J. Nonlinear Var. Anal. 1 (2017), No. 1, pp. 25-41
Available online at http://jnva.biemdas.com

THE METHOD OF PAIRWISE VARIATIONS WITH TOLERANCES FOR LINEARLY CONSTRAINED OPTIMIZATION PROBLEMS

IGOR V. KONNOV

Department of System Analysis and Information Technologies, Kazan Federal University, Kazan, Russia

Abstract. We consider a method of pairwise variations for smooth optimization problems, which involve polyhedral constraints. It consists in making steps with respect to the difference of two selected extreme points of the feasible set together with special threshold control and tolerances whose values decrease sequentially. The method is simpler and more flexible than the well-known conditional gradient method, but keeps its useful sparsity properties and is very suitable for large dimensional optimization problems. We establish its convergence under rather mild assumptions. Efficiency of the method is confirmed by its convergence rates and results of computational experiments.

Keywords. Conditional gradient method; Optimization problem; Polyhedral feasible set; Pairwise variation; Threshold control.

2010 Mathematics Subject Classification. 90C30, 90C25, 90C06.

1. Introduction

The usual optimization problem consists in finding the minimal value of some goal function $f:\mathbb{R}^m \to \mathbb{R}$ on a feasible set D such that $D \subseteq \mathbb{R}^m$. For brevity, we write this problem as

$$\min_{x\in D} \to f(x), \tag{1.1}$$

its solution set is denoted by D^* and the optimal value of the function by f^*, i.e.

$$f^* = \inf_{x\in D} f(x).$$

We shall consider a special class of optimization problems, where the function f is supposed to be smooth on D and the set D is a nonempty polyhedron, i.e., it is bounded and defined by affine constraints, e.g.

$$D = \left\{ x \in \mathbb{R}^m : \langle q^i, x\rangle \leq \beta_i,\ i = 1,\dots,l \right\},$$

where $\langle q, x\rangle$ denotes the usual scalar product of q and x. Then problem (1.1) has a solution.

The conditional gradient method is one of the oldest methods, which can be applied to the above problem. It was first suggested in [1] for the case when the goal function is quadratic and further was developed by many authors; see e.g. [2, 3, 4, 5]. We recall that the main idea of this method consists in

E-mail address: konn-igor@ya.ru.

Received January 27, 2017, Accepted March 5, 2017.

linearization of the goal function. That is, given the current iterate $x^k \in D$, one finds some solution y^k of the problem

$$\min_{y \in D} \to \langle f'(x^k), y \rangle \tag{1.2}$$

and defines $p^k = y^k - x^k$ as a descent direction at x^k. Taking a suitable stepsize $\lambda_k \in (0,1]$, one sets $x^{k+1} = x^k + \lambda_k p^k$ and so on.

During rather long time, this method was not considered as very efficient because of its relatively slow convergence in comparison with Newton and projection type methods. However, it became very popular recently due to several features significant for many applications, where huge dimensionality and inexact data create certain drawbacks for more rapid methods. In particular, its auxiliary linearized problems of form (1.2) appear simpler essentially than the quadratic ones of the most other methods. Next, it usually yields so-called sparse approximations of a solution with few non-zero components; see e.g. [6, 7]. Many efforts were directed to enhance the convergence properties of the conditional gradient method; see e.g. [8, 9, 7, 10] and the references therein. In particular, inserting the so-called away steps enabled one to attain the linear rate of convergence for some classes of optimization problems significant for applications; see e.g. [8, 11, 12, 13].

In this paper, we intend to present some other modification of the conditional gradient method, which seems more flexible and reduces the total computational expenses. The main idea follows the bi-coordinate descent method with special threshold control and tolerances for optimization problems with simplex constraints that was proposed in [14]. Unlike the previous methods, its direction choice requirements are relaxed essentially and admit different implementation versions. Its more detailed comparison with the other methods is given in Section 5.

In the next section, we give several basic properties of problem (1.1), which will be used for the substantiation of the method. In Section 3, we describe the new method and prove its convergence in the general case. In Section 4, we specialize its convergence properties for the case where the gradient of the goal function is Lipschitz continuous, propose some simplifications and obtain the complexity estimate of the method. In Section 5, we discuss its implementation issues and provide its comparison with the previously known methods. Section 6 describes the results of computational experiments.

2. Preliminary properties

We start our consideration from recalling the well known optimality condition; see e.g. [15, Theorem 11.1].

Lemma 2.1. *(a) Each solution of problem (1.1) is a solution of the variational inequality (VI for short): Find a point $x^* \in D$ such that*

$$\langle f'(x^*), x - x^* \rangle \geq 0 \quad \forall x \in D. \tag{2.1}$$

(b) If f is convex, then each solution of VI (2.1) solves problem (1.1).

We denote by D^0 the solution set of VI (2.1), its elements are called stationary points of problem (1.1).

We intend to specialize optimality conditions for problem (1.1). First we note that

$$D = \left\{ x \in \mathbb{R}^m : \ x = \sum_{i \in I} u_i z^i, \ \sum_{i \in I} u_i = 1, \ u_i \geq 0, \ i \in I \right\}, \tag{2.2}$$

where z^i is the i-th extreme point (vertex) of the polyhedron D, I is the set of indices of its extreme points, which is finite, i.e., we can set $I = \{1, \dots, n\}$. Given a point $x \in D$, we can hence define the corresponding vector of weights $u(x) = (u_1(x), \dots, u_n(x))^\top$ of some its associated representation

$$x = \sum_{i \in I} u_i(x) z^i, \ \sum_{i \in I} u_i(x) = 1, \ u_i(x) \geq 0, \ i \in I. \tag{2.3}$$

Clearly, $u(x)$ is not defined uniquely in general. Now we give the useful property of solutions of linear programming (LP for short) problems; see [16, Section 3.3].

Lemma 2.2. *Let c be a fixed vector in $\mathbb{R}^m$.*

(i) If a point x^* is a solution of the LP problem

$$\min_{x \in D} \to \langle c, x \rangle, \tag{2.4}$$

and

$$x^* = \sum_{i \in I} u_i^* z^i, \ \sum_{i \in I} u_i^* = 1, \ u_i^* \geq 0, \ i \in I; \tag{2.5}$$

then

$$\langle c, z^i \rangle \begin{cases} \geq \langle c, x^* \rangle & \text{if } u_i^* = 0, \\ = \langle c, x^* \rangle & \text{if } u_i^* > 0, \end{cases} \quad \text{for } i \in I. \tag{2.6}$$

(ii) If a point $x^* \in D$ satisfies conditions (2.6) for some representation (2.5), then it solves problem (2.4).

Proof. Let a point x^* be a solution of problem (2.4) and (2.5) holds. By definition,

$$\langle c, x^* \rangle \leq \langle c, z^i \rangle, \ \forall i \in I. \tag{2.7}$$

Define the index sets $I_+ = \{i \in I \mid u_i^* > 0\}$ and $I_0 = \{i \in I \mid u_i^* = 0\}$ and choose $s \in I_+$. Then $u_s^* > 0$ and

$$\begin{aligned} \langle c, x^* \rangle &= \sum_{i \in I_+} u_i^* \langle c, z^i \rangle = u_s^* \langle c, z^s \rangle + \sum_{i \in I_+, i \neq s} u_i \langle c, z^i \rangle \\ &\geq u_s^* \langle c, z^s \rangle + (1 - u_s^*) \langle c, x^* \rangle. \end{aligned}$$

It follows that $\langle c, x^* \rangle \geq \langle c, z^s \rangle$, hence $\langle c, x^* \rangle = \langle c, z^s \rangle$ in view of (2.7). Assertion (i) is true.

Conversely, let a point $x^* \in D$ satisfy conditions (2.6) for some representation (2.5). Take an arbitrary point $x \in D$ and some associated weight vector $v = u(x)$, then

$$x = \sum_{i \in I} v_i z^i, \ \sum_{i \in I} v_i = 1, \ v_i \geq 0, \ i \in I.$$

It follows from (2.6) that

$$\langle c, x \rangle = \sum_{i \in I} v_i \langle c, z^i \rangle \geq \langle c, x^* \rangle \sum_{i \in I} v_i = \langle c, x^* \rangle,$$

and assertion (ii) holds true. □

Now we are ready to give optimality conditions for VI (2.1), hence for problem (1.1).

Proposition 2.1. *A point x^* with representation (2.5) is a solution of VI (2.1) if and only if it satisfies each of the following equivalent conditions:*

$$x^* \in D,\ \langle f'(x^*), z^i \rangle \begin{cases} \geq \langle f'(x^*), x^* \rangle & \textit{if } u_i^* = 0, \\ = \langle f'(x^*), x^* \rangle & \textit{if } u_i^* > 0, \end{cases} \quad \textit{for } i \in I; \tag{2.8}$$

$$x^* \in D,\ \forall i, j \in I,\ \langle f'(x^*), z^i \rangle > \langle f'(x^*), z^j \rangle \implies u_i^* = 0; \tag{2.9}$$

$$x^* \in D,\ \forall i, j \in I,\ u_i^* > 0 \implies \langle f'(x^*), z^i \rangle \leq \langle f'(x^*), z^j \rangle. \tag{2.10}$$

Proof. From Lemma 2.2 we clearly have that a point x^* with some representation (2.5) is a solution of VI (2.1) if and only if it satisfies (2.8). Clearly, (2.8) implies (2.9) and (2.9) implies (2.10). Let now a point $x^* \in D$ with $u^* = u(x^*)$ satisfy (2.10). Then there exists an index k such that $u_k^* > 0$. Set

$$\alpha = \min_{i \in I} \langle f'(x^*), z^i \rangle.$$

Then (2.10) implies $\langle f'(x^*), z^i \rangle = \alpha$ if $u_i^* > 0$ and $\langle f'(x^*), z^i \rangle \geq \alpha$ if $u_i^* = 0$, hence (2.8) holds. □

Given a number $\varepsilon > 0$ and a point $x \in D$ with some associated weight vector $u(x)$ from (2.3), let

$$I_\varepsilon(x) = \{i \in I : u_i(x) \geq \varepsilon\}.$$

The number of vertices may be too large, however, we can evaluate the weights implicitly from feasible step-sizes.

Proposition 2.2. *Given $x \in D$ and $i \in I$, let*

$$x + \alpha(z^j - z^i) \notin D \textit{ for some } j \in I,\ \textit{if } \alpha > \varepsilon > 0. \tag{2.11}$$

Then $u_i(x) \leq \varepsilon$ for any weight vector $u(x)$.

Proof. On the contrary, suppose that (2.11) holds, but there exists a weight vector $u = u(x)$ with $u_i > \varepsilon$. Take $\alpha = u_i$ and arbitrary $j \in I$. Then can define the point $y = x + \alpha(z^j - z^i)$ such that

$$y = \sum_{s \in I} u_s z^s + \alpha(z^j - z^i) = \sum_{s \in I} v_s z^s,$$

where

$$v_s = \begin{cases} 0, & \text{if } s = i, \\ u_j + u_i, & \text{if } s = j, \\ u_s, & \text{otherwise;} \end{cases}$$

besides,

$$\sum_{s \in I} v_s = 1,\ v_s \geq 0,\ s \in I.$$

It follows that $y \in D$, a contradiction. □

3. Method and its convergence

The method of pairwise variations with tolerances (PVM for short) for VI (2.1) is described as follows. Let $\mathbb{Z}_+$ denote the set of non-negative integers.

Method (PVM).
Initialization: Choose a point $w^0 \in D$, numbers $\beta \in (0,1)$, $\theta \in (0,1)$, and sequences $\{\delta_l\} \searrow 0$, $\{\varepsilon_l\} \searrow 0$ with $\varepsilon_0 \in (0,1)$. Set $l := 1$.
Step 0: Set $k := 0$, $x^0 := w^{l-1}$.
Step 1: Choose an index $i \in I_{\varepsilon_l}(x^k)$ for some associated weight vector $u^k = u(x^k)$ and an index $j \in I$ such that

$$\langle f'(x^k), z^i - z^j \rangle \geq \delta_l, \tag{3.1}$$

choose $\gamma_k \in [\varepsilon_l, u^k_i]$, set $i_k := i$, $j_k := j$ and go to Step 2. Otherwise (i.e. if (3.1) does not hold for all $i \in I_{\varepsilon_l}(x^k)$ associated to some weight vector $u(x^k)$ and $j \in I$) set $w^l := x^k$, $l := l+1$ and go to Step 0. *(Restart)*
Step 2: Set $d^k := z^{j_k} - z^{i_k}$, determine m_k as the smallest number in $\mathbb{Z}_+$ such that

$$f(x^k + \theta^{m_k}\gamma_k d^k) \leq f(x^k) + \beta\theta^{m_k}\gamma_k \langle f'(x^k), d^k \rangle, \tag{3.2}$$

set $\lambda_k := \theta^{m_k}\gamma_k$, $x^{k+1} := x^k + \lambda_k d^k$, $k := k+1$ and go to Step 1.

Thus, the method has a two-level structure where each outer iteration (stage) l contains some number of inner iterations in k with the fixed tolerances δ_l and ε_l. Completing each stage, that is marked as restart, leads to decrease of their values.

Note that $i_k \neq j_k$ due to (3.1), besides, $\gamma_k \geq \varepsilon_l$ and the point $x^k + \gamma_k d^k$ is always feasible. Moreover, by definition,

$$\mu_k = \langle f'(x^k), d^k \rangle = \langle f'(x^k), z^{j_k} - z^{i_k} \rangle \leq -\delta_l < 0, \tag{3.3}$$

in (3.2). It follows that

$$f(x^{k+1}) \leq f(x^k) + \beta\lambda_k\mu_k \leq f(x^k) - \beta\lambda_k\delta_l. \tag{3.4}$$

We now justify the linesearch.

Lemma 3.1. *The linesearch procedure in Step 2 is always finite.*

Proof. If we suppose that the linesearch procedure is infinite, then (3.2) does not hold and

$$(\theta^{m_k}\gamma_k)^{-1}(f(x^k + \theta^{m_k}\gamma_k d^k) - f(x^k)) > \beta\mu_k,$$

for $m_k \to \infty$. Hence, by taking the limit we have $\mu_k \geq \beta\mu_k$, hence $\mu_k \geq 0$, a contradiction with $\mu_k \leq -\delta_l < 0$ in (3.3). □

We show that each stage is well defined.

Proposition 3.1. *The number of iterations at each stage l is finite.*

Proof. Fix any l. Since the sequence $\{x^k\}$ is bounded, it has limit points. Besides, by (3.4), we have $f^* \leq f(x^k)$ and $f(x^{k+1}) \leq f(x^k) - \beta \delta_l \lambda_k$, hence

$$\lim_{k\to\infty} \lambda_k = 0.$$

Suppose that the sequence $\{x^k\}$ is infinite. Since the set I is finite, there is a pair of indices $(i_k, j_k) = (i, j)$, which is repeated infinitely. Take the corresponding subsequence $\{k_s\}$, then $d^{k_s} = \bar{d} = z^j - z^i$. Without loss of generality, we can suppose that the subsequence $\{x^{k_s}\}$ converges to a point $\bar{x}$ and due to (3.3) we have

$$\langle f'(\bar{x}), \bar{d}\rangle = \lim_{s\to\infty} \langle f'(x^{k_s}), \bar{d}\rangle \leq -\delta_l.$$

However, (3.2) does not hold for the step-size λ_k/θ. Setting $k = k_s$ gives

$$(\lambda_{k_s}/\theta)^{-1}(f(x^{k_s} + (\lambda_{k_s}/\theta)\bar{d}) - f(x^{k_s})) > \beta\langle f'(x^{k_s}), \bar{d}\rangle,$$

hence, by taking the limit $s \to \infty$ we obtain

$$\langle f'(\bar{x}), \bar{d}\rangle = \lim_{s\to\infty} \left\{(\lambda_{k_s}/\theta)^{-1}(f(x^{k_s} + (\lambda_{k_s}/\theta)\bar{d}) - f(x^{k_s}))\right\} \geq \beta\langle f'(\bar{x}), \bar{d}\rangle,$$

i.e., $(1-\beta)\langle f'(\bar{x}), \bar{d}\rangle \geq 0$, which is a contradiction. □

We are ready to prove convergence of the whole method.

Theorem 3.1. *Under the assumptions made it holds that:*

(i) the number of changes of index k at each stage l is finite;

(ii) the sequence $\{w^l\}$ generated by method (PVM) has limit points, all these limit points are solutions of VI (2.1);

(iii) if f is convex, then

$$\lim_{l\to\infty} f(w^l) = f^*; \tag{3.5}$$

and all the limit points of $\{w^l\}$ belong to D^*.

Proof. Assertion (i) has been obtained in Proposition 3.1. By construction, the sequence $\{w^l\}$ is bounded, hence it has limit points. Moreover, $f(w^{l+1}) \leq f(w^l)$, hence

$$\lim_{l\to\infty} f(w^l) = \mu. \tag{3.6}$$

Take an arbitrary limit point $\bar{w}$ of $\{w^l\}$, then $\bar{w} \in D$,

$$\lim_{t\to\infty} w^{l_t} = \bar{w}.$$

By definition (2.2), each point w^l is associated with some weight vector $v^l = u(w^l)$ such that

$$w^l = \sum_{s\in I} v^l_s z^s, \ \sum_{i\in I} v^l_s = 1, \ v^l_s \geq 0, \ s \in I.$$

Clearly, the sequence $\{v^l\}$ is bounded and must have limit points. Without loss of generality we can suppose that

$$\bar{v} = \lim_{t\to\infty} v^{l_t},$$

then

$$\bar{w} = \sum_{s\in I} \bar{v}_s z^s, \ \sum_{i\in I} \bar{v}_s = 1, \ \bar{v}_s \geq 0, \ s \in I.$$

For $l > 0$ we must have

$$\langle f'(w^l), z^i - z^j \rangle \leq \delta_l \text{ for all } i, j \in I \text{ with } v_i^l \geq \varepsilon_l.$$

Let p be an arbitrary index such that $\bar{v}_p > 0$. Then $v_p^{l_t} \geq \varepsilon_{l_t}$ for t large enough, hence

$$\langle f'(w^{l_t}), z^p - z^q \rangle \leq \delta_{l_t} \text{ for all } q \in I.$$

Taking the limit $t \to \infty$, we obtain

$$\langle f'(\bar{w}), z^p - z^q \rangle \leq 0 \text{ for all } q \in I.$$

This means that the point $\bar{w}$ satisfies the optimality conditions (2.10). Due to Proposition 2.1, $\bar{w}$ solves VI (2.1) and assertion (ii) holds. Next, if f is convex, then by Lemma 2.1 each limit point of $\{w^l\}$ belongs to D^0, hence $\mu = f^*$ in (3.6). This gives (3.5) and assertion (iii). □

4. Convergence in the Lipschitz gradient case

The above descent method is very flexible and admits various modifications and extensions. In particular, we can take the exact one-dimensional minimization rule instead of the current Armijo rule in (3.2). The convergence then can be obtained along the same lines; see e.g. [17, Section 6.1].

If the gradient of the function f is Lipschitz continuous on D with some constant $L > 0$, i.e., $\|f'(y) - f'(x)\| \leq L\|y - x\|$ for any vectors x and y, we can take the useful property of such functions

$$f(y) \leq f(x) + \langle f'(x), y - x \rangle + 0.5L\|y - x\|^2;$$

see [3, Chapter III, Lemma 1.2]. This gives us an explicit lower bound for the step-size. In fact, at Step 2 we have

$$f(x^k + \lambda d^k) - f(x^k) \leq \lambda[\langle f'(x^k), d^k \rangle + 0.5L\lambda\|d^k\|^2] \leq \beta\lambda\langle f'(x^k), d^k \rangle,$$

if $\lambda \leq -(1-\beta)\langle f'(x^k), d^k \rangle / (L\|d^k\|^2)$. However, $\langle f'(x^k), d^k \rangle \leq -\delta_l$ at stage l, besides, $\|d^k\| \leq B = \text{Diam}D < \infty$. If we take $\lambda_k = \lambda\delta_l$ with $\lambda \in (0, \bar{\lambda}]$ and

$$\bar{\lambda} = \min\{(1-\beta)/(LB^2), \varepsilon_l\},$$

then

$$f(x^k + \lambda_k d^k) \leq f(x^k) + \beta\lambda_k\langle f'(x^k), d^k \rangle, \tag{4.1}$$

as desired. In such a way we can drop the line-search procedure in Step 2. Obviously, the assertions of Proposition 3.1 and Theorem 3.1 remain true for this version. This version reduces the computational expenses essentially but require the evaluation of the Lipschitz constants. We can use several approaches to avoid this drawback.

Firstly, we can apply the step-size rule $\lambda_0 \in (0, 1]$ and $\lambda_k = \varepsilon_l\delta_l$ at stage l without any line-search. Then $\varepsilon_l \leq \bar{\lambda}$ for l large enough and the convergence can be proved as in the previous case since the values of the function f are bounded from above on the compact set D. Besides, after the finite number of stages we will have the basic inequality $f(w^{l+1}) \leq f(w^l)$, which implies (3.6).

Secondly, we can apply the divergent step-size rule

$$\sum_{k=0}^{\infty} \lambda_k = \infty, \ \sum_{k=0}^{\infty} \lambda_k^2 < \infty, \ \lambda_k \in (0, \varepsilon_l], \ k = 1, 2, \dots, \tag{4.2}$$

at stage l. For instance, we can set $\lambda_k = \varepsilon_l/(k+1)$. Then again $\lambda_k \leq \bar{\lambda}\delta_l$ for k large enough. Then the assertion of Proposition 3.1 remains true. In fact, if we suppose that the sequence $\{x^k\}$ is infinite, (3.4) gives $f(x^s) \leq f(x^{s-1}) - \beta\delta_l\lambda_s$, hence

$$f^* \leq f(x^k) \leq f(x^0) - \beta\delta_l \sum_{s=0}^{k} \lambda_s,$$

which is a contradiction. Then assertion (ii) of Theorem 3.1 can be proved as above. Assertion (iii) follows from (ii) and the continuity of f. Therefore, rule (4.2) also provides convergence.

Of course, there is no necessity now to evaluate the Lipschitz constant and diameter of D.

Due to Lemma 2.1, the value

$$\Delta(x) = \max_{y \in D} \langle f'(x), x - y \rangle$$

gives a gap function for VI (2.1). We intend to obtain an error bound for VI (2.1) at w^l. Since D is compact, we can define

$$\sigma = \max_{i \in I} \max_{x \in D} \langle f'(x), z^i \rangle.$$

Proposition 4.1. *For each stage l, we have*

$$\Delta(w^l) \leq \delta_l + 2n\varepsilon_l\sigma. \tag{4.3}$$

Proof. By definition,

$$\min_{y \in D} \langle f'(w^l), y \rangle = \langle f'(x), z^t \rangle$$

for some $t \in I$. We recall that $u(w^l) = v^l$ and $I_{\varepsilon_l}(w^l) = \{i \in I \mid v_i^l \geq \varepsilon_l\}$. It follows that

$$\begin{aligned}
\Delta(w^l) &= \sum_{s \in I} v_s^l \langle f'(w^l), z^s \rangle - \langle f'(x), z^t \rangle = \sum_{s \in I} v_s^l \langle f'(w^l), z^s - z^t \rangle \\
&= \sum_{s \in I_{\varepsilon_l}(w^l)} v_s^l \langle f'(w^l), z^s - z^t \rangle + \sum_{s \notin I_{\varepsilon_l}(w^l)} v_s^l \langle f'(w^l), z^s - z^t \rangle \\
&\leq \delta_l \sum_{s \in I_{\varepsilon_l}(w^l)} v_s^l + \varepsilon_l \sum_{s \notin I_{\varepsilon_l}(w^l)} \langle f'(w^l), z^s - z^t \rangle \\
&\leq \delta_l + 2n\varepsilon_l\sigma.
\end{aligned}$$

Therefore, estimate (4.3) holds true. □

As the method has a two-level structure with each stage containing a finite number of inner iterations, it is more suitable to derive its complexity estimate, which gives the total amount of work of the method. We now suppose that the function f is convex and its gradient is Lipschitz continuous with constant L. For simplicity, we take the above version with the fixed stepsize $\lambda_k = \bar{\lambda}\delta_l$.

We take the value $\Phi(x) = f(x) - f^*$ as an accuracy measure for our method. More precisely, given a starting point w^0 and a number $\alpha > 0$, we define the complexity of the method, denoted by $N(\alpha)$, as the total number of inner iterations at $l(\alpha)$ stages such that $l(\alpha)$ is the maximal number l with $\Phi(w^l) \geq \alpha$, hence,

$$N(\alpha) \leq \sum_{l=1}^{l(\alpha)} N_{(l)}, \tag{4.4}$$

where $N_{(l)}$ denotes the total number of iterations at stage l. We proceed to estimate the right-hand side of (4.4). To change the parameters, we apply the rule

$$\delta_l = \varepsilon_l = \nu^l \delta_0, l = 0, 1, \dots; \quad \nu \in (0,1), \delta_0 > 0. \tag{4.5}$$

By (4.1), we have

$$f(x^{k+1}) \leq f(x^k) - \beta \bar{\lambda} \delta_l^2,$$

hence

$$N_{(l)} \leq \Phi(w^{l-1})/(\beta \bar{\lambda} \delta_l^2). \tag{4.6}$$

Under the above assumptions from Proposition 4.1 we obtain

$$\Phi(w^l) = f(w^l) - f^* \leq \Delta(z^l) \leq \delta_l + 2n\sigma\varepsilon_l = \delta_0 C_1 \nu^l,$$

where $C_1 = 1 + 2n\sigma$. It follows that

$$\nu^{-l(\alpha)} \leq \delta_0 C_1/\alpha.$$

Besides, using (4.6) now gives

$$N_{(l)} \leq C_1 \delta_0 \nu^{l-1}/(\beta \bar{\lambda} \nu^{2l} \delta_0^2) = C_1 L B^2/(\beta(1-\beta)\nu^{l+1}\delta_0) = C_2 \nu^{-l-1},$$

where $C_2 = C_1 L B^2/(\beta(1-\beta)\delta_0)$.

Combining both the inequalities in (4.4), we obtain

$$\begin{aligned} N(\alpha) \quad & \leq C_2 \sum_{l=1}^{l(\alpha)} \nu^{-l-1} \leq C_2 \nu(\nu^{-l(\alpha)} - 1)/(1-\nu) \\ & \leq C_2 \nu (C_1/\alpha - 1)/(1-\nu). \end{aligned}$$

We have established the complexity estimate.

Theorem 4.1. *Let the function $f : X \to \mathbb{R}$ be convex and its gradient be Lipschitz continuous with constant L. Let a sequence $\{w^l\}$ be generated by (PVM) with the stepsize rule $\lambda_k = \bar{\lambda}\delta_l$ at stage l. If the parameters satisfy conditions (4.5), the method has the complexity estimate*

$$N(\alpha) \leq C_2 \nu (C_1/\alpha - 1)/(1-\nu),$$

where $C_1 = 1 + 2n\sigma$ and $C_2 = C_1 L B^2/(\beta(1-\beta)\delta_0)$.

We see that the above estimate corresponds to those of the usual conditional gradient methods, which solves the linearized problem (1.2) at each iteration; see [2, 5].

5. Implementation issues

In this section, we discuss some questions of implementation of (PVM) for different kinds of feasible sets and provide its comparison with the previously known methods. In fact, implementation of (PVM) requires some associated weight vector $u^k = u(x^k)$ for each iteration point x^k. This vector is used for finding a suitable index $i \in I_{\varepsilon_l}(x^k)$. We again note that it suffices to have an arbitrary weight vector of x^k. The first way is to choose such a vector at the starting point w^0 and change it sequentially in conformity with the iteration process. For the sake of clarity, we give its full description now.

(PVM) with explicit weight changes.

Initialization: Choose a point $w^0 \in D$ with some associated weight vector $v^0 = u(w^0)$, numbers $\beta \in (0,1)$, $\theta \in (0,1)$, and sequences $\{\delta_l\} \searrow 0$, $\{\varepsilon_l\} \searrow 0$ with $\varepsilon_0 \in (0,1)$. Set $l := 1$.

Step 0: Set $k := 0$, $x^0 := w^{l-1}$, $u(x^0) := u^0 := v^{l-1}$.

Step 1: Choose a pair of indices $i \in I_{\varepsilon_l}(x^k)$ and $j \in I$ such that

$$\langle f'(x^k), z^i - z^j \rangle \geq \delta_l, \tag{5.1}$$

set $\gamma_k := u^k_i$, $i_k := i$, $j_k := j$ and go to Step 2. Otherwise (i.e. if (5.1) does not hold for all $i \in I_{\varepsilon_l}(x^k)$ and $j \in I$) set $w^l := x^k$, $u(w^l) := v^l := u^k$, $l := l+1$ and go to Step 0. *(Restart)*

Step 2: Set $d^k := z^{j_k} - z^{i_k}$, determine m_k as the smallest number in $\mathbb{Z}_+$ such that

$$f(x^k + \theta^{m_k}\gamma_k d^k) \leq f(x^k) + \beta\theta^{m_k}\gamma_k \langle f'(x^k), d^k \rangle,$$

set $\lambda_k := \theta^{m_k}\gamma_k$, $x^{k+1} := x^k + \lambda_k d^k$,

$$u_s(x^{k+1}) := u^{k+1}_s := \begin{cases} u^k_s - \lambda_k & \text{if } s = i_k, \\ u^k_s + \lambda_k & \text{if } s = j_k, \\ u^k_s & \text{otherwise}; \end{cases} \tag{5.2}$$

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

Observe that

$$I_{\varepsilon_l}(x^k) = \{i \in I : u^k_i \geq \varepsilon_l\}$$

and that we can simply set $\gamma_k := u^k_i$ in Step 1. Clearly, formula (5.2) gives the weight vector $u(x^{k+1})$ associated to x^{k+1} without solution of any system of equations. In fact,

$$\begin{aligned} x^{k+1} &= x^k + \lambda_k d^k = \sum_{i \in I} u^k_i z^i + \lambda_k (z^{j_k} - z^{i_k}) \\ &= \sum_{i \in I, i \neq i_k, j_k} u^k_i z^i + (u^k_{i_k} - \lambda_k) z^{i_k} + (u^k_{j_k} + \lambda_k) z^{j_k} = \sum_{i \in I} u^{k+1}_i z^i; \end{aligned}$$

in addition, we have

$$\sum_{i \in I} u^{k+1}_i = 1 \text{ and } u^{k+1}_i \geq 0, \ i \in I.$$

Therefore, each iterate changes only two components of the current weight vector. Clearly, it suffices to keep only positive components of this vector. Set

$$I_+(x^k) = \{i \in I : u_i^k > 0\},$$

then the number of indices in $I_+(x^k)$ is much more smaller than that in I. For instance, any segment $[a,b]$ in $\mathbb{R}^m$ has 2^m vertices, whereas any point $x \in [a,b]$ can be represented by $m+1$ vertices, i.e., for this weight vector, set $I_+(x)$ contains $m+1$ items.

The other way to implementation consists in calculation the necessary weights from the iteration point x^k. Both the approaches coincide if each point $x \in D$ has the unique weight vector $u = u(x)$. This is the case for the simplices. In fact, take

$$D = \left\{x \in \mathbb{R}^m_+ : \langle a, x\rangle = \tau\right\},$$

τ is a fixed positive number, a is a fixed vector with positive coordinates, $\mathbb{R}^m_+$ denotes the non-negative orthant in $\mathbb{R}^m$. Given $x \in D$, set

$$\sigma(x) = \sum_{i=1}^{m} x_i,$$

then

$$z_s^i = \begin{cases} \tau/a_s & \text{if } s = i, \\ 0 & \text{otherwise;} \end{cases}$$

for $i = 1, \dots, n$ and

$$u_s = \begin{cases} x_s/\sigma(x) & \text{if } s = i, \\ 0 & \text{otherwise.} \end{cases}$$

However, the second way may be useful if the weight vector $u(x)$ is not defined uniquely. Moreover, we can evaluate the weight implicitly by using Proposition 2.2.

Next, the current condition (3.1) (or (5.2)) for selection of the pair of indices i_k and j_k can be implemented within various rules. It seems suitable to find z^{i_k} as an approximate solution of problem (1.2) and j_k as an approximate solution of the problem

$$\max_{s \in I_{\varepsilon_l}(x^k)} \to \langle f'(x^k), z^s\rangle. \tag{5.3}$$

That is, we can make several steps of any algorithm toward the solutions of (1.2) and (5.3) for satisfying (3.1). We can even solve (1.2) exactly, and then check the indices from $I_+(x^k)$ sequentially. This procedure does not seem too difficult since the number of indices in $I_+(x^k)$ is much more smaller than that in I.

We should observe that all these implementations of (PVM) are closely related with the so-called “atomic” or weighting representation (2.2) of the feasible set D. The usual conditional gradient method and its version with away steps can utilize the standard definition of D, whereas their “pure” weighting versions are also rather popular; see e.g. [7, 12, 13]. It should be also noticed that all the weighting versions of the methods including (PVM) can be in principle applied to problem (1.1) where the feasible set D is represented as

$$D = \left\{ x : x = \sum_{i\in I} u_i z^i, \ \sum_{i\in I} u_i = 1, \ u_i \geq 0, \ z^i \in H, \ i \in I \right\},$$

where H is some Hilbert space and the index set $I = \{1,\dots,n\}$ is finite. This is treated as linear variable transformation, i.e. $x = Tu$ for some linear mapping $T : \mathbb{R}^n \to H$, which transforms the standard simplex in $\mathbb{R}^n$ into D. In turn, the goal function $f(x)$ is also replaced with the function $\varphi(u) = f(Tu)$. Since the simplex is convex and compact, convergence of the method can be proved along the same lines.

It was mentioned in Section 1 that the main drawback of the usual conditional gradient method is its rather slow convergence. Incorporating the away steps, whose calculation requires the solution of the auxiliary problem

$$\max_{s\in I_+(x^k)} \to \langle f'(x^k), z^s \rangle, \tag{5.4}$$

enables one to attain the linear rate of convergence for some classes of optimization problems, but the computational experiments do not reveal this preference; see e.g. [8, 11, 13]. Instead of these steps we can utilize the so-called pairwise away or swap directions. Namely, let z^j and z^i be solutions of problems (1.2) and (5.4), respectively. Then, we can take $d^k = z^j - z^i$ as the descent direction at the k-th iteration; see [11]. It should be noted that the method based on the same pairwise directions was first suggested in [18] for network equilibrium problems. In [19], a similar method was suggested for general smooth optimization problems with simplex type constraints. These marginal based index choice methods became very popular after appearance of their big data applications; see e.g. [20] for more details and references. It was also mentioned in Section 1 that (PVM) can be viewed as an extension of the bi-coordinate descent method (BCV) with special threshold control proposed in [14] for optimization problems with simplex constraints. That is, (PVM) can be applied for optimization problems with arbitrary affine constraints due to the utilization of the weight vectors which is treated as variable transformation. In comparison with the marginal swap direction strategy, (PVM) does not insist on solutions of auxiliary problems of form (1.2) and (5.4), which enables us to reduce the computational expenses significantly. Nevertheless, (PVM) maintains the useful sparse iteration point property, as all the mentioned conditional gradient methods.

6. Computational experiments

In order to check the performance of (PVM) we carried out computational experiments. We took also the usual conditional gradient method (CGM), the marginal-based swap direction descent method (MDM), with the same Armijo linesearch and compared them with (PVM). They were implemented in Delphi with double precision arithmetic. The main goal was to compare the numbers of iterations (it) and calculations of partial derivatives of f (calc) for attaining the same accuracy $\delta' = 0.1$. We took the following accuracy measure:

$$\Delta_k = \max_{y\in D} \langle f'(x^k), x^k - y \rangle.$$

We chose $\beta = \theta = 0.5$ for the methods, and the rule $\delta_{l+1} = \nu\delta_l$, $\varepsilon_{l+1} = \nu\varepsilon_l$ with $\nu = 0.5$ for (PVM).

TABLE 1. Starting point x', quadratic cost function

	(CGM)	(MDM)	(PVM)
	it / calc	it / calc	it / calc
$m = 5$	202 / 1010	11 / 55	11 / 53
$m = 10$	at 500 / 5000 $\Delta_k = 0.25$	34 / 340	37 / 279
$m = 20$	at 500 / 10000 $\Delta_k = 0.11$	49 / 980	50 / 703
$m = 50$	at 500 / 25000 $\Delta_k = 0.39$	87 / 4350	108 / 3574
$m = 100$	at 500 / 50000 $\Delta_k = 0.62$	221 / 22100	267 / 17594

We first took the simplex as the feasible set, i.e.,

$$D = \left\{ x \in \mathbb{R}^m_+ : \sum_{i=1}^{m} x_i = \tau \right\}. \tag{6.1}$$

We took two starting points, namely, $x' = (\tau/m)e$ where e denote the vector of units in $\mathbb{R}^m$, and $x'' = \tau e^1$ where e^1 denote the first coordinate vector in $\mathbb{R}^m$. Also, we set $\tau = 10$.

In the first series, we took the quadratic cost function. We chose $f(x) = \varphi(x)$ where

$$\varphi(x) = 0.5\langle Px, x\rangle - \langle q, x\rangle, \tag{6.2}$$

the elements of the matrix P are defined by

$$p_{ij} = \begin{cases} \sin(i)\cos(j) & \text{if } i < j, \\ \sin(j)\cos(i) & \text{if } i > j, \\ \sum\limits_{i=1}^{m} |p_{ij}| + 1 & \text{if } i = j; \end{cases} \tag{6.3}$$

and $q = \mathbf{0}$. The results for the starting points x' and x'' are given in Tables 1 and 2, respectively.

In the second series, we took the convex cost function

$$f(x) = \varphi(x) + 1/(\langle c, x\rangle + \mu), \tag{6.4}$$

where the function φ was defined as above in (6.2)–(6.3), the elements of the vector c are defined by

$$c_i = 2 + \sin(i) \quad \text{for} \quad i = 1, \dots, m,$$

and $\mu = 5$. The results for the starting points x' and x'' are given in Tables 3 and 4, respectively.

TABLE 2. Starting point x'', quadratic cost function

	(CGM)	(MDM)	(PVM)
	it / calc	it / calc	it / calc
$m = 5$	47 / 235	14 / 70	17 / 74
$m = 10$	194 / 1940	37 / 370	42 / 307
$m = 20$	at 500 / 10000 $\Delta_k = 0.44$	124 / 2480	124 / 1668
$m = 50$	at 500 / 25000 $\Delta_k = 1.22$	326 / 16300	211 / 7046
$m = 100$	at 500 / 50000 $\Delta_k = 2.93$	at 500 / 50000 $\Delta_k = 0.31$	399 / 25213

TABLE 3. Starting point x', convex cost function

	(CGM)	(MDM)	(PVM)
	it / calc	it / calc	it / calc
$m = 5$	203 / 1015	11 / 55	11 / 53
$m = 10$	at 500 / 5000 $\Delta_k = 0.21$	34 / 340	38 / 287
$m = 20$	491 / 9820	53 / 1060	46 / 666
$m = 50$	at 500 / 25000 $\Delta_k = 0.41$	83 / 4150	107 / 3427
$m = 100$	at 500 / 50000 $\Delta_k = 0.61$	211 / 21100	267 / 17012

Next, we took the more general feasible set instead of (6.1):

$$D = \left\{ x \in \mathbb{R}^m_+ : \sum_{i=1}^{m} a_i x_i = \tau \right\}.$$

the elements of the vector a were defined by

$$a_i = 1.5 + \sin(i) \quad \text{for} \quad i = 1, \dots, m,$$

and fixed $\tau = 10$. We took only the starting point $x'' = (\tau/a_1)e^1$.

TABLE 4. Starting point x'', convex cost function

	(CGM)	(MDM)	(PVM)
	it / calc	it / calc	it / calc
$m = 5$	44 / 220	14 / 70	15 / 67
$m = 10$	198 / 1980	37 / 370	43 / 312
$m = 20$	at 500 / 10000 $\Delta_k = 0.45$	114 / 2280	138 / 1839
$m = 50$	at 500 / 25000 $\Delta_k = 1.24$	319 / 15950	227 / 7354
$m = 100$	at 500 / 50000 $\Delta_k = 2.97$	at 500 / 50000 $\Delta_k = 0.33$	405 / 25758

TABLE 5. Quadratic cost function

	(CGM)	(MDM)	(PVM)
	it / calc	it / calc	it / calc
$m = 5$	20 / 100	9 / 45	11 / 48
$m = 10$	82 / 820	29 / 290	27 / 210
$m = 20$	199 / 3980	48 / 960	49 / 644
$m = 50$	at 500 / 25000 $\Delta_k = 0.21$	101 / 5050	119 / 3630
$m = 100$	at 500 / 50000 $\Delta_k = 0.62$	203 / 20300	286 / 17080

In the first series, we took the quadratic cost function from (6.2)–(6.3), the elements of the vector q were defined by

$$q_i = \sin(i)/i \;\text{ for }\; i = 1, \dots, m.$$

The results are given in Table 5.

In the second series, we took the convex cost function from (6.4) where the function φ was defined as above. The results are given in Table 6.

TABLE 6. Convex cost function

	(CGM)	(MDM)	(PVM)
	it / calc	it / calc	it / calc
$m = 5$	20 / 100	7 / 35	11 / 48
$m = 10$	79 / 790	27 / 270	25 / 189
$m = 20$	204 / 4080	49 / 980	51 / 677
$m = 50$	at 500 / 25000 $\Delta_k = 0.19$	100 / 5000	117 / 3618
$m = 100$	at 500 / 50000 $\Delta_k = 0.64$	210 / 21000	307 / 18468

In all the cases, (PVM) showed rather rapid convergence, it outperformed (MDM) in the number of total calculations if $m \geq 10$, besides, (PVM) and (MDM) appeared better essentially than (CGM).

7. CONCLUSIONS

We suggested a new class of descent methods for smooth optimization problems involving general affine constraints. The method is based on selective pairwise variations together with some threshold strategy. It keeps the convergence properties of the usual gradient ones together with reduction of the total computational expenses. Besides, it is suitable for large scale problems. The preliminary results of computational tests show rather rapid and stable convergence of the new method in comparison with the previous conditional gradient type methods.

Acknowledgments

This work was supported by the RFBR Grant, Project No. 16-01-00109a and by Grant No. 297689 from Academy of Finland.

REFERENCES

[1] M. Frank, P. Wolfe, An algorithm for quadratic programming, Naval. Res. Logist. Quart. 3 (1956), 95-110.

[2] E.S. Levitin, B.T. Polyak, Constrained minimization methods, USSR Comp. Maths. Math. Phys. 6 (1966), 1-50.

[3] V.F. Dem'yanov, A.M. Rubinov, Approximate Methods for Solving Extremum Problems, Leningrad Univ. Press, Leningrad, 1968. [Engl. transl. in Elsevier, Amsterdam, 1970]

[4] B.N. Pshenichnyi, Y.M. Danilin, Numerical Methods in Extremal Problems, MIR, Moscow, 1978.

[5] J.C. Dunn, Convergence rates for conditional gradient sequences generated by implicit step length rules, SIAM J. Control Optim. 18 (1980), 473-487.

[6] K.L. Clarkson, Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm, ACM Tran. Algorithms 6 (2010), Article ID 63.

[7] M. Jaggi, Revisiting Frank-Wolfe: Projection-free sparse convex optimization, Proc. of the 30th International Conference on Machine Learning (ICML-13), (2013), 427-435.

[8] J. Guelat, P. Marcotte, Some comments on Wolfe's "away step", Math. Program. 35 (1986), 110-119.

[9] A. Beck, M. Teboulle, A conditional gradient method with linear rate of convergence for solving convex linear systems, Math. Meth. Oper. Res. 59 (2004), 235-247.

[10] R.M. Freund, P. Grigas, New analysis and results for the Frank-Wolfe method, Math. Program. 155 (2016), 199-230.

[11] R. Nanculef, E. Frandi, C. Sartori, H. Allende, A novel Frank-Wolfe algorithm. Analysis and applications to large-scale SVM training, Inform. Sci. 285 (2014), 66-99.

[12] S. Lacoste-Julien, M. Jaggi, On the global linear convergence of Frank-Wolfe optimization variants, Proc. of the 28th International Conference on Neural Information Processing Systems (NIPS15), (2015), 496-504.

[13] A. Beck, S. Shtern, Linearly convergent away-step conditional gradient for non-strongly convex functions, Math. Progam. (2016). doi:10.1007/s10107-016-1069-4.

[14] I.V. Konnov, Selective bi-coordinate variations for resource allocation type problems, Comput. Optim. Appl. 64 (2016), 821-842.

[15] I.V. Konnov, Equilibrium Models and Variational Inequalities, Elsevier, Amsterdam, 2007.

[16] D.B. Yudin, E.G. Gol'shtein, Linear Programming, Nauka, Moscow, 1969. (in Russian)

[17] I.V. Konnov, Nonlinear Optimization and Variational Inequalities, Kazan Univ. Press, Kazan, 2013. (in Russian).

[18] S.C. Dafermos, F.T. Sparrow, The traffic assignment problem for a general network, J. Res. National Bureau Stand. 73B (1969), 91-118.

[19] G.M. Korpelevich, Coordinate descent method for constrained minimization problems, linear inequalities, and matrix games, In: Gol'shtein, E.G. (ed.) Mathematical Methods for Solving Economic Problems, vol.9, pp. 84-97, Nauka, Moscow, 1980. (in Russian)

[20] A. Beck, The 2-coordinate descent method for solving double-sided simplex constrained minimization problems, J. Optim. Theory Appl. 162 (2014), 892-919.