration step is regulated additionally by an adaptation of the $\varepsilon$-normalization parameter for the descent direction. There was justified the finiteness of the procedures of adaptive controlling both the parameter of an $\varepsilon$-normalization of a descent direction and the step length. For the problem of minimizing a continuously differentiable pseudo-convex function on a convex and closed subset of Euclidean space, we justified the sublinear rate of the convergence for the adaptive variant of the conditional gradient algorithm.

One of the motivating ideas was that of using in the future the adaptive method to solve the problems of sets separation (in particular, the programs of projecting the point onto the convex polyhedron) as well as some related problems of data mining.

**Acknowledgements** The author thanks the anonymous referees and the editor for their helpful comments and remarks on a previous version of the paper.

## References

1. Dunn, J.C., Hafghbarger, S.: Conditional gradient algorithms with open loop step size rules. J. Math. Anal. Appl. **62**, 432–444 (1978)

2. Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P.: Block-coordinate Frank–Wolfe optimization for structural SVMs. In: Proceedings of the 30th International Conference on Machine Learning, PMLR, vol. 28(1), pp. 53–61 (2013)
3. Braun, G., Pokutta, S., Zink, D.: Lazifying conditional gradient algorithms. arXiv preprint arXiv:1610.051220v4 (2018)
4. Lacoste-Julien, S.: Convergence rate of Frank–Wolfe for non-convex objectives. arXiv preprint arXiv:1607.00345 (2016)
5. Nesterov, Y.: Complexity bounds for primal-dual methods minimizing the model of objective function. Math. Program. **171**(1–2), 311–330 (2018)
6. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
7. Yu, Y., Zhang, X., Schuurmans, D.: Generalized conditional gradient for sparse estimation. arXiv preprint arXiv:1410.4828v1 (2014)
8. Gabidullina, Z.R.: Relaxation methods with step regulation for solving constrained optimization problems. J. Math. Sci. **73**(5), 538–543 (1995)
9. Mitchell, V.F., Dem'yanov, V.F., Malozemov, V.N.: Finding the point of polyhedron closest to origin. SIAM J. Control Optim. **12**, 19–26 (1974)
10. Gabidullina, Z.R.: The problem of projecting the origin of euclidean space onto the convex polyhedron. Lobachevskii J. Math. **39**(1), 35–45 (2018)
11. Gabidullina, Z.R.: Solving of a projection problem for convex polyhedra given by a system of linear constraints. In: Constructive Nonsmooth Analysis and Related Topics (Dedicated to the Memory of V.F. Demyanov), CNSA Proceedings—IEEE, Art. 7973958 (2017). http://ieeexplore.ieee.org/abstract/document/7973958/
12. Gabidullina, Z.R.: The Minkowski difference for convex polyhedra and some its applications. arXiv preprint arXiv:1903.03590 (2019)
13. Mangasarian, O.L.: Pseudo-convex functions. J. Soc. Ind. Appl. Math. Ser. A Control **3**, 281–290 (1965)
14. Gabidullina, Z.R.: Convergence of the constrained gradient method for a class of nonconvex functions. J. Sov. Math. **50**(5), 1803–1809 (1990)
15. Gabidullina, Z.R.: Adaptive methods with step length regulation for solving pseudo-convex programming problems. Dissertation for the Degree of Candidate of Science in Physics and Mathematics, Kazan (1994)
16. Vasil'ev, F.P.: Numerical Methods for Solving Extremum Problems. Nauka, Moscow (1980)
17. Reklaitis, G.V., Ravindran, A., Ragsdell, K.M.: Engineering Optimization: Methods and Applications, 2nd edn. Wiley, Hoboken (2006)
18. Beltukov, I.B., Shurygina, M.N.: A study of one adaptive method for mathematical programming. In: Optimization Methods and Applications, pp. 5–13. Irkutsk (1988) **(in Russian)**