Sequential Threshold Control in Descent Splitting Methods for Decomposable Optimization Problems[1]

I.V. Konnov[2]

Abstract

We suggest a modification of the descent splitting methods for decomposable composite optimization problems, which maintains the basic convergence properties, but enables one to reduce the computational expenses per iteration and to provide computations in a distributed manner. It consists in making coordinate-wise steps together with a special threshold control.

Key words: Composite optimization; decomposable problems; descent splitting methods; projection methods; coordinate-wise steps; threshold control.

1 Introduction

The general optimization problem consists in finding the minimal value of some goal function $\mu : \mathbb{R}^N \to \mathbb{R}$ on a feasible set $X \subseteq \mathbb{R}^N$. For brevity, we write this problem as

$$\min_{\mathbf{x}\in X} \to \mu(\mathbf{x}), \tag{1}$$

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

$$\mu^* = \inf_{\mathbf{x}\in X} \mu(\mathbf{x}).$$

It is well known that solution methods that take into account peculiarities of particular problems show better computational results than general purpose oriented ones. For this reason, there exists a necessity to develop new versions and modifications, which are adjusted for certain applied problems, despite the wide variety of existing optimization methods. For instance, a great number of applications reduce to problem (1), where

$$\mu(\mathbf{x}) = f(\mathbf{x}) + h(\mathbf{x}), \tag{2}$$

where $f : \mathbb{R}^N \to \mathbb{R}$ is a smooth, but not necessary convex function and $h : \mathbb{R}^N \to \mathbb{R}$ is not necessary smooth, but rather simple and convex function. That is, we obtain a

[1]This work was supported by grant No. 276064 from Academy of Finland and by the RFBR grant, project No. 13-01-00029a.

[2]Department of System Analysis and Information Technologies, Kazan Federal University, ul. Kremlevskaya, 18, Kazan 420008, Russia.

non-convex and non-differentiable optimization problem, which appears too difficult for solution with usual subgradient type methods, especially, if it has high dimensionality. However, the "simplicity" of f suggests a different treatment of these functions during the computational process and leads to the (forward-backward) splitting iteration:

$$\begin{array}{c}\langle f'(\mathbf{x}^k) + \alpha^{-1}(\mathbf{x}^{k+1} - \mathbf{x}^k), \mathbf{y} - \mathbf{x}^{k+1}\rangle \\ +h(\mathbf{y}) - h(\mathbf{x}^{k+1}) \geq 0 \quad \forall \mathbf{y} \in X,\end{array} \tag{3}$$

where $\alpha > 0$ is a stepsize parameter. Clearly, (3) coincides with the (explicit) projection method if $h \equiv 0$ and with the (implicit) proximal method if $f \equiv 0$. Observe also that (3) has always a unique solution because its cost function is strongly convex. After inserting a suitable line-search procedure, we obtain a descent splitting method (see [1]), which provides convergence to stationary points under rather mild assumptions. The usefulness of this approach becomes clear if (1)–(2) is decomposable. For instance, let

$$h(x) = \sum_{i=1}^{N} h_i(x_i)$$

and $X = X_1 \times \ldots \times X_N$, $X_i \subseteq \mathbb{R}$ for $i = 1, \ldots, N$. Then (3) becomes equivalent to n independent one-dimensional problems of the form

$$\min_{x_i \in X_i} \rightarrow \{x_i g_i(\mathbf{x}^k) + (2\alpha)^{-1}(x_i - x_i^k)^2 + h_i(x_i)\}, \tag{4}$$

where $g_i(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial x_i}$. Further development of this class of methods can be found e.g. in [2, 3, 4, 5].

Rather recently, decomposable optimization problems of form (1)–(2) were paid significant attention due to their big data applications; see e.g. [3, 4, 6, 5] and the references therein. However, these problems can have huge dimensionality, besides, the proper methods should be adjusted to possible distributed computational process. For this reason, even solution of all the one-dimensional problems of form (4) may appear too expensive.

The main goal of this paper is to suggest a modification of descent splitting methods for decomposable composite optimization problems of form (1)–(2), which maintains the basic convergence properties, but enables one to reduce the computational expenses per iteration and to provide computations in a distributed (multi-agent) manner. It consists in making coordinate-wise steps together with a special threshold control procedure. This procedure was proposed in [7] for bi-coordinate descent smooth optimization methods. We observe that it is rather usual for general non-differentiable optimization methods (see e.g. [8]), and was suggested for decomposable variational inequalities in [9, 10, 11].

A few words about our notation. As usual, we denote by $\mathbb{R}^s$ the real s-dimensional Euclidean space, all elements of such spaces being column vectors represented by a lower case Roman alphabet in boldface, e.g. $\mathbf{x}$. We use superscripts to denote different

vectors, and subscripts to denote different scalars or components of vectors. For any vectors $\mathbf{x}$ and $\mathbf{y}$ of $\mathbb{R}^s$, we denote by $\langle \mathbf{x}, \mathbf{y} \rangle$ their scalar product, i.e.,

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_{i=1}^{s} x_i y_i,$$

and by $\|\mathbf{x}\|$ the Euclidean norm of $\mathbf{x}$, i.e., $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

2 Problem formulation and preliminary properties

Let us consider a partitionable optimization problem of form (1)–(2). That is, set $\mathcal{N} = \{1, \ldots, N\}$ and suppose that there exists a partition

$$\mathcal{N} = \sum_{i=1}^{n} \mathcal{N}_i$$

with $|\mathcal{N}_i| = N_i$, $N = \sum\limits_{i=1}^{n} N_i$, and $\mathcal{N}_i \bigcap \mathcal{N}_j = \varnothing$ if $i \neq j$ such that

$$X = X_1 \times \ldots \times X_n = \prod_{i=1}^{n} X_i, \tag{5}$$

where X_i is a non-empty, convex, and closed set in $\mathbb{R}^{N_i}$ for $i = 1, \ldots, n$. Then, any point $\mathbf{x} = (x_1, \ldots, x_N)^\top \in \mathbb{R}^N$ is represented by $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ where $\mathbf{x}_i = (x_j)_{j \in \mathcal{N}_i} \in \mathbb{R}^{N_i}$ for $i = 1, \ldots, n$. For brevity, set

$$(\mathbf{x}_{-i}, \mathbf{y}_i) = (\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{y}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n).$$

Also, we suppose that

$$h(\mathbf{x}) = \sum_{i=1}^{n} h_i(\mathbf{x}_i), \tag{6}$$

where $h_i : X_i \to \mathbb{R}$ is convex and has the non-empty subdifferential $\partial h_i(\mathbf{x}_i)$ at each point $\mathbf{x}_i \in X_i$, for $i = 1, \ldots, n$. Then each function h_i is lower semi-continuous on X_i, the function h is lower semi-continuous on X, and

$$\partial h(\mathbf{x}) = \partial h_1(\mathbf{x}_1) \times \ldots \times \partial h_n(\mathbf{x}_n), \quad \forall \mathbf{x} \in X.$$

So, our problem (1)–(2), (5)–(6) is rewritten as

$$\min_{\mathbf{x} \in X_1 \times \ldots \times X_n} \to \mu(\mathbf{x}) = \left\{ f(\mathbf{x}) + \sum_{i=1}^{n} h_i(\mathbf{x}_i) \right\}. \tag{7}$$

As before, we suppose that the function $f : \mathbb{R}^N \to \mathbb{R}$ is smooth, but not necessary convex. Set $\mathbf{g}(\mathbf{x}) = f'(\mathbf{x})$, then

$$\mathbf{g}(\mathbf{x}) = (\mathbf{g}_1(\mathbf{x}), \ldots, \mathbf{g}_n(\mathbf{x}))^\top, \text{ where } \mathbf{g}_i(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_j}\right)_{j \in \mathcal{N}_i} \in \mathbb{R}^{N_i}, \ i = 1, \ldots, n.$$

The simplest case where $n_i = 1$ for all $i = 1, \ldots, n$ and $n = N$ corresponds to the scalar coordinate partition; cf. (4). From the assumptions above it follows that the function μ is directionally differentiable at each point $\mathbf{x} \in X$, that is, its directional derivative with respect to any vector $\mathbf{d}$ is defined by the formula:

$$\mu'(\mathbf{x}; \mathbf{d}) = \langle \mathbf{g}(\mathbf{x}), \mathbf{d} \rangle + h'(\mathbf{x}; \mathbf{d}), \text{ with } h'(\mathbf{x}; \mathbf{d}) = \sum_{i=1}^{n} \max_{\mathbf{b}_i \in \partial h_i(\mathbf{x}_i)} \langle \mathbf{b}_i, \mathbf{d}_i \rangle; \tag{8}$$

see e.g. [12].

We recall that a function $\varphi : \mathbb{R}^s \to \mathbb{R}$ is said to be *coercive* on a set $D \subset \mathbb{R}^s$ if $\{\varphi(\mathbf{u}^k)\} \to +\infty$ for any sequence $\{\mathbf{u}^k\} \subset D$, $\|\mathbf{u}^k\| \to \infty$. We will in addition suppose that the function $\mu : \mathbb{R}^N \to \mathbb{R}$ is coercive on X, then problem (1)–(2), (5)–(6) (or (7)) has a solution.

We start our considerations from the optimality condition.

Proposition 2.1 *(a) Each solution of problem (7) is a solution of the mixed variational inequality (MVI for short): Find a point $x^* \in X = X_1 \times \ldots \times X_n$ such that*

$$\begin{gathered} \sum_{i=1}^{n} \langle \mathbf{g}_i(\mathbf{x}^*), \mathbf{y}_i - \mathbf{x}_i^* \rangle + \sum_{i=1}^{n} [h_i(\mathbf{y}_i) - h_i(\mathbf{x}_i^*)] \geq 0 \\ \forall \mathbf{y}_i \in X_i, \quad \textit{for } i = 1, \ldots, n. \end{gathered} \tag{9}$$

(b) If f is convex, then each solution of MVI (9) solves problem (7).

Proof. If $\mathbf{x}^*$ solves MVI (9), then, by convexity, we have

$$f(\mathbf{y}) - f(\mathbf{x}^*) + h(\mathbf{y}) - h(\mathbf{x}^*) \geq \langle \mathbf{g}(\mathbf{x}^*), \mathbf{y} - \mathbf{x}^* \rangle + h(\mathbf{y}) - h(\mathbf{x}^*) \geq 0$$

for every $\mathbf{y} \in X$, i.e. $\mathbf{x}^*$ solves (7). Conversely, let $\mathbf{x}^*$ solve problem (7). If $\mathbf{x}^*$ does not solve MVI (9), there is a point $\mathbf{x}' \in X$ such that

$$\langle \mathbf{g}(\mathbf{x}^*), \mathbf{x}' - \mathbf{x}^* \rangle + h(\mathbf{x}') - h(\mathbf{x}^*) = \delta < 0.$$

Take $\lambda > 0$ and set $\mathbf{x}(\lambda) = \lambda \mathbf{x}' + (1 - \lambda)\mathbf{x}^*$. Then $\mathbf{x}(\lambda) \in X$ if $\lambda \in (0, 1)$. At the same time, we have

$$\begin{aligned} &\langle \mathbf{g}(\mathbf{x}^*), \mathbf{x}(\lambda) - \mathbf{x}^* \rangle + h(\mathbf{x}(\lambda)) - h(\mathbf{x}^*) \\ &\leq \lambda \langle \mathbf{g}(\mathbf{x}^*), \mathbf{x}' - \mathbf{x}^* \rangle + \lambda h(\mathbf{x}') + (1 - \lambda) h(\mathbf{x}^*) - h(\mathbf{x}^*) \\ &= \lambda \left\{ \langle \mathbf{g}(\mathbf{x}^*), \mathbf{x}' - \mathbf{x}^* \rangle + h(\mathbf{x}') - h(\mathbf{x}^*) \right\} = \lambda \delta < 0. \end{aligned}$$

It follows that

$$\begin{aligned} & f(\mathbf{x}(\lambda)) - f(\mathbf{x}^*) + h(\mathbf{x}(\lambda)) - h(\mathbf{x}^*) \\ & = \langle \mathbf{g}(\mathbf{x}^*), \mathbf{x}(\lambda) - \mathbf{x}^* \rangle + o(\lambda) + h(\mathbf{x}(\lambda)) - h(\mathbf{x}^*) \\ & \le \lambda\delta + o(\lambda) < 0 \end{aligned}$$

for $\lambda \in (0,1)$ small enough, a contradiction. □

In what follows, we denote by X^0 the solution set of MVI (9) and call it the set of *stationary points* of problem (7).

Fix $\alpha > 0$. For each point $\mathbf{x} \in X$ we can define $\mathbf{y}(\mathbf{x}) = (\mathbf{y}_1(\mathbf{x}), \dots, \mathbf{y}_n(\mathbf{x}))^\top \in X$ such that

$$\begin{gathered} \sum_{i=1}^n \langle \mathbf{g}_i(\mathbf{x}) + \alpha^{-1}(\mathbf{y}_i(\mathbf{x}) - \mathbf{x}_i), \mathbf{y}_i - \mathbf{y}_i(\mathbf{x}) \rangle + \sum_{i=1}^n [h_i(\mathbf{y}_i) - h_i(\mathbf{y}_i(\mathbf{x}))] \ge 0 \\ \forall \mathbf{y}_i \in X_i, \quad \text{for } i = 1, \dots, n. \end{gathered} \tag{10}$$

This MVI gives a necessary and sufficient optimality condition for the optimization problem:

$$\min_{\mathbf{y} \in X_1 \times \dots \times X_n} \to \sum_{i=1}^n \Phi_i(\mathbf{x}, \mathbf{y}_i), \tag{11}$$

where

$$\Phi_i(\mathbf{x}, \mathbf{y}_i) = \langle \mathbf{g}_i(\mathbf{x}), \mathbf{y}_i \rangle + 0.5\alpha^{-1}\|\mathbf{x}_i - \mathbf{y}_i\|^2 + h_i(\mathbf{y}_i) \tag{12}$$

for $i = 1, \dots, n$; cf. (4). Under the above assumptions each $\Phi_i(\mathbf{x}, \cdot)$ is strongly convex, hence problem (11)–(12) (or (10)) has the unique solution $\mathbf{y}(\mathbf{x})$, thus defining the single-valued mapping $\mathbf{x} \mapsto \mathbf{y}(\mathbf{x})$. Observe that all the components of $\mathbf{y}(\mathbf{x})$ can be found independently, i.e. (11)–(12) is equivalent to n independent optimization problems of the form

$$\min_{\mathbf{y}_i \in X_i} \to \Phi_i(\mathbf{x}, \mathbf{y}_i), \tag{13}$$

for $i = 1, \dots, n$ and $\mathbf{y}_i(\mathbf{x})$ just solves (13).

Proposition 2.2 *(a)* $\mathbf{x} = \mathbf{y}(\mathbf{x}) \Longleftrightarrow \mathbf{x} \in X^0$*;*

(b) The mapping $\mathbf{x} \mapsto \mathbf{y}(\mathbf{x})$ *is continuous on* X*;*

(c) For any point $\mathbf{u} \in X$ *it holds that*

$$\begin{aligned} & \langle \mathbf{g}(\mathbf{y}(\mathbf{x})), \mathbf{y}(\mathbf{x}) - \mathbf{u} \rangle + h(\mathbf{y}(\mathbf{x})) - h(\mathbf{u}) \\ & \le \langle \mathbf{g}(\mathbf{y}(\mathbf{x})) - \mathbf{g}(\mathbf{x}), \mathbf{y}(\mathbf{x}) - \mathbf{u} \rangle + \alpha^{-1} \langle \mathbf{y}(\mathbf{x}) - \mathbf{x}, \mathbf{u} - \mathbf{y}(\mathbf{x}) \rangle. \end{aligned} \tag{14}$$

Proof. If $\mathbf{x} = \mathbf{y}(\mathbf{x})$, then (10) implies $\mathbf{x} \in X^0$. Conversely, let $\mathbf{x}$ solve MVI (9), but $\mathbf{x} \ne \mathbf{y}(\mathbf{x})$. Then setting $\mathbf{y} = \mathbf{x}$ in (10) gives

$$\langle \mathbf{g}(\mathbf{x}), \mathbf{x} - \mathbf{y}(\mathbf{x}) \rangle + h(\mathbf{x}) - h(\mathbf{y}(\mathbf{x})) \ge \alpha^{-1}\|\mathbf{y}(\mathbf{x}) - \mathbf{x}\|^2 > 0,$$

which is a contradiction. Part (a) is true. To prove (b), take arbitrary $\mathbf{x}'$, $\mathbf{x}'' \in X$ and set $\mathbf{y}' = \mathbf{y}(\mathbf{x}')$ and $\mathbf{y}'' = \mathbf{y}(\mathbf{x}'')$ for brevity. Then from (10) it follows that

$$\langle \mathbf{g}(\mathbf{x}') + \alpha^{-1}(\mathbf{y}' - \mathbf{x}'), \mathbf{y}'' - \mathbf{y}' \rangle + h(\mathbf{y}'') - h(\mathbf{y}') \geq 0$$

and

$$\langle \mathbf{g}(\mathbf{x}'') + \alpha^{-1}(\mathbf{y}'' - \mathbf{x}''), \mathbf{y}' - \mathbf{y}'' \rangle + h(\mathbf{y}') - h(\mathbf{y}'') \geq 0.$$

Summing these inequalities gives

$$\langle \mathbf{g}(\mathbf{x}') - \mathbf{g}(\mathbf{x}'') - \alpha^{-1}(\mathbf{x}' - \mathbf{x}''), \mathbf{y}'' - \mathbf{y}' \rangle \geq \alpha^{-1} \|\mathbf{y}'' - \mathbf{y}'\|^2,$$

hence

$$\|\mathbf{g}(\mathbf{x}') - \mathbf{g}(\mathbf{x}'')\| + \alpha^{-1}\|\mathbf{x}' - \mathbf{x}''\| \geq \alpha^{-1}\|\mathbf{y}'' - \mathbf{y}'\|.$$

This means the mapping $\mathbf{x} \mapsto \mathbf{y}(\mathbf{x})$ is continuous and part (b) is true. To prove (c), we again use (10) and obtain

$$\begin{aligned}
&\langle \mathbf{g}(\mathbf{y}(\mathbf{x})), \mathbf{y}(\mathbf{x}) - \mathbf{u} \rangle + h(\mathbf{y}(\mathbf{x})) - h(\mathbf{u}) \\
&= \langle \mathbf{g}(\mathbf{y}(\mathbf{x})) - \mathbf{g}(\mathbf{x}) - \alpha^{-1}(\mathbf{y}(\mathbf{x}) - \mathbf{x}), \mathbf{y}(\mathbf{x}) - \mathbf{u} \rangle \\
&+ \langle \mathbf{g}(\mathbf{x}) + \alpha^{-1}(\mathbf{y}(\mathbf{x}) - \mathbf{x}), \mathbf{y}(\mathbf{x}) - \mathbf{u} \rangle + h(\mathbf{y}(\mathbf{x})) - h(\mathbf{u}) \\
&\leq \langle \mathbf{g}(\mathbf{y}(\mathbf{x})) - \mathbf{g}(\mathbf{x}) - \alpha^{-1}(\mathbf{y}(\mathbf{x}) - \mathbf{x}), \mathbf{y}(\mathbf{x}) - \mathbf{u} \rangle,
\end{aligned}$$

which gives (14). □

Set $\Delta(\mathbf{x}) = \|\mathbf{x} - \mathbf{y}(\mathbf{x})\|$ and $\Delta_i(\mathbf{x}) = \|\mathbf{x}_i - \mathbf{y}_i(\mathbf{x})\|$, then $\Delta^2(\mathbf{x}) = \sum_{i=1}^{n} \Delta_i^2(\mathbf{x})$. We see that the value $\Delta(\mathbf{x})$ can serve as accuracy measure at a point $\mathbf{x}$.

We establish now a useful descent property. Define for brevity $M = \{1, \ldots, n\}$.

Lemma 2.1 *Take any point $\mathbf{x} \in X$ and an index $i \in M$. If*

$$\mathbf{d}_s = \begin{cases} \mathbf{y}_i(\mathbf{x}) - \mathbf{x}_i & \text{if } s = i, \\ \mathbf{0} & \text{if } s \neq i; \end{cases}$$

then

$$\mu'(\mathbf{x}; \mathbf{d}) \leq -\alpha^{-1}\|\mathbf{y}_i(\mathbf{x}) - \mathbf{x}_i\|^2. \tag{15}$$

Proof. Due to the definition of $\mathbf{d}$ and (8), we have

$$\mu'(\mathbf{x}; \mathbf{d}) = \langle \mathbf{g}(\mathbf{x}), \mathbf{d} \rangle + h'(\mathbf{x}; \mathbf{d}) = \langle \mathbf{g}_i(\mathbf{x}), \mathbf{d}_i \rangle + \max_{\mathbf{b}_i \in \partial h_i(\mathbf{x}_i)} \langle \mathbf{b}_i, \mathbf{d}_i \rangle. \tag{16}$$

At the same time, (10) is equivalent to m independent problems of the form

$$\langle \mathbf{g}_i(\mathbf{x}) + \alpha^{-1}(\mathbf{y}_i(\mathbf{x}) - \mathbf{x}_i), \mathbf{y}_i - \mathbf{y}_i(\mathbf{x}) \rangle + h_i(\mathbf{y}_i) - h_i(\mathbf{y}_i(\mathbf{x})) \geq 0 \quad \forall \mathbf{y}_i \in X_i.$$

Setting $\mathbf{y}_i = \mathbf{x}_i$ here gives

$$\langle \mathbf{g}_i(\mathbf{x}), \mathbf{d}_i \rangle + h_i(\mathbf{y}_i(\mathbf{x})) - h_i(\mathbf{x}_i) \leq -\alpha^{-1}\|\mathbf{x}_i - \mathbf{y}_i\|^2. \tag{17}$$

By convexity, we have

$$\langle \mathbf{b}_i, \mathbf{d}_i \rangle \leq h_i(\mathbf{y}_i(\mathbf{x})) - h_i(\mathbf{x}_i)$$

for any $\mathbf{b}_i \in \partial h_i(\mathbf{x}_i)$. In view of (16), we now obtain (15). □

3 The descent splitting method with inexact line-search

Denote by $\mathbb{Z}_+$ the set of non-negative integers. The basic cycle of the descent splitting method with inexact line-search for MVI (9) is described as follows.

Basic cycle 1. Choose a point $\mathbf{x}^0 \in X$ and numbers $\alpha > 0$, $\delta > 0$, $\beta \in (0,1)$, $\theta \in (0,1)$.

At the k-th iteration, $k = 0, 1, \ldots$, we have a point $\mathbf{x}^k \in X$.

Step 1: Choose an index $i \in M$ such that $\Delta_i(\mathbf{x}^k) \geq \delta$, set $i_k = i$,

$$\mathbf{d}_s^k = \begin{cases} \mathbf{y}_s(\mathbf{x}^k) - \mathbf{x}_s^k & \text{if } s = i_k, \\ \mathbf{0} & \text{if } s \neq i_k; \end{cases}$$

and go to Step 3. Otherwise (i.e. when $\Delta_s(\mathbf{x}^k) < \delta$ for all $s \in M$) go to Step 2.

Step 2: Set $\mathbf{z} = \mathbf{x}^k$ and stop.

Step 3: Determine m as the smallest number in $\mathbb{Z}_+$ such that

$$\mu(\mathbf{x}^k + \theta^m \mathbf{d}^k) \leq \mu(\mathbf{x}^k) - \beta\alpha^{-1}\theta^m \Delta_i^2(\mathbf{x}^k), \tag{18}$$

set $\lambda_k = \theta^m$, $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda_k \mathbf{d}^k$, and $k = k + 1$. The iteration is complete.

Lemma 3.1 *The line-search procedure in Step 3 is always finite.*

Proof. If we suppose that the line-search procedure is infinite, then

$$\theta^{-m}(\mu(\mathbf{x}^k + \theta^m \mathbf{d}^k) - \mu(\mathbf{x}^k)) > -\beta\alpha^{-1}\Delta_i^2(\mathbf{x}^k),$$

for $m \to \infty$, hence, by taking the limit we have $\mu'(\mathbf{x}^k; \mathbf{d}^k) \geq -\beta\alpha^{-1}\Delta_i^2(\mathbf{x}^k)$, but Lemma 2.1 gives $\mu'(\mathbf{x}^k; \mathbf{d}^k) \leq -\alpha^{-1}\Delta_i^2(\mathbf{x}^k)$, hence $(1-\beta)\Delta_i^2(\mathbf{x}^k) \leq 0$, a contradiction. □

We obtain the main property of the basic cycle.

Proposition 3.1 *The number of iterations in Basic cycle 1 is finite.*

Proof. By construction, we have $-\infty < \mu^* \leq \mu(\mathbf{x}^k)$ and $\mu(\mathbf{x}^{k+1}) \leq \mu(\mathbf{x}^k) - \beta\alpha^{-1}\delta\lambda_k$, hence the sequence $\{x^k\}$ is bounded and has limit points, besides,

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

Suppose that the sequence $\{\mathbf{x}^k\}$ is infinite. Since the set M is finite, there is an index $i_k = i$, which is repeated infinitely. Take the corresponding subsequence $\{k_s\}$, then, without loss of generality, we can suppose that the subsequence $\{\mathbf{x}^{k_s}\}$ converges to a point $\bar{\mathbf{x}}$, besides, $\Delta_{i_{k_s}}(\mathbf{x}^{k_s}) = \|\mathbf{d}_i^{k_s}\|$, and we have

$$(\lambda_{k_s}/\theta)^{-1}(\mu(\mathbf{x}^{k_s} + (\lambda_{k_s}/\theta)\mathbf{d}^{k_s}) - \mu(\mathbf{x}^{k_s})) > -\beta\alpha^{-1}\|\mathbf{d}_i^{k_s}\|^2.$$

Using the mean value theorem (see e.g. [12, Theorem 2.3.7]), we obtain

$$\langle \mathbf{g}^{k_s} + \mathbf{t}^{k_s}, \mathbf{d}^{k_s} \rangle > -\beta\alpha^{-1}\|\mathbf{d}_i^{k_s}\|^2,$$

for some $\mathbf{g}^{k_s} = f'(\mathbf{x}^{k_s} + (\lambda_{k_s}/\theta)\xi_{k_s}\mathbf{d}^{k_s})$, $\mathbf{t}^{k_s} \in \partial h(\mathbf{x}^{k_s} + (\lambda_{k_s}/\theta)\xi_{k_s}\mathbf{d}^{k_s})$, $\xi_{k_s} \in (0,1)$. By taking the limit $s \to \infty$ we have

$$\langle f'(\bar{\mathbf{x}}) + \bar{\mathbf{t}}, \bar{\mathbf{d}} \rangle \geq -\beta\alpha^{-1}\|\bar{\mathbf{d}}_i\|^2,$$

for some $\bar{\mathbf{t}} \in \partial h(\bar{\mathbf{x}})$, where $\bar{\mathbf{d}} = \mathbf{y}(\bar{\mathbf{x}}) - \bar{\mathbf{x}}$ due to Proposition 2.2 (b). On the other hand, using Lemma 2.1 gives

$$\langle f'(\bar{\mathbf{x}}) + \bar{\mathbf{t}}, \bar{\mathbf{d}} \rangle \leq \mu'(\bar{\mathbf{x}}; \bar{\mathbf{d}}) \leq -\alpha^{-1}\|\bar{\mathbf{d}}_i\|^2,$$

hence $(1-\beta)\|\bar{\mathbf{d}}_i\|^2 \leq 0$, which implies $\bar{\mathbf{d}}_i = \mathbf{0}$. However, by construction, we have $\|\mathbf{d}_i^{k_s}\| \geq \delta$, hence $\|\bar{\mathbf{d}}_i\| \geq \delta > 0$, which is a contradiction. □

The whole method has two-level iteration scheme where each stage of the upper level invokes Basic cycle 1 with different parameters. Similar two-level schemes were applied in solution methods for decomposable VI's in [9, 10, 11].

Method (Upper level). Choose a point $\mathbf{z}^0 \in X$ and a sequence $\{\delta_l\} \searrow 0$.

At the l-th stage, $l = 1, 2, \ldots$, we have a point $\mathbf{z}^{l-1} \in X$ and a number δ_l. Apply Basic cycle 1 with $\mathbf{x}^0 = \mathbf{z}^{l-1}$, $\delta = \delta_l$ and obtain a point $\mathbf{z}^l = \mathbf{z}$ as its output.

Theorem 3.1 *The sequence $\{\mathbf{z}^l\}$ generated by the method with Basic cycle 1 has limit points, all these limit points are solutions of MVI (9). Besides, if f is convex, then*

$$\lim_{l\to\infty} \mu(\mathbf{z}^l) = \mu^*; \tag{19}$$

and all the limit points of $\{\mathbf{z}^l\}$ belong to X^*.

Proof. Following the proof of Proposition 3.1, we see that the sequence $\{\mathbf{z}^l\}$ is bounded and has limit points, besides, $\mu(\mathbf{z}^{l+1}) \leq \mu(\mathbf{z}^l)$, hence

$$\lim_{l\to\infty} \mu(\mathbf{z}^l) = \mu.$$

Take an arbitrary limit point $\bar{\mathbf{z}}$ of $\{\mathbf{z}^l\}$, then

$$\lim_{s\to\infty} \mathbf{z}^{l_s} = \bar{\mathbf{z}}.$$

For $l > 0$ we have

$$\Delta_i(\mathbf{z}^l) \leq \delta_l \text{ for all } i \in M,$$

hence $\Delta(\mathbf{z}^l) \leq \delta_l\sqrt{n}$. Due to Proposition 2.2 (b), taking the limit $l = l_s \to \infty$, we obtain $\Delta(\bar{\mathbf{z}}) = 0$ or $\mathbf{y}(\bar{\mathbf{z}}) = \bar{\mathbf{z}}$. Due to Proposition 2.2 (a), this means that the point $\bar{\mathbf{z}}$ solves MVI (9). Next, if f is convex, then by Proposition 2.1 (b), each limit point of $\{\mathbf{z}^l\}$ solves problem (7). It follows that $\mu = \mu^*$ and (19) holds. □

In case $h \equiv 0$, the method is a new decomposable version of the well known projection ones; see [13, 14]. Similarly, in case $h \equiv 0$ and $X = \mathbb{R}^N$, it differs from the known versions of the coordinate descent methods; see e.g. [15, 14].

4 Modifications

The above descent method admits various modifications and extensions. For instance, we can replace Step 2 in Basic cycle 1 with the following.

Step 2: Set $\mathbf{u} = \mathbf{x}^k$, $\mathbf{v} = \mathbf{y}(\mathbf{x}^k)$,

$$\mathbf{z} = \begin{cases} \mathbf{u} & \text{if } \mu(\mathbf{u}) \le \mu(\mathbf{v}), \\ \mathbf{v} & \text{if } \mu(\mathbf{v}) \le \mu(\mathbf{u}); \end{cases}$$

and stop.

We call this modification **Basic cycle 2.** Then the assertions of Lemma 3.1 and Proposition 3.1 clearly remain true for this version. This is also the case for Theorem 3.1. It suffices to observe that now $\Delta(\mathbf{u}^l) \le \delta_l\sqrt{n}$ and $\|\mathbf{z}^l - \mathbf{u}^l\| \le \delta_l\sqrt{n}$, where $\mathbf{u}^l = \mathbf{u}$ at the end of the l-th stage. Then the sequences $\{\mathbf{z}^l\}$ and $\{\mathbf{u}^l\}$ have the same limit points and the result follows.

Next, we can take the exact one-dimensional minimization rule instead of the current Armijo rule in (18). We call this modification **Basic cycle 1a** and **Basic cycle 2a**, respectively. The convergence then can be obtained along the same lines; see e.g. [16, Section 6.1]. However, it seems more valuable for parallel and distributed computational schemes to provide the step-size choice without calculation of the function value in all the variables; in contrast to (18). We can attain this goal by several means. First of all we note that (18) becomes decomposable and reduces to

$$f_i(\mathbf{x}_i^k + \theta^m \mathbf{d}_i^k) + h_i(\mathbf{x}_i^k + \theta^m \mathbf{d}_i^k) \le f_i(\mathbf{x}_i^k) + h_i(\mathbf{x}_i^k) - \beta\alpha^{-1}\theta^m \Delta_i^2(\mathbf{x}^k),$$

if f is separable, i.e.

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}_i).$$

However, we intend to give proper implementations in the non-separable case.

4.1 Step-size choice in the convex case

If the function f is convex, we can replace (18) with the following:

$$\langle \mathbf{g}_i(\mathbf{x}^k + \theta^m \mathbf{d}^k), \mathbf{d}_i^k \rangle + \theta^{-m} \left\{ h_i(\mathbf{x}_i^k + \theta^m \mathbf{d}_i^k) - h_i(\mathbf{x}_i^k) \right\} \le -\beta\alpha^{-1}\Delta_i^2(\mathbf{x}^k); \tag{20}$$

see e.g. [17, p.198] and [16, p.297]. Note that the trial point $\mathbf{x}^k + \theta^m \mathbf{d}^k$ has the shift from $\mathbf{x}^k$ only in $\mathbf{d}_i^k$, hence it can be implemented independently of other variables. This modification seems also useful if the computation of partial derivatives is not so expensive in comparison with that of the function f. From (20) it now follows that

$$\begin{aligned} &\mu(\mathbf{x}^k + \theta^m \mathbf{d}^k) - \mu(\mathbf{x}^k) = f(\mathbf{x}^k + \theta^m \mathbf{d}^k) - f(\mathbf{x}^k) + h_i(\mathbf{x}_i^k + \theta^m \mathbf{d}_i^k) - h_i(\mathbf{x}_i^k) \\ &\le \theta^m \langle \mathbf{g}_i(\mathbf{x}^k + \theta^m \mathbf{d}^k), \mathbf{d}_i^k \rangle + h_i(\mathbf{x}_i^k + \theta^m \mathbf{d}_i^k) - h_i(\mathbf{x}_i^k) \\ &\le -\beta\theta^m \alpha^{-1} \Delta_i^2(\mathbf{x}^k), \end{aligned}$$

and (18) holds true. We call this modification **Basic cycle 1b** and **Basic cycle 2b**, respectively. In order to justify this version, we show that Lemma 3.1 remains true. In fact, if we suppose that the line-search procedure is infinite, then

$$\langle \mathbf{g}_i(\mathbf{x}^k + \theta^m \mathbf{d}^k), \mathbf{d}_i^k \rangle + \theta^{-m} \left\{ h_i(\mathbf{x}_i^k + \theta^m \mathbf{d}_i^k) - h_i(\mathbf{x}_i^k) \right\} > -\beta \alpha^{-1} \Delta_i^2(\mathbf{x}^k);,$$

for $m \to \infty$, hence, by taking the limit we have $\mu'(\mathbf{x}^k; \mathbf{d}^k) \geq -\beta \alpha^{-1} \Delta_i^2(\mathbf{x}^k)$, but Lemma 2.1 gives $\mu'(\mathbf{x}^k; \mathbf{d}^k) \leq -\alpha^{-1} \Delta_i^2(\mathbf{x}^k)$, hence $(1-\beta)\Delta_i^2(\mathbf{x}^k) \leq 0$, a contradiction.

The proof of Proposition 3.1 follows the same lines and is omitted. Of course, the assertion of Theorem 3.1 remains true for this version.

4.2 Step-size choice in the Lipschitz gradient case

If the gradient of the function f is Lipschitz continuous with some constant $L > 0$, i.e.,

$$\|f'(\mathbf{y}) - f'(\mathbf{x})\| \leq L \|\mathbf{y} - \mathbf{x}\|$$

for any vectors $\mathbf{x}$ and $\mathbf{y}$, we can give an explicit lower bound for the stepsize. However, it seems more suitable to utilize partial Lipschitz continuity conditions of the form

$$\|\mathbf{g}_i(\mathbf{x} + \mathbf{d}^{(i)}) - \mathbf{g}_i(\mathbf{x})\| \leq L_i \|\mathbf{d}^{(i)}\| = L_i \|\mathbf{d}_i\|$$

for any vector $\mathbf{x}$, where

$$\mathbf{d}_s^{(i)} = \begin{cases} \mathbf{d}_i & \text{if } s = i, \\ \mathbf{0} & \text{if } s \neq i; \end{cases}$$

for $i \in M$ and any vector $\mathbf{d} = (\mathbf{d}_1, \ldots, \mathbf{d}_n)^\top \in \mathbb{R}^N$. Then, clearly, $L_i \leq L$ for each $i \in M$. We recall the useful property of the functions having the Lipschitz continuous gradient

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

see [18, Lemma 1.2]. Similarly, for any vectors $\mathbf{x}$ and $\mathbf{d}$, we have

$$f(\mathbf{x} + \mathbf{d}^{(i)}) \leq f(\mathbf{x}) + \langle \mathbf{g}_i(\mathbf{x}), \mathbf{d}_i \rangle + 0.5 L_i \|\mathbf{d}_i\|^2 \quad \forall i \in M. \tag{21}$$

If $\mathbf{d}_i = \mathbf{y}_i(\mathbf{x}) - \mathbf{x}_i$, then (17) and (21) give

$$\begin{aligned} &\mu(\mathbf{x} + \lambda \mathbf{d}^{(i)}) - \mu(\mathbf{x}) = f(\mathbf{x} + \lambda \mathbf{d}^{(i)}) - f(\mathbf{x}) + h_i(\mathbf{x}_i + \lambda \mathbf{d}_i) - h_i(\mathbf{x}_i) \\ &\leq \lambda \left\{ \langle \mathbf{g}_i(\mathbf{x}), \mathbf{d}_i \rangle + h_i(\mathbf{y}_i(\mathbf{x})) - h_i(\mathbf{x}_i) \right\} + 0.5 L_i \lambda^2 \|\mathbf{d}_i\|^2 \\ &\leq -\lambda \alpha^{-1} \Delta_i^2(\mathbf{x}) + 0.5 L_i \lambda^2 \Delta_i^2(\mathbf{x}) = -\lambda(\alpha^{-1} - 0.5 L_i \lambda) \Delta_i^2(\mathbf{x}) \\ &\leq -\beta \lambda \alpha^{-1} \Delta_i^2(\mathbf{x}), \end{aligned}$$

if $\lambda \leq \bar{\lambda}_{(i)} = 2(1-\beta)/L_i$. It follows that (18) holds with $\lambda_k \geq \min\{1, \theta \bar{\lambda}_{(i_k)}\} \geq \gamma > 0$.

Besides, knowing some evaluation of the partial Lipschitz constants, we can simply take the restricted stepsize values $\lambda_k \in [\lambda', \bar{\lambda}_{(i_k)}]$ with $\lambda' > 0$ at Step 3, which reduces

the computational expenses essentially and admits independent or parallel implementation. Then calculations of the goal function values are not necessary. We call this modification **Basic cycle 1c** and **Basic cycle 2c**, respectively. Obviously, the assertions of Proposition 3.1 and Theorem 3.1 remain true for this version. Observe, that the smaller is the partial Lipschitz constant, the longer step can be made. Furthermore, we can take a collection $\mathbf{a} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$ with positive entries and solve the partial auxiliary optimization problems of form (11)–(12) with different $\alpha = \alpha_i$, thus obtaining vectors $\mathbf{y}_i(\mathbf{x}^k)$, with verifying the condition $\alpha_i^{-1}\Delta_i(\mathbf{x}^k) \geq \delta$. Then we can again remove the line-search procedure in Step 3 and simply set

$$\mathbf{x}_s^{k+1} = \begin{cases} \mathbf{y}_s(\mathbf{x}^k) & \text{if } s = i_k, \\ \mathbf{x}_s^k & \text{if } s \neq i_k. \end{cases}$$

It is easy to see that fixed values of α_i, e.g. $\alpha_i = 1/L_i$, will provide the same convergence properties.

Moreover, we can make modifications in some other direction. Namely, at Step 1 of the basic cycle we can take an arbitrary number of indices $i \in M_k$, such that $\sum\limits_{i \in M_k} \Delta_i(\mathbf{x}^k) \geq \delta$ and define

$$\mathbf{d}_s^k = \begin{cases} \mathbf{y}_s(\mathbf{x}^k) - \mathbf{x}_s^k & \text{if } s \in M_k, \\ \mathbf{0} & \text{if } s \notin M_k. \end{cases}$$

Clearly, this modification will keep all the convergence properties with possible acceleration if necessary. The order of verification of the indices has no influence for the theory, but may be very significant for implementation. We can take for example cyclical or certain learning verification strategies. Similarly, we can implement some adaptive strategy to avoid calculation of all the partial derivatives even in the restart situation, which implies Step 2. For instance, take two threshold numbers t' and t'', $t' \ll t''$. If we have an index $i \in M$ such that $\Delta_i(\mathbf{x}^k) \geq \delta$ after $t < t'$ trials, we can increase δ, otherwise, if we have $\Delta_i(\mathbf{x}^k) < \delta$ after $t > t''$ trials, we can decrease δ, etc.

These opportunities make the method very flexible and suitable for parallel and distributed computational schemes in the case of decomposable high-dimensional optimization problems; see e.g. [14, 2]. A great number of coordinate-wise methods were developed for various big data optimization problems, which just have the form (1)–(2), i.e. involve smooth and non-smooth functions; see e.g. [3, 4, 6] and the references therein. A method based on the splitting type auxiliary problem of form (11)–(12) was proposed in [5]. However, it requires calculation of all the partial derivatives at each iteration, because it is based on the computation of the marginal value $\max_{i \in M} \Delta_i(\mathbf{x}^k)$, because the basic index i_k is either its exact maximizer or attains it with certain accuracy. Our approach does not require such a computation. So, the coordinate-wise steps in our method can be implemented independently. Therefore, the method appears also suitable for multi-agent applications; see e.g. [19].

5 Convergence rates

We recall the basic assumptions on the main problem (7). We suppose that the function $f : \mathbb{R}^N \to \mathbb{R}$ is smooth, but not necessary convex, X_i is a non-empty, convex, and closed set in $\mathbb{R}^{N_i}$ for $i = 1, \ldots, N$, and $h_i : X_i \to \mathbb{R}$ is convex and has the non-empty subdifferential $\partial h_i(\mathbf{x}_i)$ at each point $\mathbf{x}_i \in X_i$, for $i = 1, \ldots, n$. We also suppose that the cost function μ is coercive on X.

We take the method with Basic cycle 1 and establish its finite termination under the following *sharp solution condition*; see [20, Section 2.2].

There exists a number $\tau > 0$ such that, for each point $\mathbf{v} \in X$, it holds that

$$\langle \mathbf{g}(\mathbf{v}), \mathbf{v} - \pi_0(\mathbf{v}) \rangle + h(\mathbf{v}) - h(\pi_0(\mathbf{v})) \geq \tau \|\mathbf{v} - \pi_0(\mathbf{v})\|,$$

where $\pi_0(\mathbf{v})$ denotes the projection of a point $\mathbf{v}$ onto X^0.

Theorem 5.1 *Let a sequence $\{\mathbf{z}^l\}$ generated by the method with Basic cycle 1. Suppose that the sharp solution condition holds. Then the method terminates with a point of X^0.*

Proof. From Proposition 2.2 (c) with $\mathbf{u} = \pi_0(\mathbf{y}(\mathbf{x}))$ we now have

$$\begin{aligned} \tau \|\mathbf{y}(\mathbf{x}) - \mathbf{u}\| &\leq \langle \mathbf{g}(\mathbf{y}(\mathbf{x})), \mathbf{y}(\mathbf{x}) - \mathbf{u} \rangle + h(\mathbf{y}(\mathbf{x})) - h(\mathbf{u}) \\ &\leq (\|\mathbf{g}(\mathbf{y}(\mathbf{x})) - \mathbf{g}(\mathbf{x})\| + \alpha^{-1} \|\mathbf{y}(\mathbf{x}) - \mathbf{x}\|) \|\mathbf{u} - \mathbf{y}(\mathbf{x})\|, \end{aligned}$$

hence,

$$\tau \leq (\|\mathbf{g}(\mathbf{y}(\mathbf{x})) - \mathbf{g}(\mathbf{x})\| + \alpha^{-1} \|\mathbf{y}(\mathbf{x}) - \mathbf{x}\|).$$

Setting $\mathbf{x} = \mathbf{z}^l$ here, we obtain a contradiction with $\Delta(\mathbf{z}^l) \leq \delta_l \sqrt{n} \to 0$ if $l \to \infty$. □

This property holds true for all the modifications of the basic cycle and corresponds to the similar properties of the projection methods; see [21].

As the method has a two-level structure with each stage containing a finite number of iterations of the basic cycle, it is more suitable to derive its complexity estimate, which gives the total amount of work of the method. We take for simplicity the method with Basic cycle 2c and suppose that the function f is convex and Lipschitz continuous and its partial gradients satisfy Lipschitz continuity conditions with constants L_i for each $i \in M$. Then it was showed in Section 3 that $\lambda_k \geq \bar{\lambda}_{(i_k)} \geq \bar{\lambda} > 0$ and $\bar{\lambda}_{(i)} = 2(1 - \beta)/L_i$.

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

$$N(\varepsilon) \leq \sum_{l=1}^{l(\varepsilon)} N_{(l)}, \tag{22}$$

where $N_{(l)}$ denotes the total number of iterations at stage l. We proceed to estimate the right-hand side of (22). To change the δ_l, we apply the rule

$$\delta_l = \nu^l \delta_0, l = 0, 1, \ldots; \quad \nu \in (0, 1), \delta_0 > 0. \tag{23}$$

By (22), we have

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

hence

$$N_{(l)} \leq \alpha\Phi(\mathbf{z}^{l-1})/(\beta\bar{\lambda}\delta_l^2). \tag{24}$$

Under the above assumptions from Proposition 2.2 (c) with $\mathbf{u} = \mathbf{x}^* \in X^*$ we now have

$$\begin{aligned}&\mu(\mathbf{y}(\mathbf{x})) - \mu(\mathbf{x}^*) \leq \langle \mathbf{g}(\mathbf{y}(\mathbf{x})), \mathbf{y}(\mathbf{x}) - \mathbf{x}^* \rangle + h(\mathbf{y}(\mathbf{x})) - h(\mathbf{x}^*) \\ &\leq (L + \alpha^{-1})\|\mathbf{y}(\mathbf{x}) - \mathbf{x}\|\|\mathbf{x}^* - \mathbf{y}(\mathbf{x})\|.\end{aligned}$$

Setting $\mathbf{x} = \mathbf{u}^l$, we obtain

$$\Phi(\mathbf{z}^l) \leq \Phi(\mathbf{u}^l) \leq d(L + \alpha^{-1})\Delta(\mathbf{u}^l) \leq d(L + \alpha^{-1})\delta_l\sqrt{n},$$

where d denotes the diameter of the set

$$X' = \left\{\mathbf{x} \in X \,\middle|\, \mu(\mathbf{x}) \leq \mu(\mathbf{z}^0)\right\}.$$

It follows that

$$\nu^{-l(\varepsilon)} \leq d(L + \alpha^{-1})\delta_0\sqrt{n}/\varepsilon = C_1/\varepsilon.$$

Next, using (24) now gives

$$N_{(l)} \leq \alpha d(L + \alpha^{-1})\delta_{l-1}\sqrt{n}/(\beta\bar{\lambda}\delta_l^2) = \alpha d(L + \alpha^{-1})\sqrt{n}/(\beta\bar{\lambda}\nu\delta_l) = C_2\nu^{-l-1}.$$

Combining both the inequalities in (22), we obtain

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

We have obtained the first estimate.

Theorem 5.2 *Let a sequence $\{\mathbf{z}^l\}$ generated by the method with Basic cycle 1c, where rule (23) is used. Suppose that the function f is convex and Lipschitz continuous and its partial gradients satisfy Lipschitz continuity conditions with constants L_i for each $i \in M$. Then the method has the complexity estimate*

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

where $C_1 = d(L + \alpha^{-1})\delta_0\sqrt{n}$ and $C_2 = \alpha d(L + \alpha^{-1})\sqrt{n}/(\beta\bar{\lambda}\delta_0)$.

Suppose additionally that the function f is strongly convex with constant $\varkappa$. Then problem (7) has a unique solution $\mathbf{x}^*$. From Proposition 2.2 (c) with $\mathbf{u} = \mathbf{x}^*$ we have

$$\varkappa\|\mathbf{x}^* - \mathbf{y}(\mathbf{x})\|^2 \le \mu(\mathbf{y}(\mathbf{x})) - \mu(\mathbf{x}^*) \le \langle \mathbf{g}(\mathbf{y}(\mathbf{x})), \mathbf{y}(\mathbf{x}) - \mathbf{x}^*\rangle + h(\mathbf{y}(\mathbf{x})) - h(\mathbf{x}^*)$$
$$\le (L + \alpha^{-1})\|\mathbf{y}(\mathbf{x}) - \mathbf{x}\|\|\mathbf{x}^* - \mathbf{y}(\mathbf{x})\|,$$

hence

$$\varkappa\|\mathbf{x}^* - \mathbf{y}(\mathbf{x})\| \le (L + \alpha^{-1})\|\mathbf{y}(\mathbf{x}) - \mathbf{x}\|$$

and

$$\mu(\mathbf{y}(\mathbf{x})) - \mu(\mathbf{x}^*) \le \varkappa^{-1}(L + \alpha^{-1})^2\|\mathbf{y}(\mathbf{x}) - \mathbf{x}\|^2.$$

Setting $\mathbf{x} = \mathbf{u}^l$, we obtain

$$\Phi(\mathbf{z}^l) \le \Phi(\mathbf{u}^l) \le \varkappa^{-1}(L + \alpha^{-1})^2 n\delta_l^2 = C_3\delta_l^2,$$

It follows from (23) and (24) that

$$N_{(l)} \le \varkappa^{-1}(L + \alpha^{-1})^2 n\delta_{l-1}^2/(\beta\bar{\lambda}\delta_l^2) = \varkappa^{-1}(L + \alpha^{-1})^2 n/(\beta\bar{\lambda}\nu^2) = C_4\nu^{-2},$$

and

$$l(\varepsilon) \le 0.5\ln(C_3/\varepsilon)/\ln(\nu^{-1}).$$

Combining these inequalities in (22), we obtain

$$N(\varepsilon) \le l(\varepsilon)C_4\nu^{-2} \le 0.5C_4\ln(C_3/\varepsilon)/(\nu^2\ln(\nu^{-1})).$$

This estimate corresponds to the linear rate of convergence.

Theorem 5.3 *Let a sequence $\{\mathbf{z}^l\}$ generated by the method with Basic cycle 1c, where rule (23) is used. Suppose that the function f is strongly convex with constant $\varkappa$ and Lipschitz continuous and its partial gradients satisfy Lipschitz continuity conditions with constants L_i for each $i \in M$. Then the method has the complexity estimate*

$$N(\varepsilon) \le l(\varepsilon)C_4\nu^{-2} \le 0.5C_4\ln(C_3/\varepsilon)/(\nu^2\ln(\nu^{-1})),$$

where $C_3 = d\varkappa^{-1}(L + \alpha^{-1})^2 n$ and $C_4 = \varkappa^{-1}(L + \alpha^{-1})^2 n/(\beta\bar{\lambda})$.

We observe that the order of the estimates is similar to that in the usual projection (splitting) methods under the same assumptions; see e.g. [13, 2].

Table 1: The numbers of iterations (it) and derivative calculations (cl)

	(GDS)		(MDS)		(DS1a)	
	it	cl	it	cl	it	cl
$N = 2$	2	4	2	4	2	4
$N = 5$	10	50	10	50	19	60
$N = 10$	17	170	57	570	65	209
$N = 20$	35	700	161	3220	203	679
$N = 40$	105	4100	901	36040	738	2869
$N = 80$	228	18240	3244	259520	3555	11638
$N = 100$	201	20100	4787	478700	4331	16869

6 Computational experiments

In order to check the performance of the above methods we carried out preliminary series of computational experiments. Due to the similarity of iterative processes, we tested only descent splitting method with Basic cycle 1 or 1a ((DS1) or (DS1a), respectively, for short). The main goal was to compare it with the marginal coordinate descent splitting method, which selects the direction with the maximal value of $\Delta_i(\mathbf{x}^k)$, i.e., calculates all the partial derivatives ((MDS) for short). Besides, we took also the usual descent splitting method, which coincides with the steepest descent one in case $h \equiv 0$ and $X = \mathbb{R}^N$ ((GDS) for short). We took the single coordinate partition of $\mathbb{R}^N$, i.e., set $N = n$ and $N_i = 1$ for $i = 1, \ldots, N$. We took $\Delta(\mathbf{x}^k)$ as accuracy measure, chose the accuracy 0.1 and set $\alpha = 1$ for all the methods. The methods were implemented in Delphi with double precision arithmetic. For simplicity, we took only unconstrained test problems when $X = \mathbb{R}^N$.

In the first series, we took the convex quadratic cost function

$$\mu(\mathbf{x}) = 0.5\|A\mathbf{x} - \mathbf{b}\|^2 + 0.5\|\mathbf{x}\|^2,$$

where $\mathbf{b}$ was a fixed vector whose elements were defined by trigonometric functions, for instance, $a_{ij} = \sin(i/j)\cos(ij)$ and $b_i = (1/i)\sin(i)$, as well as the starting point $\mathbf{z}^0$ with $z_j^0 = j|\sin(j)|$. Here we took versions with exact line-search. For (DS1a), we chose the rule $\delta_{l+1} = \nu\delta_l$ with $\nu = 0.5$. The results are given in Table 1.

In the second series, we took the composite convex cost function

$$\mu(\mathbf{x}) = f(\mathbf{x}) + h(\mathbf{x}),$$

where

$$f(\mathbf{x}) = 0.5\|A\mathbf{x} - b\|^2 + 0.5\|\mathbf{x}\|^2$$

Table 2: The numbers of iterations (it) and derivative calculations (cl)

	(GDS)		(MDS)		(DS1)	
	it	cl	it	cl	it	cl
$N = 2$	7	14	7	14	7	14
$N = 5$	10	50	28	56	32	75
$N = 10$	10	100	75	750	74	210
$N = 20$	30	600	225	4500	258	980
$N = 40$	66	2640	468	18720	471	1957
$N = 80$	73	5840	1120	89600	1159	5037
$N = 100$	85	8500	1271	127100	1271	5890

and

$$h(\mathbf{x}) = \sum_{i=1}^{N} |x_i|,$$

where elements of A and $\mathbf{b}$ were again defined by trigonometric functions, for instance, $m_{ij} = 1/(i+1) + 2\sin(i/j)\cos(ij)/j$ and $b_i = n\sin(i)$, as well as the starting point $\mathbf{z}^0$ with $z_j^0 = n|\sin(j)|$. Here we took versions with the Armijo line-search of form (20). For (DS1), we chose the parameters $\beta = \theta = 0.5$ and the rule $\delta_{l+1} = \nu\delta_l$ with $\nu = 0.5$. The results are given in Table 2. In all the cases, (DS1a) and (DS1) showed the explicit preference over (MDS). Here, we see even their preference over (GDS). However, for problems where $A^\top A$ was closer to the unit matrix, (GDS) showed rather rapid convergence in comparison with (DS1a). Besides, we compared (DS1) with the usual cyclic descent splitting method ((CDS) for short) with additional evaluation of all the line-searches. The results are given in Table 3. We see that (CDS) requires less iterations for the same accuracy, but taking arbitrary descent directions increases the number of line-searches. Thus, we conclude that (DS1) performance seems rather satisfactory, but tuning its parameters needs further investigations.

7 Conclusions

We described a new class of coordinate-wise descent splitting methods for decomposable composite optimization problems involving non-smooth functions. These methods are based on selective coordinate variations together with some threshold strategy. We show that they keep the convergence properties of the usual descent splitting ones together with reduction of the total computational expenses. Besides, they are suitable for large scale problems and can be implemented in a distributed (multi-agent) manner. The preliminary results of computational tests showed rather satisfactory convergence.

Table 3: The numbers of iterations (it), derivative calculations (cl), and line-searches (ls)

	(CDS)			(DS1)		
	it	cl	ls	it	cl	ls
$N = 2$	11	11	22	7	14	23
$N = 5$	51	51	172	32	75	96
$N = 10$	130	130	345	74	210	227
$N = 20$	543	543	1463	258	980	939
$N = 40$	883	883	2303	471	1957	1618
$N = 80$	2165	2165	5805	1159	5037	3734
$N = 100$	2605	2605	6203	1271	5890	4090

References

[1] M. Fukushima and H. Mine, *A generalized proximal point algorithm for certain non-convex minimization problems*, Intern. J. Syst. Sci., 12 (1981), pp.989–1000.

[2] M. Patriksson, *Nonlinear Programming and Variational Inequality Problems: A Unified Approach*, Kluwer Academic Publishers, Dordrecht, 1999.

[3] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Progr., 117 (2009), pp. 387-423.

[4] Richtárik, P., Takáč, M.: *Parallel coordinate descent methods for big data optimization*, (November 25, 2013), Available at http://arxiv.org/pdf/1212.0873.pdf

[5] F. Facchinei, G. Scutari, and S. Sagratella, *Parallel selective algorithms for nonconvex big data optimization*, (August 8, 2014), Available at http://arxiv.org/pdf/1402.5521.pdf

[6] Z. Yin, P. Ming, and Y. Wotao, *Parallel and distributed sparse optimization*, Asilomar Conference on Signals, Systems and Computers, IEEE, pp.646–659, 2013.

[7] I.V. Konnov, *Selective bi-coordinate variations for resource allocation type problems*, (November 5, 2014), Available at SSRN: http://ssrn.com/abstract=2519662

[8] M.L. Balinski and P. Wolfe (ed.), *Nondifferentiable Optimization. Math. Progr. Study 3*, North - Holland, Amsterdam, 1975.

[9] I.V. Konnov, *Combined relaxation method for decomposable variational inequalities*, Optimiz. Meth. Software, 10 (1999), pp.711–728.

[10] I.V. Konnov, *A class of combined relaxation methods for decomposable variational inequalities*, Optimization, 51 (2002), pp.109–125.

[11] E. Allevi, A. Gnudi, and I.V. Konnov, *Combined relaxation method with Frank-Wolfe type auxiliary procedures for variational inequalities over product sets*, Pure Math. Appl., 12 (2001), pp.1–9.

[12] F.H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983.

[13] E.S. Levitin and B.T. Polyak, *Constrained minimization methods*, USSR Comp. Maths. Math. Phys., 6 (1966), pp.1–50.

[14] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, 1989.

[15] V.G. Karmanov, *Convergence estimates for iterative minimization methods*, USSR Comp. Maths. Math. Phys., 1 (1974), pp.1–13.

[16] I.V. Konnov, *Nonlinear Optimization and Variational Inequalities*, Kazan Univ. Press, Kazan, 2013. [In Russian]

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

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

[19] I. Lobel, A. Ozdaglar, and D. Feijer, *Distributed multi-agent optimization with state-dependent communication*, Math. Program. 129 (2011), 255–284.

[20] I.V. Konnov, *Combined Relaxation Methods for Variational Inequalities*, Springer, Berlin, 1985.

[21] B.T. Polyak, *Introduction to Optimization*, Nauka, Moscow, 1983; Engl. transl. in Optimization Software, New York, 1987.