Block coordinate descent for smooth nonconvex constrained minimization

E. G. Birgin[1] · **J. M. Martínez[2]**

Abstract

At each iteration of a block coordinate descent method one minimizes an approximation of the objective function with respect to a generally small set of variables subject to constraints in which these variables are involved. The unconstrained case and the case in which the constraints are simple were analyzed in the recent literature. In this paper we address the problem in which block constraints are not simple and, moreover, the case in which they are not defined by global sets of equations and inequations. A general algorithm that minimizes quadratic models with quadratic regularization over blocks of variables is defined and convergence and complexity are proved. In particular, given tolerances $\delta > 0$ and $\varepsilon > 0$ for feasibility/complementarity and optimality, respectively, it is shown that a measure of $(\delta, 0)$-criticality tends to zero; and the number of iterations and functional evaluations required to achieve (δ, ε)-criticality is $O(\varepsilon^{-2})$. Numerical experiments in which the proposed method is used to solve a continuous version of the traveling salesman problem are presented.

Keywords Coordinate descent methods · Convergence · Complexity

AMS Subject Classifications: 90C30 · 65K05 · 49M37 · 90C60 · 68Q25

✉ E. G. Birgin
  egbirgin@ime.usp.br

  J. M. Martínez
  martinez@ime.unicamp.br

1   Department of Computer Science, Institute of Mathematics and Statistics, University of São
    Paulo, Rua do Matão, 1010, Cidade Universitária, São Paulo, SP 05508-090, Brazil

2   Department of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific
    Computing (IMECC), State University of Campinas, Campinas, SP 13083-859, Brazil

1 Introduction

The structure of many practical problems suggests the employment of Block Coordinate Descent (BCD) methods for Optimization. At each iteration of a BCD method only a block of variables is modified with the purpose of obtaining sufficient decrease of the objective function.

Wright [11] surveyed traditional approaches and modern advances on the definition and analysis of Coordinate Descent methods. His analysis addresses mostly unconstrained problems in which the objective function is convex. The case in which a smooth function is minimized over a closed and convex set with a separable structure was considered in [7]. Although the Coordinate Descent paradigm is very natural and is implicitly used in different mathematical contexts, a classical example by Powell [10] showed that convergence results cannot be achieved under excessively naive implementations.

In a recent report [2] it was shown that, requiring sufficient descent based on regularization, the drawbacks represented by Powell's example can be removed. In that paper it was also shown that methods based on high-order regularization can be defined in which convergence and worst-case complexity can be proved. However, the main results shown in [2] indicate that it is not worthwhile to use Taylor-like models of order greater than 2 because complexity is dominated by the necessity of keeping consecutive iterations close enough, a requirement that is hard to achieve if models and regularizations are of high order. This is the reason why, in the present paper, we restrict ourselves to quadratic models of the objective function and quadratic regularization.

The novelty of our approach relies on the extension of the techniques of [2] to the case in which there are arbitrary constraints at every block. As a consequence, at each iteration of BCD we minimize a problem with (probably) a small number of variables that must satisfy arbitrary constraints. Moreover, the block feasible set may not be defined by a global set of equalities and inequalities, as usually in constrained optimization. Instead, equalities and inequalities that define the feasible set are local in nature in a sense that will be defined below, making it possible more general domains than the ones defined by global equalities and inequalities.

This paper is organized as follows. In Sect. 2 the definition of the optimization problem is given. In Sect. 3 we define the BCD method for solving the main problem. In Sect. 4 we prove convergence and complexity results. In Sect. 5 we explain how to solve subproblems. Experiments are shown in Sects. 6 and 7 we state conclusions and lines for future research.

Notation. $\| \cdot \|$ denotes the Euclidean norm. ∇_i denotes the gradient with respect to the ith block of coordinates. $\mathbb{N} = \{0, 1, 2, \dots\}$.

2 The problem

Assume that $f : \mathbb{R}^n \to \mathbb{R}$, $n_i \geq 1$ for all $i = 1, \ldots, n_{\text{blocks}}$, and $\sum_{i=1}^{n_{\text{blocks}}} n_i = n$. Let us write $x^T = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_{n_{\text{blocks}}}^T)^T$, where $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$ for all $i = 1, \ldots, n_{\text{blocks}}$, so

$$f(x) = f(x_1, \ldots, x_n) = f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_{\text{blocks}}}).$$

The problem considered in the present work is given by

$$\text{Minimize} f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_{\text{blocks}}}) \text{ subject to } \boldsymbol{x}_i \in \Omega_i, i = 1, \ldots, n_{\text{blocks}}. \tag{1}$$

The assumption below includes conditions on the sets Ω_i and on the way they are described.

Assumption A1 For all $i = 1, \ldots, n_{\text{blocks}}$ the set $\Omega_i \subset \mathbb{R}^{n_i}$ is closed and bounded. Moreover, for all $i = 1, \ldots, n_{\text{blocks}}$, there exist open sets $A_{i,j}$, $j = 1, \ldots, n_{\text{ops}}(i)$, such that

$$\Omega_i \subset \cup_{j=1}^{n_{\text{ops}}(i)} A_{i,j}$$

and there exist $n_{\text{g}}(i, j)$ smooth functions

$$g_{i,j,\ell} : A_{i,j} \to \mathbb{R}, \ \ell = 1, \ldots, n_{\text{g}}(i, j) \tag{2}$$

such that

$$\Omega_i \cap A_{i,j} = \{\boldsymbol{x}_i \in A_{i,j} \mid g_{i,j,\ell}(\boldsymbol{x}_i) \leq 0, \ \ell = 1, \ldots, n_{\text{g}}(i, j)\}. \tag{3}$$

The constraints $g_{i,j,\ell}(\boldsymbol{x}_i) \leq 0$ are said to be *constitutive* of Ω_i in the open covering set $A_{i,j}$. The explicit inclusion of equality constraints in (2,3) offers no difficulty and we state the case with only inequalities for the sole purpose of simplifying the notation. See Fig. 1 for an example of a set Ω_i, the open covering $A_{i,j}$ and functions $g_{i,j,\ell}$.

In Fig. 1, sets $A_{1,j}$ for $j = 1, 2, 3, 5$ (four out of the five open circles) are such that $n_{\text{g}}(1, j) = 2$; while sets $A_{1,j}$ for $j = 6, 7, 8, 9, 10$ (the open rectangles) are such that $n_{\text{g}}(1, j) = 1$. In all cases, constitutive constraints are linear. Sets $A_{1,11}$ and $A_{1,12}$ are *not* displayed in the picture for clarity. They are two congruent open triangles that cover the interior of Ω_1 that appears uncovered in the picture; and they are such that $n_{\text{g}}(1, 11) = n_{\text{g}}(1, 12) = 0$. The "internal kink" at $A_{1,4}$ makes the constitutive constraint of $A_{1,4}$ to deserve special consideration. In $A_{1,4}$ the feasible region is the complement of a set defined by constraints of the form $a_1 x_1 + b_1 x_2 + c_1 \geq 0$ and $a_2 x_1 + b_2 x_2 + c_2 \geq 0$. Let us write $z_1 = a_1 x_1 + b_1 x_2 + c_1$ and $z_2 = a_2 x_1 + b_2 x_2 + c_2$. Then, in the plane (z_1, z_2) the feasible region is, locally, the complement of the non-negative orthant. Define $\varphi(z_1, z_2) = (z_1 z_2)^2$ if $z_1 \geq 0$ and $z_2 \geq 0$, whereas $\varphi(z_1, z_2) = -(z_1 z_2)^2$ otherwise. Then, the feasible region is, locally, given by $\varphi(z_1, z_2) \leq 0$; i.e. $n_{\text{g}}(1, 4) = 1$ and the constitutive constraint is given by φ.

In Sect. 6, we illustrate our method using a problem in which the sets Ω_i are simple nonconvex polygons. A small set Ω_i of this type is depicted in Fig. 1. Let us show here that we can cover Ω_i with open sets A_{ij}, and define constitutive constraints,
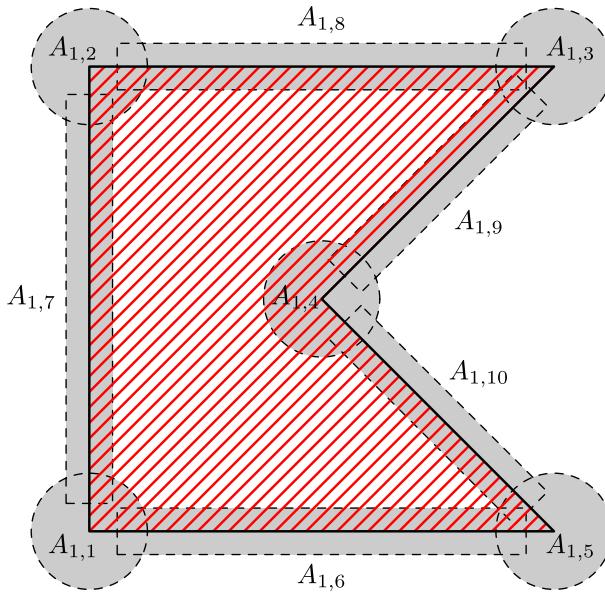
Fig. 1 Illustration of a set Ω_1 (dashed red) covered by $n_{\text{ops}}(1) = 12$ open sets $A_{1,1}, \ldots, A_{1,12}$ (Color figure online)

in such a way that, if $\bar{x}$ is a (local or global) minimizer of a smooth function $\psi(x_1, x_2)$ over Ω_i, then there exists A_{ij} containing $\bar{x}$ such that the KKT conditions with respect to the constitutive constraints in A_{ij} are satisfied; see Assumption A7 in Sect. 5. In fact, the form described here for defining the open sets A_{ij} and the constitutive constraints clearly satisfies Assumption A1 and also satisfies Assumption A6 in Sect. 5.

The idea is to cover the polygon with the following open sets: (a) The interior of Ω_i; (b) Open sets whose intersection with Ω_i is an open side of the boundary (a side of the boundary that does not contains the extreme vertices); (c) Convex open sets whose intersection with the boundary of Ω_i is a vertex plus a fraction of the two segments that connect the vertex with its two neighbors. In Case (a) there are no constitutive constraints. In Case (b) there is only one constitutive constraint, which defines a half-plane. In Case (c), the corresponding open set is divided in a region contained in Ω_i and a region that is not contained in Ω_i. If the region contained in Ω_i is convex the constitutive constraints are linear and define the intersection of two half-planes. Otherwise the expression of the constitutive constraints is not relevant and could be, for example, as the constraint φ mentioned in the example in Fig. 1.

In Cases (a) and (b) it is straightforward that a local minimizer of ψ must satisfy KKT conditions. Analogously, KKT conditions also hold in Case (c) if the region contained in Ω_i is convex. In fact, in these three situations constitutive constraints are naturally linear, a condition under which minimizers are necessarily KKT points [9]. Therefore, we only need to analyze Case (c) when the region contained in Ω is not convex. In this case, by the geometry of the feasible region in a neighborhood of $\bar{x}$, there exist linearly independent directions v_1 and v_2 such

that $\pm v_1$ and $\pm v_2$ are feasible directions. Since $\bar{x}$ is a minimizer, we must have $\nabla \psi(\bar{x})^T v_1 \geq 0$, $\nabla \psi(\bar{x})^T (-v_1) \geq 0$, $\nabla \psi(\bar{x})^T v_2 \geq 0$, and $\nabla \psi(\bar{x})^T (-v_2) \geq 0$, implying that $\nabla \psi(\bar{x}) = 0$. So $\bar{x}$ is a KKT point with null multipliers.

3 Block coordinate descent method

In this section we present the main algorithm designed for solving (1). At each iteration k of the Block Coordinate Descent (BCD) method, given the current iterate $x^k = ((x_1^k)^T, \ldots, (x_{n_{\text{blocks}}}^k)^T)^T$, we choose $i_k \in \{1, \ldots, n_{\text{blocks}}\}$ and compute $x_{i_k}^{k+1}$ by approximately solving

$$\text{Minimize}_{x_{i_k} \in \mathbb{R}^{n_i}} f(x_1, \ldots, x_{n_{\text{blocks}}}) \text{ subject to } x_{i_k} \in \Omega_{i_k} \text{ and } x_j = x_j^k \text{ for all } j \neq i_k, \tag{4}$$

i.e. we approximately minimize the function f fixing the blocks x_j such that $j \neq i_k$. For $j \neq i_k$, we define $x_j^{k+1} = x_j^k$. The sense in which problem (4) is solved only approximately is clarified below.

The algorithmic parameters of BCD are the sufficient descent parameter α, which defines progress of the objective function and implicitly penalizes the distance between consecutive iterates, the tolerance δ with respect to complementarity conditions, the parameter θ that defines sufficient descent of the model at each iteration, and the minimal positive regularization parameter σ_{min}. The model Hessian matrices B_k may be used to mimic available second derivative information but do not play any significant role from the point of view of complexity or convergence and the choice $B_k = 0$ is always possible. At Step 2 of the algorithm we choose the block of variables with respect to which we wish to improve the objective function. In general, we minimize approximately a quadratic model increasing the regularizing parameter as far as the sufficient condition (10) is not satisfied. Alternatively, we employ the true objective function as a model, because such alternative is possible in many practical problems.

Algorithm 3.1 Assume that $\alpha > 0$, $\delta \geq 0$, $\theta > 0$, $\sigma_{\text{min}} > 0$, and $x_i^0 \in \Omega_i$ for $i = 1, \ldots, n_{\text{blocks}}$ are given. Initialize the iteration number $k \leftarrow 0$.

Step 1. Set $\sigma \leftarrow 0$, choose $i_k \in \{1, \ldots, n_{\text{blocks}}\}$, and define $B_k \in \mathbb{R}^{n_{i_k}} \times \mathbb{R}^{n_{i_k}}$ symmetric.

Step 2. Find $j_k \in \{1, \ldots, n_{\text{ops}}(i_k)\}$ and $x_{i_k}^{\text{trial}} \in A_{i_k j_k}$ such that if $\sigma > 0$ then Alternative 1 holds while if $\sigma = 0$ then either Alternative 1 or Alternative 2 holds.
Alternative 1:

$$\nabla_{i_k} f(x^k)^T (x_{i_k}^{\text{trial}} - x_{i_k}^k) + \frac{1}{2}(x_{i_k}^{\text{trial}} - x_{i_k}^k)^T B_k (x_{i_k}^{\text{trial}} - x_{i_k}^k) + \frac{\sigma}{2} \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2 \leq 0 \tag{5}$$

and there exist $\mu_{i_k j_k, \ell} \geq 0$ for $\ell = 1, \ldots, n_g(i_k, j_k)$ for which

$$\left\| \nabla_{i_k} f(x^k) + B_k(x_{i_k}^{\text{trial}} - x_{i_k}^k) + \sigma(x_{i_k}^{\text{trial}} - x_{i_k}^k) \right.$$
$$\left. + \sum_{\ell=1}^{n_{\text{g}}(i_k,j_k)} \mu_{i_k j_k,\ell} \nabla g_{i_k j_k,\ell}(x_{i_k}^{\text{trial}}) \right\| \leq \theta \| x_{i_k}^{\text{trial}} - x_{i_k}^k \| \tag{6}$$

and

$$\min\{\mu_{i_k j_k,\ell}, -g_{i_k j_k,\ell}(x_{i_k}^{\text{trial}})\} \leq \delta, \ \ell = 1, \ldots, n_{\text{g}}(i_k,j_k). \tag{7}$$

Alternative 2: There exist $\mu_{i_k j_k,\ell} \geq 0$ for $\ell = 1, \ldots, n_{\text{g}}(i_k,j_k)$ for which

$$\left\| \nabla_{i_k} f(x_1^k, \ldots, x_{i_k}^{\text{trial}}, \ldots, x_{n_{\text{blocks}}}^k) + \sum_{\ell=1}^{n_{\text{g}}(i_k,j_k)} \mu_{i_k j_k,\ell} \nabla g_{i_k j_k,\ell}(x_{i_k}^{\text{trial}}) \right\| = 0 \tag{8}$$

and

$$\min\{\mu_{i_k j_k,\ell}, -g_{i_k j_k,\ell}(x_{i_k}^{\text{trial}})\} \leq 0, \ \ell = 1, \ldots, n_{\text{g}}(i_k,j_k). \tag{9}$$

Step 3.   Test the sufficient descent condition

$$f(x_1^k, \ldots, x_{i_k}^{\text{trial}}, \ldots, x_{n_{\text{blocks}}}^k) \leq f(x^k) - \alpha \| x_{i_k}^{\text{trial}} - x_{i_k}^k \|^2. \tag{10}$$

If (10) holds, then define $\sigma_k = \sigma$, $x_{i_k}^{k+1} = x_{i_k}^{\text{trial}}$ and $x_i^{k+1} = x_i^k$ for all $i \neq i_k$, set $k \leftarrow k + 1$ and go to Step 1. Otherwise, set $\sigma \leftarrow \max\{\sigma_{\min}, 2\sigma\}$ and go to Step 2.

Remark We will see that, from the theoretical point of view, Alternative 2 is not necessary. Convergence and complexity theoretical results follow without any difficulty with Alternative 1 only. Alternative 2 was included because, in many cases, a procedure exists to find a global minimizer with respect to a single block. So, in Alternative 2 we allow the algorithm to choose such minimizer, with the only condition that it must satisfy KKT conditions in the block. However, note that the test (10) is still necessary and cannot be eliminated. The reason is that its fulfillment implies that the difference between consecutive iterations tends to zero and this feature is essential for the convergence of coordinate search methods. See the counterexample in [10] and the discussion with only box constraints in [2].

4 Convergence and complexity

In this section we prove convergence and complexity results. We say that $(x_1, \ldots, x_{n_{\text{blocks}}})$ is (δ, ϵ)-critical if there exist open sets $A_{i,j}$ ($i = 1, \ldots, n_{\text{blocks}}$, $j = 1, \ldots, n_{\text{ops}}(i)$) satisfying Assumption A1 such that, for all $i = 1, \ldots, n_{\text{blocks}}$, x_i satisfies the KKT conditions for the minimization of $f(x_1, \ldots, x_{n_{\text{blocks}}})$ restricted to the constitutive constraints of $A_{i,j}$, $j = 1 \ldots, n_{\text{ops}}(i)$, with tolerance $\epsilon > 0$ and satisfies complementarity and feasibility with respect to the same constraints with

tolerance $\delta > 0$. Under proper assumptions we prove that, given $\delta \geq 0$ (which is a parameter of the algorithm BCD), the natural measure of $(\delta, 0)$-criticality tends to zero and the number of iterations and evaluations that are necessary to obtain (δ, ε)-criticality is $O(\varepsilon^{-2})$.

In Assumption A2 we state that the gradients of the objective function must satisfy Lipschitz conditions.

Assumption A2 There exists $\gamma > 0$ such that, for all x_{i_k} and $x_{i_k}^{\text{trial}}$ computed at Algorithm 3.1,

$$\|\nabla_{i_k} f(x^k) - \nabla_{i_k} f(x_1^k, \ldots, x_{i_k}^{\text{trial}}, \ldots, x_{n_{\text{blocks}}}^k)\| \leq \gamma \|x_{i_k} - x_{i_k}^{\text{trial}}\| \tag{11}$$

and

$$f(x_1^k, \ldots, x_{i_k}^{\text{trial}}, \ldots, x_{n_{\text{blocks}}}^k) \leq f(x^k) + \nabla_{i_k} f(x^k)^T (x_{i_k}^{\text{trial}} - x_{i_k}^k) + \frac{\gamma}{2} \|x_{i_k} - x_{i_k}^{\text{trial}}\|^2. \tag{12}$$

Moreover, if x^{k_1} differs from x^{k_2} in only one block i_{diff},

$$\|\nabla_i f(x^{k_1}) - \nabla_i f(x^{k_2})\| \leq \gamma \|x_{i_{\text{diff}}}^{k_2} - x_{i_{\text{diff}}}^{k_1}\|. \tag{13}$$

for all $i = 1, \ldots, n_{\text{blocks}}$.

Assumption A3 merely states that model Hessians should be uniformly bounded.

Assumption A3 There exist $c_B > 0$ such that for all $k \in \mathbb{N}$,

$$\|B_k\| \leq c_B. \tag{14}$$

Assumptions A2 and A3 are sufficient to prove that every iteration of BCD is well defined, as sufficient descent (10) is obtained increasing the regularization parameter σ a finite number of times.

Lemma 4.1 *Assume that Assumptions A2 and A3 hold and that, for all $k \in \mathbb{N}$, the computation of j_k and $x_{i_k}^{\text{trial}}$ according to Step 2 of BCD is possible. Then, the test (10) is satisfied after at most*

$$\log_2 \left(\frac{\gamma + c_B + 2\alpha}{\sigma_{\min}} \right) + 1$$

increases of σ at Step 3. Moreover,

$$\sigma_k < \sigma_{\max} := 2(\gamma + c_B + 2\alpha). \tag{15}$$

Proof If the test (10) is satisfied when Step 2 is executed with $\sigma = 0$, then the thesis holds trivially. So, we need to consider only the case in which $\sigma > 0$ and, in consequence, j_k and $x_{i_k}^{\text{trial}}$ satisfy Alternative 1. By (12) in Assumption A2,

$$f(x_1^k, \ldots, x_{i_k}^{\text{trial}}, \ldots, x_{n_{\text{blocks}}}^k) \leq f(x^k) + \nabla_{i_k} f(x^k)^T (x_{i_k}^{\text{trial}} - x_{i_k}^k) + \frac{\gamma}{2} \|x_{i_k} - x_{i_k}^{\text{trial}}\|^2$$

$$+ \left(\frac{1}{2} (x_{i_k}^{\text{trial}} - x_{i_k}^k)^T B_k (x_{i_k}^{\text{trial}} - x_{i_k}^k) + \frac{1}{2} \sigma \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2 \right)$$

$$- \left(\frac{1}{2} (x_{i_k}^{\text{trial}} - x_{i_k}^k)^T B_k (x_{i_k}^{\text{trial}} - x_{i_k}^k) + \frac{1}{2} \sigma \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2 \right)$$

Then, by (5),

$$f(x_1^k, \ldots, x_{i_k}^{\text{trial}}, \ldots, x_{n_{\text{blocks}}}^k) \leq f(x^k) + \frac{\gamma}{2} \|x_{i_k} - x_{i_k}^{\text{trial}}\|^2 - \frac{1}{2} (x_{i_k}^{\text{trial}}$$

$$- x_{i_k}^k)^T B_k (x_{i_k}^{\text{trial}} - x_{i_k}^k) - \frac{1}{2} \sigma \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2.$$

Therefore, by (14) in Assumption A3,

$$f(x_1^k, \ldots, x_{i_k}^{\text{trial}}, \ldots, x_{n_{\text{blocks}}}^k) \leq f(x^k) + \frac{\gamma}{2} \|x_{i_k} - x_{i_k}^{\text{trial}}\|^2 + \frac{1}{2} c_B \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2$$

$$- \frac{1}{2} \sigma \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2$$

$$= f(x^k) + \frac{1}{2} (\gamma + c_B - \sigma) \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2.$$

Then, the inequality (10) holds if

$$\frac{1}{2} (\gamma + c_B - \sigma) \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2 \leq -\alpha \|x_{i_k}^{\text{trial}} - x_{i_k}^k\|^2,$$

i.e. if $\sigma \geq \gamma + c_B + 2\alpha$. Since, by definition, σ initially receives the value zero and then receives values of the form $2^{\ell-1} \sigma_{\min}$, where ℓ is the number of executions of $\sigma \leftarrow \max\{\sigma_{\min}, 2\sigma\}$, then the number of increases of σ that are necessary to obtain (10) is bounded above by

$$\log_2 \left(\frac{\gamma + c_B + 2\alpha}{\sigma_{\min}} \right) + 1$$

as we wanted to prove. Finally, (15) comes from the fact that the largest unsuccessful value of σ must be strictly less than $\gamma + c_B + 2\alpha$ and the next (successful) value is twice that amount by definition. □

Assumption A4 The sequence $\{f(x^k)\}$ is bounded below.

The following lemma is a simple consequence of (10) and Assumption A4.

Lemma 4.2 Assume that Assumptions A2, A3, and A4 hold and that for all $k \in \mathbb{N}$, the computation of j_k and $x_{i_k}^{\text{trial}}$ at Step 2 is possible. Then, $\lim_{k \to \infty} \|x_{i_k}^{k+1} - x_{i_k}^k\| = \lim_{k \to \infty} \|x^{k+1} - x^k\| = 0$ and, given $\varepsilon > 0$, the number of iterations at which $\|x_{i_k}^{k+1} - x_{i_k}^k\| > \varepsilon$ is bounded above by

$$\frac{f(\boldsymbol{x}^0) - f_{\text{bound}}}{\alpha\varepsilon^2} \tag{16}$$

where f_{bound} is an arbitrary lower bound of $\{f(\boldsymbol{x}^k)\}$.

Proof By Assumption A4 there exists $f_{\text{bound}} \in \mathbb{R}$ such that $f(\boldsymbol{x}^k) \geq f_{\text{bound}}$ for all $k \in \mathbb{N}$. Then, the fact that $\|\boldsymbol{x}_{i_k}^{k+1} - \boldsymbol{x}_{i_k}^k\|$ tends to zero comes from (10); and this implies that $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|$ tends to zero as well because, by definition, $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\| = \|\boldsymbol{x}_{i_k}^{k+1} - \boldsymbol{x}_{i_k}^k\|$. Finally, if $\|\boldsymbol{x}_{i_k}^{k+1} - \boldsymbol{x}_{i_k}^k\| > \varepsilon$, then, by (10), $f(x^{k+1}) \leq f(x^k) - \alpha\varepsilon^2$; and this reduction can no occur more than (16) times, as we wanted to prove. □

Assumption A5 states that every block of components i is chosen for minimization at infinitely many iterations and, moreover, at every m consecutive iterations we necessarily find at least one at which i is chosen.

Assumption A5 There exists $m \in \{1, 2, \dots\}$ such that, for all $v \in \{1, \dots, n_{\text{blocks}}\}$, $i_k = v$ infinitely many times and, if $k_1 < k_2 < k_3, \dots$ is the set of all the iteration indices k such that $i_k = v$, one has that $k_1 \leq m$, and $k_{j+1} - k_j \leq m$ for all $j = 1, 2, 3, \dots$.

The following theorems are the main convergence result of this paper. The idea is the following. According to Algorithm 3.1, at iteration k, we select a block $i_k = i_{\text{chosen}}$ and optimize with respect to the variables of this block up to the approximate fulfillment of restricted KKT conditions. These restricted KKT conditions hold in one of the open sets that cover $\Omega_{i_{\text{chosen}}}$ and involve the constraints that are constitutive in this open set. The variables $\boldsymbol{x}_{i_{\text{chosen}}}$ do not change during some (less than m) iterations; therefore, during these iterations, thanks to the Lipschitz assumption (13), the approximate KKT conditions with respect to the variables i_{chosen} still hold with respect to the same open set and the same constitutive constraints used at iteration k. After these (less than m) iterations the block i_{chosen} is selected again, and the process is repeated. Since all the blocks are chosen infinitely many times in the way described by Assumption A5, approximate KKT conditions eventually hold with respect to all the blocks and we are able to establish the number of iterations that we need for the fulfillment of KKT conditions up to an arbitrary precision ε.

Theorem 4.1 Assume that Assumptions A2, A3, A4, and A5 hold and that for all $k \in \mathbb{N}$, the computation of j_k and $\boldsymbol{x}_{i_k}^{\text{trial}}$ at Step 2 is possible. For $i \in \{1, \dots, n_{\text{blocks}}\}$ and $k \geq m$, define $o(i, k) := j_{\hat{k}}$, where $\hat{k}$ is the latest iteration (not larger than k) at which $i_{\hat{k}} = i$. Then, for $i \in \{1, \dots, n_{\text{blocks}}\}$ and $k \geq m$, we have that $\mu_{i,o(i,k),\ell} \geq 0$ for $\ell = 1, \dots, n_{\text{g}}(i, o(i, k))$,

$$\boldsymbol{x}_i^k \in A_{i,o(i,k)},$$
$$\min\{\mu_{i,o(i,k),\ell}, -g_{i,o(i,k),\ell}(\boldsymbol{x}_i^k)\} \leq \delta,$$

and

$$\lim_{k \to \infty} \left\| \nabla_i f(x^k) + \sum_{\ell=1}^{n_g(i,o(i,k))} \mu_{i,o(i,k),\ell} \nabla g_{i,o(i,k),\ell}(\boldsymbol{x}_i^k) \right\| = 0.$$

Moreover, given $\varepsilon > 0$, the number of iterations at which

$$\left\| \nabla_i f(x^k) + \sum_{\ell=1}^{n_g(i,o(i,k))} \mu_{i,o(i,k),\ell} \nabla g_{i,o(i,k),\ell}(\boldsymbol{x}_i^k) \right\| > \varepsilon$$

is bounded above by

$$\frac{f(\boldsymbol{x}^0) - f_{\text{bound}}}{\alpha(\varepsilon/(c_4 m))^2},$$

where f_{bound} is an arbitrary lower bound of $\{f(\boldsymbol{x}^k)\}$ and

$$c_4 := c_B + \sigma_{\max} + \theta + \gamma. \tag{17}$$

Proof Let $i \in \{1, \dots, n_{\text{blocks}}\}$ be arbitrary. By Assumption A5, there exists $k_1 \leq m$ such that $i_{k_1} = i$. Without loss of generality, in order to simplify the notation, suppose that $k_1 = 0$. Consider first the case where, in Step 2, Alternative 1 holds. Then, at iteration $k_1 = 0$ one defines B_0 and finds j_0, $\boldsymbol{x}_i^1 \in A_{i,j_0}$, and $\mu_{i,j_0,\ell} \geq 0$ for $\ell = 1, \dots, n_g(i, j_0)$ such that

$$\left\| \nabla_i f(x^0) + B_0(\boldsymbol{x}_i^1 - \boldsymbol{x}_i^0) + \sigma_0(\boldsymbol{x}_i^1 - \boldsymbol{x}_i^0) + \sum_{\ell=1}^{n_g(i,j_0)} \mu_{i,j_0,\ell} \nabla g_{i,j_0,\ell}(\boldsymbol{x}_i^1) \right\| \leq \theta \| \boldsymbol{x}_i^1 - \boldsymbol{x}_i^0 \| \tag{18}$$

and

$$\min\{\mu_{i,j_0,\ell}, -g_{i,j_0,\ell}(\boldsymbol{x}_i^1)\} \leq \delta \quad \text{for } \ell = 1, \dots, n_g(i, j_0). \tag{19}$$

By (18) and (15),

$$\left\| \nabla_i f(x^0) + \sum_{\ell=1}^{n_g(i,j_0)} \mu_{i,j_0,\ell} \nabla g_{i,j_0,\ell}(\boldsymbol{x}_i^1) \right\| \leq (c_B + \sigma_{\max} + \theta) \| \boldsymbol{x}_i^1 - \boldsymbol{x}_i^0 \|. \tag{20}$$

Then, by (11),

$$\left\| \nabla_i f(\boldsymbol{x}_1^0, \dots, \boldsymbol{x}_i^1, \dots, \boldsymbol{x}_{n_{\text{blocks}}}^0) + \sum_{\ell=1}^{n_g(i,j_0)} \mu_{i,j_0,\ell} \nabla g_{i,j_0,\ell}(\boldsymbol{x}_i^1) \right\| \leq (c_B + \sigma_{\max} + \theta + \gamma) \| \boldsymbol{x}_i^1 - \boldsymbol{x}_i^0 \|. \tag{21}$$

Since $\boldsymbol{x}_s^1 = \boldsymbol{x}_s^0$ for every $s = 1, \dots, n_{\text{blocks}}$, $s \neq i$, by (21) and the definition of c_4 in (17), we have that

$$\left\|\nabla_i f(x^1) + \sum_{\ell=1}^{n_g(i,j_0)} \mu_{i,j_0,\ell} \nabla g_{i,j_0,\ell}(x_i^1)\right\| \le c_4 \|x_i^1 - x_i^0\|. \tag{22}$$

Recall that (19) and (22) were obtained under Alternative 1. On the other hand, under Alternative 2, (19) and (22) follow trivially from (9) and (8), respectively.

Since x^2 may differ from x^1 only in the block i_1, by (13), we have that

$$\|\nabla_i f(x^2) - \nabla_i f(x^1)\| \le \gamma \|x_{i_1}^2 - x_{i_1}^1\|.$$

Then, by (22),

$$\left\|\nabla_i f(x^2) + \sum_{\ell=1}^{n_g(i,j_0)} \mu_{i,j_0,\ell} \nabla g_{i,j_0,\ell}(x_i^1)\right\| \le c_4 \|x_i^1 - x_i^0\| + \gamma \|x_{i_1}^2 - x_{i_1}^1\|. \tag{23}$$

Since x^3 may differ from x^2 only in the block i_2, by (13) and (23), we have that

$$\left\|\nabla_i f(x^3) + \sum_{\ell=1}^{n_g(i,j_0)} \mu_{i,j_0,\ell} \nabla g_{i,j_0,\ell}(x_i^1)\right\| \le c_4 \|x_i^1 - x_i^0\| + \gamma \|x_{i_1}^2 - x_{i_1}^1\| + \gamma \|x_{i_2}^3 - x_{i_2}^2\|. \tag{24}$$

So, using an inductive argument, for all $k \in \mathbb{N}$,

$$\left\|\nabla_i f(x^k) + \sum_{\ell=1}^{n_g(i,j_0)} \mu_{i,j_0,\ell} \nabla g_{i,j_0,\ell}(x_i^1)\right\| \le c_4 \|x_i^1 - x_i^0\| + \gamma \sum_{v=2}^{k} \|x^v - x^{v-1}\|. \tag{25}$$

Thus, by the definition of c_4 in (17), for all $k \in \mathbb{N}$,

$$\left\|\nabla_i f(x^k) + \sum_{\ell=1}^{n_g(i,j_0)} \mu_{i,j_0,\ell} \nabla g_{i,j_0,\ell}(x_i^1)\right\| \le c_4 \sum_{v=1}^{k} \|x^v - x^{v-1}\|. \tag{26}$$

Let us now get rid of the simplifying assumption $i_0 = i$ and assume that the set of indices k at which $i_k = i$ is $k_1 < k_2 < \dots$. Renaming the indices in (19) and (26), we get that

$$\min\{\mu_{i,j_{k_r},\ell}, -g_{i,j_{k_r},\ell}(x_i^{k_r+1})\} \le \delta \quad \text{for} \quad \ell = 1, \dots, n_g(i,j_{k_r}) \tag{27}$$

and for all $r = 1, 2, \dots$ and all $k = k_r + 1, k_r + 2, \dots$, in particular for all $k = k_r + 1, k_r + 2, \dots, k_{r+1}$,

$$\left\|\nabla_i f(x^k) + \sum_{\ell=1}^{n_g(i,j_{k_r})} \mu_{i,j_{k_r},\ell} \nabla g_{i,j_{k_r},\ell}(x_i^{k_r+1})\right\| \le c_4 \sum_{v=k_r+1}^{k} \|x^v - x^{v-1}\|. \tag{28}$$

But, by the definition of the sequence $k_1, k_2, \dots$, $x_i^{k_r}$ may change from iteration k_r to iteration $k_r + 1$ but it does not change from $k_r + 1$ to $k_r + 2$, $k_r + 2$ to $k_r + 3$, until

$k_{r+1} - 1$ to k_{r+1}; and it may change again from iteration k_{r+1} to iteration $k_{r+1} + 1$. This means that, for all $r = 1, 2, \ldots$, we have that $x_i^{k_r+1} = x_i^{k_r+2} = \cdots = x_i^{k_{r+1}}$. Therefore, (27) and (28) imply that for all $r = 1, 2, \ldots$ and $k = k_r + 1, k_r + 2, \ldots, k_{r+1}$,

$$\min\{\mu_{i,j_{k_r},\ell}, -g_{i,j_{k_r},\ell}(x_i^k)\} \leq \delta \quad \text{for} \quad \ell = 1, \ldots, n_g(i, j_{k_r}) \tag{29}$$

and

$$\left\| \nabla_i f(x^k) + \sum_{\ell=1}^{n_g(i,j_{k_r})} \mu_{i,j_{k_r},\ell} \nabla g_{i,j_{k_r},\ell}(x_i^k) \right\| \leq c_4 \sum_{v=k_r+1}^{k} \|x^v - x^{v-1}\|. \tag{30}$$

Moreover, by definition, $o(i, k_r + 1) = o(i, k_r + 2) = \cdots = o(i, k_{r+1}) = j_{k_r}$ for $r = 1, 2, \ldots$. So, from (29) and (30), we get that, for $k \geq m$, $x_i^k \in A_{i,o(i,k)}$ and $\mu_{i,j_0,\ell} \geq 0$ for $\ell = 1, \ldots, n_g(i, j_0)$ are such that

$$\min\{\mu_{i,o(i,k),\ell}, -g_{i,o(i,k),\ell}(x_i^k)\} \leq \delta \quad \text{for} \quad \ell = 1, \ldots, n_g(i, o(i, k)) \tag{31}$$

and

$$\left\| \nabla_i f(x^k) + \sum_{\ell=1}^{n_g(i,o(i,k))} \mu_{i,o(i,k),\ell} \nabla g_{i,o(i,k),\ell}(x_i^k) \right\| \leq c_4 \sum_{v=k_r+1}^{k} \|x^v - x^{v-1}\|. \tag{32}$$

By Lemma 4.2, given $\varepsilon > 0$, the number of iterations at which $\|x^v - x^{v-1}\| > \varepsilon/(c_4 m)$ is bounded above by $(f(x^0) - f_{\text{bound}})/(\alpha(\varepsilon/(c_4 m))^2)$. Then, since the sum in the second member of (30) involves at most m terms, the number of iterations at which the left-hand side of (30) is bigger than ε is bounded above by $(f(x^0) - f_{\text{bound}})/(\alpha(\varepsilon/(c_4 m))^2)$. Moreover, since $\varepsilon > 0$ is arbitrary, taking limits on both sides of (32), we have that

$$\lim_{k \to \infty} \left\| \nabla_i f(x^k) + \sum_{\ell=1}^{n_g(i,o(i,k))} \mu_{i,o(i,k),\ell} \nabla g_{i,o(i,k),\ell}(x_i^k) \right\| = 0. \tag{33}$$

$\square$

The final theorem of this section proves worst-case functional complexity of order $O(\varepsilon^{-2})$.

Theorem 4.2 Assume that Assumptions A2, A3, A4, and A5 hold and that for all $k \in \mathbb{N}$, the computation of j_k and $x_{i_k}^{\text{trial}}$ at Step 2 is possible. Let the indices $o(i, k)$ be as defined in Theorem 4.1. Then, given $\varepsilon > 0$, Algorithm 3.1 performs at most

$$n_{\text{blocks}} \left(\frac{f(x^0) - f_{\text{bound}}}{\alpha(\varepsilon/(c_4 m))^2} \right)$$

iterations and at most

$$\log_2\left(\frac{\gamma + c_B + 2\alpha}{\sigma_{\min}}\right) + 2$$

functional evaluations per iteration, where f_{bound} is an arbitrary lower bound of $\{f(x^k)\}$ and c_4 is given by (17), to compute an iterate x^{k+1} such that

$$x_i^{k+1} \in A_{i,o(i,k)},$$
$$\mu_{i,o(i,k),\ell} \geq 0 \ \text{ and } \ \min\{\mu_{i,o(i,k),\ell}, -g_{i,o(i,k),\ell}(x_i^{k+1})\} \leq \delta, \ \ell = 1,\dots,n_g(i,o(i,k)),$$

and

$$\left\| \nabla_i f(x^{k+1}) + \sum_{\ell=1}^{n_g(i,o(i,k))} \mu_{i,o(i,k),\ell} \nabla g_{i,o(i,k),\ell}(x_i^{k+1}) \right\| \leq \varepsilon$$

for all $i = 1,\dots,n_{\text{blocks}}$.

Proof The proof follows from Theorem 4.1, Lemma 4.1, and the definition of Algorithm 3.1, because the number of functional evaluations per iterations is equal to the number of increments of σ plus one. (This ignores the fact that, disregarding the first iteration, the value of $f(x^k)$ can in fact be obtained from the previous iteration, in which case the number of functional evaluations and the number of increases of σ per iteration coincide.) □

5 Solving subproblems

In this section we present an algorithm for computing $x_{i_k}^{\text{trial}}$ at Step 2 of iteration k of Algorithm 3.1. The well-definiteness of the proposed approach requires, in addition to Assumption A1, two additional assumptions. The first one concerns the relation between each set Ω_i, its covering sets $A_{i,j}$, and its constitutive constraints $g_{i,j,\ell}$. This assumption states that, if a point is in the closure of $A_{i,j}$ and satisfies the constitutive constraints associated with $A_{i,j}$ then this point necessarily belongs to Ω_i.

Assumption A6 For all $i \in \{1,\dots,n_{\text{blocks}}\}$ and all $j \in \{1,\dots,n_{\text{ops}}(i)\}$, if $x_i \in \overline{A_{i,j}}$ and $g_{i,j,\ell}(x_i) \leq 0$ for $\ell = 1,\dots,n_g(i,j)$, then $x_i \in \Omega_i$.

To illustrate the necessity of Assumption A6, Fig. 2 exhibits a pathological example at which the assumption does not hold. In the figure, the set Ω_1 (in red) is the union of the segments $S_h = \{(t,2) \mid t \in [2,4]\}$ and $S_v = \{(2,t) \mid t \in [-1,2]\}$. Defining $n_{\text{ops}}(1) = 2$ and the open sets $A_{1,1} = \{(x_1,x_2) \mid (x_1-2)^2 + x_2^2 < 1\}$ and $A_{1,2} = \{(x_1,x_2) \mid x_2 > 1/2\}$, we have that $\Omega_1 \subset \cup_{j=1}^{n_{\text{ops}}(1)} A_{1,j}$. The open ball $A_{1,1}$ is displayed in the figure, while the open half-space $A_{1,2}$ is not. Inside the open ball $A_{1,1}$, the set Ω_1 can be described by the deliberately unorthodox choice $(x_1 - 1)(x_1 - 2) = 0$, while inside the open half-space $A_{1,2}$, the set Ω_1 can be described by the equality $(x_1 - 2)(x_2 - 2) = 0$ plus the inequalities $2 \leq x_1 \leq 4$ and
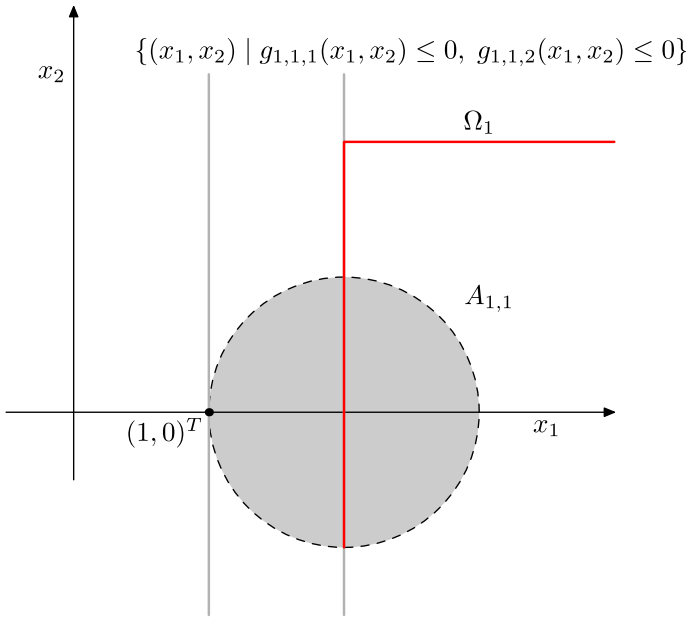
Fig. 2 A pathological example at which Assumption A6 does not hold

$1/2 \leq x_2 \leq 2$. Therefore, (3) is satisfied for $i = 1$ and $j = 1, \ldots, n_{\text{ops}}(1)$ by defining $n_g(1, 1) = 2$, $\qquad\qquad n_g(1, 2) = 6$, $\qquad\qquad g_{1,1,1}(x_1, x_2) = (x_1 - 1)(x_1 - 2)$, $g_{1,1,2}(x_1, x_2) = -g_{1,1,1}(x_1, x_2)$, $\qquad\qquad g_{1,2,1}(x_1, x_2) = (x_1 - 2)(x_2 - 2)$, $g_{1,2,2}(x_1, x_2) = -g_{1,2,1}(x_1, x_2)$, $\qquad g_{1,2,3}(x_1, x_2) = 2 - x_1$, $\qquad g_{1,2,4}(x_1, x_2) = x_1 - 4$, $g_{1,2,5}(x_1, x_2) = 1/2 - x_2$, and $g_{1,2,6}(x_1, x_2) = x_2 - 2$. The set of points that satisfy the constitutive constraints of Ω_1 in $A_{1,1}$ is shown in light-gray in the figure. The point $(x_1, x_2)^T = (1, 0)^T$ belongs to $\overline{A_{1,1}}$ and satisfies the corresponding constitutive constraints $g_{1,1,1}(x_1, x_2) \leq 0$ and $g_{1,1,2}(x_1, x_2) \leq 0$. However, it does not belong to Ω_1. Therefore, Assumption A6 does not hold. The equality that describes Ω_1 within $A_{1,1}$ was deliberately chosen to construct an example that does not satisfy Assumption A6 and to show that Assumption A6 is necessary. To satisfy Assumption A6 in this example, it would suffice to consider as constitutive constraints of Ω_1 within $A_{1,1}$ the two inequalities equivalent to $x_1 = 2$ instead of the two inequalities equivalent to $(x_1 - 1)(x_1 - 2) = 0$.

The second additional assumption states that every global minimizer of a smooth function onto Ω_i belongs to some $A_{i,j}$ satisfying the KKT conditions with respect to the constitutive constraints related with $A_{i,j}$.

Assumption A7 For all $i \in \{1, \ldots, n_{\text{blocks}}\}$, if x_i is a global minimizer of a smooth function onto Ω_i, there exists $j \in \{1, \ldots, n_{\text{ops}}(i)\}$ such that $x_i \in A_{i,j}$ and x_i satisfies the KKT conditions with respect to the constitutive constraints associated with $A_{i,j}$.

To illustrate the necessity of Assumption A7, let us show that it does not hold in the example of Fig. 1. Assume that the global minimizer G of a smooth function ψ over the set Ω_1 belongs to the boundary of Ω_1 and to the ball $A_{1,4}$ but does not belong to $A_{1,9} \cup A_{1,10} \cup A_{1,11} \cup A_{1,12}$ and is not the center $C_{1,4}$ of $A_{1,4}$[1]. For example, take an adequate infeasible point Q very close to the desired global minimizer G and define $\psi(P) = \|P - Q\|^2$. The global minimizer G belongs only to the open set $A_{1,4}$, the gradient $\nabla\psi(G)$ is nonnull but, according to the definition of the constitutive constraint φ of $A_{1,4}$, $\nabla\varphi(G) = 0$. Therefore, the KKT condition does not hold in this case. Fortunately, Fig. 1 shows just an arbitrary illustrative example and, as already described in Sect. 2, there are simple ways to define the open sets A_{ij} and the constitutive constraints in such a way that Assumptions A1, A6, and A7 hold.

In order to pursue Alternative 1, at Step 2 of Algorithm 3.1, $j_k \in \{1, \dots, n_{\mathrm{ops}}(i_k)\}$ and $x_{i_k}^{\mathrm{trial}} \in A_{i_k j_k}$ must be found such that (5) holds and such that there exist $\mu_{i_k j_k, \ell} \geq 0$ for $\ell = 1, \dots, n_{\mathrm{g}}(i_k, j_k)$ for which (6) and (7) hold. To accomplish this, for j from 1 to $n_{\mathrm{ops}}(i_k)$, provided it is affordable, we could compute a global minimizer z_j^* of

$$\text{Minimize}_{x \in \mathbb{R}^{n_{i_k}}} \nabla_{i_k} f(x^k)^T (x - x_{i_k}^k) + \frac{1}{2}(x - x_{i_k}^k) B_k (x - x_{i_k}^k) + \frac{\sigma}{2}\|x - x_{i_k}^k\|^2 \tag{34}$$
$$[2mm] \text{ subject to } x \in \overline{A_{i_k j}} \text{ and } g_{i_k j, \ell}(x) \leq 0 \text{ for } \ell = 1, \dots, n_{\mathrm{g}}(i_k, j).$$

If the objective function value at z_j^* is non-positive and $z_j^* \in A_{i_k j}$, then, defining $j_k = j$, we have that $x_{i_k}^{\mathrm{trial}} = z_j^*$ satisfies (5). If we additionally assume that global minimizers of (34) for every $j \in \{1, \dots, n_{\mathrm{ops}}(i_k)\}$ satisfy KKT conditions, then we have that there exist $\mu_{i_k j_k, \ell} \geq 0$ for $\ell = 1, \dots, n_{\mathrm{g}}(i_k, j_k)$ for which (6) and (7) hold.

If none of the global minimizers z_j^* is such the objective function value at z_j^* is non-positive and $z_j^* \in A_{i_k j}$, then we can define $j_k = j^a$ and $x_{i_k}^{\mathrm{trial}} = z_{j^b}^*$, where $z_{j^b}^*$ is the global minimizer (among the $n_{\mathrm{ops}}(i_k)$ computed global minimizers $z_1^*, \dots, z_{n_{\mathrm{ops}}(i_k)}^*$) that achieves the lowest functional value of the objective function in (34) and j^a is such that $z_{j^b}^* \in A_{i_k j^a}$. The functional value of the objective function of (34) at $z_{j^b}^*$ is non-positive because the objective function vanishes at $x_{i_k}^k$ that is a feasible point of (34) for at least one $j \in \{1, \dots, n_{\mathrm{ops}}(i_k)\}$; and $z_{j^b}^* \in A_{i_k j^a}$ for some j^a because, by Assumption A6, $z_{j^b}^* \in \Omega_{i_k}$ and, by definition, $\Omega_{i_k} \subset A_{i_k} = \cup_{j=1}^{n_{\mathrm{ops}}(i_k)} A_{i_k j}$. Moreover, $z_{j^b}^*$ must also be a global minimizer of (34) with $j = j^a$. Thus, by Assumption A7, it fulfills KKT conditions and, therefore, there exist $\mu_{i_k j_k, \ell} \geq 0$ for $\ell = 1, \dots, n_{\mathrm{g}}(i_k, j_k)$ for which (6) and (7) hold.

It is worth noting that the objective function in (34) is a linear function if $B_k = 0$ and $\sigma = 0$; and it is a convex quadratic function if $B_k + \sigma I$ is positive definite. Moreover, it is always possible to choose the open covering sets $A_{i,j}$ for all i and j in such a way their closures are simple sets like balls, boxes, or polyhedrons. Furthermore, it

---

[1] We already shown, in Sect. 2, that, although non-regular, if a smooth function ψ has a minimizer at $C_{1,4}$, its gradient $\nabla\psi(C_{1,4})$ is necessarily null; so that $C_{1,4}$ is a KKT point of the minimization of ψ subject to $\varphi \leq 0$.

is also possible that more efficient problem-dependent alternatives exist for the computation of $x^{\text{trial}}_{i_k}$; and it is also possible trying to find $x^{\text{trial}}_{i_k}$ satisfying Alternative 2 when $\sigma = 0$.

The correctness of the strategy proposed in this section depends on the satisfaction of Assumptions A1 to A7. Assumptions A1, A6, and A7 refer to the relationship between the feasible region of problem (1), the open sets, and the constitutive constraints. We have already shown that, for simple nonconvex polygons, these things can be defined in such a way that the assumptions are satisfied. Assumption A3 asks the models' Hessians B_k to be uniformly bounded. This can be satisfied by definition, for example by choosing $B_k = 0$ in all iterations, or any other option that satisfies the assumption. Likewise, Assumption A5 refers to the way in which the blocks i_k are chosen at each iteration. This assumption can also be satisfied in many ways, for example with a cyclic choice of i_k. The remaining assumptions (Assumptions A2 and A4) depend on the objective function of problem (1) and their validity will be explained in Sect. 6.1, when the problem under consideration is introduced.

6 Experiments

In this section we describe numerical experiments using the BCD method. Sect. 6.1 describes what we have called the continuous version of the traveling salesman problem (TSP). In this problem, the BCD method is used to evaluate the merit of the function that should be minimized. Since this function is computed many times along the whole process, this problem provides many experimental applications of BCD. In Sect. 6.2, we describe a simple heuristic and a way to generate a starting point for solving the continuous TSP. Although simple, these considered methods are part of the state of the art of methods used to solve the classical TSP. Moreover, they serve to illustrate the application of the BCD method, which could be used in the same way in combination with any other strategy. Sect. 6.3 describes a problem-dependent way to find a $x^{\text{trial}}_{i_k}$ in the BCD method that satisfies the requirements of Alternative 2. Sect. 6.4 describes the computational experiment itself.

6.1 Continuous traveling salesman problem

The travelling salesman problem (TSP) is one of the most studied combinatorial optimization problems for which a vast literature exists; see, for example, [3], and the references therein. In its classical version, p cities with known pairwise "distances" $d_{ij} > 0$ are given and the problem consists in finding a permutation $i_1, i_2, \ldots, i_p$ that minimizes $d_{i_p, i_1} + \sum_{\nu=1}^{p-1} d_{i_\nu, i_{\nu+1}}$. In the present work, we consider a continuous variant of the classical TSP in which "cities" are not fixed and, therefore, their pairwise distances vary. More precisely, given a set of polygons $\Omega_1, \Omega_2, \ldots, \Omega_p$, that may be nonconvex, the problem consists of finding points $x_i \in \Omega_i$ for $i = 1, \ldots, p$ and a permutation $i_1, i_2, \ldots, i_p$ that minimize $\|x_{i_p} - x_{i_1}\| + \sum_{\nu=1}^{p-1} \|x_{i_\nu} - x_{i_{\nu+1}}\|$. Polygons may be seen as representing countries, regions, districts, or neighbourhoods of

a city; and the interpretation is that "visiting a polygon" is equivalent to "visiting any point within the polygon".

The application of the BCD method in this context is very natural. Any method to solve the classical TSP requires to evaluate the merit of a permutation $i_1, i_2, \ldots, i_p$ by calculating $d_{i_p, i_1} + \sum_{v=1}^{p-1} d_{i_v, i_{v+1}}$. In the variant we are considering, given a permutation $i_1, i_2, \ldots, i_p$, the BCD method is used to find the $x_{i_v} \in \Omega_{i_v}$ for $v = 1, \ldots, p$ that minimize $\|x_{i_p} - x_{i_1}\| + \sum_{v=1}^{p-1} \|x_{i_v} - x_{i_{v+1}}\|$. In other words, given a permutation $i_1, i_2, \ldots, i_p$, the BCD method is used to find a solution x^* to the problem

$$
\begin{aligned}
\text{Minimize}_{x \in \mathbb{R}^n} f(i_1, \ldots, i_p; x) &:= \|x_{i_p} - x_{i_1}\| \\
&+ \sum_{v=1}^{p-1} \|x_{i_v} - x_{i_{v+1}}\| \text{ subject to } x_{i_v} \in \Omega_{i_v} \text{ for } v = 1, \ldots, p,
\end{aligned}
\tag{35}
$$

other words, given a permutation where $n_{\text{blocks}} = p$, $n_i = 2$ for $i = 1, \ldots, n_{\text{blocks}}$, $n = 2p$, and $x = (x_1^T, \ldots, x_p^T)^T$; while the problem as a whole consists in finding a permutation $i_1^*, \ldots, i_p^*$ such that $f(i_1^*, \ldots, i_p^*; x^*)$ is as small as possible. That is, the BCD integrates the process of evaluating the merit of a given permutation. With this tool, constructive heuristics and neighborhood-based local searches already developed for the classical TSP can be adapted to the problem under consideration.

Assumption A4 requires that the sequence $\{f(x^k)\}$ is bounded below. This is clearly valid for the objective function f defined in (35) since, being a sum of distances, it is always nonnegative. Let us now show that Assumption A2 also holds. The closed polygons $\Omega_1, \ldots, \Omega_p$ are such that $\Omega_{i_1} \cap \Omega_{i_2} = \emptyset$ for all $i_1 \neq i_2$. Then, there exists $\eta > 0$ such that, for every trial point computed by the algorithm, the distance between two blocks of coordinates is greater than η. Consider the auxiliary function $d : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ given by $d(x, y) = \sqrt{\|x - y\|^2 + (\|x - y\|^2 - \eta^2)^4}$ if $\|x - y\| \leq \eta$ and $d(x, y) = \|x - y\|$ otherwise; and consider the function $\tilde{f}$ defined as the function f in (35) but replacing each appearance of $\|x - y\|$ by $d(x, y)$. Since $d(x, y)$ coincides with $\|x - y\|$ for every pair of blocks of coordinates computed by the algorithm, this implies that the sequence of iterates computed using f as defined in (35) coincides with the sequence of iterates computed if we use $\tilde{f}$ instead of f. Therefore, we may consider that the objective function of problem (35) is $\tilde{f}$. Clearly, the new function has continuous second derivatives, therefore its gradient satisfy a Lipschitz condition as required in Assumption A2. (It should be noted that $\tilde{f}$, being the sum of nonnegative terms, also satisfies Assumption A4.)

6.2 Discrete optimization strategy

In the present work, among the huge range of possibilities and in order to illustrate the usage of the BCD method, we consider a local search with an insertion-based neighborhood. The initial solution is given by a constructive heuristic also based on insertions, as we now describe; see [1] and the references therein. The construction of the initial guess starts defining $(i_1, i_2) = (1, 2)$ and $x_{i_1} \in \Omega_{i_1}$ and $x_{i_2} \in \Omega_{i_2}$ as the ones that minimize $\|x_{i_1} - x_{i_2}\|$, computed with the BCD method. Then, to construct (i_1, i_2, i_3), the method considers inserting index 3 before i_1, between i_1 and i_2, and after i_2. For

each of the three possibilities, optimal $x_{i_1} \in \Omega_{i_1}$, $x_{i_2} \in \Omega_{i_2}$, and $x_{i_3} \in \Omega_{i_3}$ are computed with the BCD method. Among the three permutations, the one with smallest $\|x_{i_1} - x_{i_2}\| + \|x_{i_2} - x_{i_3}\| + \|x_{i_3} - x_{i_1}\|$ is chosen. The method proceeds in this way until a permutation with p elements, that constitutes the initial guess, is completed. A typical iteration of the local search proceeds as follows. Given the current permutation $(i_1, i_2, \ldots, i_p)$ and its associated points $x_{i_v} \in \Omega_{i_v}$ for $v = 1, \ldots, p$, each i_s for $s = 1, \ldots, p$ is removed and reinserted at all possible places $t \neq s$. For each possible insertion, corresponding $x_{i_1}, x_{i_2}, \ldots, x_p$ are computed with the BCD method. This type of movement is also known as *relocation* and, as mentioned in [1, p.342], it has been used with great success in the TSP [8]. Once an insertion is found that improves the current solution, the iteration is completed, i.e. the first neighbour that improves the current solution defines the new iterate, in contrast to a "best movement" strategy in which all neighbors are considered and the best of them defines the new iterate. The local search ends when no neighbour is found that improves the current iterate.

6.3 Finding optimal points with BCD method for a given permutation

In this section we describe how to solve problem (35) with the BCD method. At iteration k of the BCD method, an index $i_k \in \{1, \ldots, n_{\text{blocks}}\}$ is chosen at Step 1. Then at Step 2, there are two alternatives. If $\sigma = 0$, then $x_{i_k}^{\text{trial}}$ satisfying Alternative 1: (5,6,7) or Alternative 2: (8,9) must be computed; while, if $\sigma > 0$, then only Alternative 1 is a possibility. Sect. 5 describes a way of computing $x_{i_k}^{\text{trial}}$ satisfying Alternative 1 for any value of σ. However, for the particular problem under consideration, minimizing $f(x)$ as a function of $x_{i_k} \in \Omega_{i_k}$ reduces to

$$\text{Minimize}_{x_{i_k} \in \mathbb{R}^2} \|a - x_{i_k}\| + \|x_{i_k} - b\| \text{ subject to } x_{i_k} \in \Omega_{i_k}, \tag{36}$$

where a and $b \in \mathbb{R}^2$ stand for the "previous" and the "next" point in the permutation; that, in general, correspond to $x_{i_{k-1}}$ and $x_{i_{k+1}}$, respectively. Thus, when $\sigma = 0$, it is easy, computationally tractable, and affordable to compute the global minimizer of (36), which clearly satisfies the requirements of Alternative 2. The global minimizer is either on the segment $[a, b]$ intersected with Ω_{i_k} (that intersection is given by a finite set of segments) or on the boundary of Ω_{i_k}, which is also given by a finite set of segments (its edges). Each segment can be parameterized with a single variable $\lambda \in [0, 1]$. Then, the global minimizer of (36) is given by the best global minimizer among the global minimizers of these simple box-constrained one-dimensional problems. The global minimizer of each box-constrained one-dimensional problem can be computed with brute force up any desired precision. Moreover, if multiple solutions exist, in order in increase the chance of satisfying (10), the closest one to $x_{i_k}^k$ should be preferred.

6.4 Traveling in São Paulo City

For the numerical experiments, we implemented the discrete optimization strategy described in Sect. 6.2 and the BCD method (Algorithm 3.1) described in Sect. 3

with the strategy described in Sect. 6.3 for the computation of $x^{\text{trial}}_{i_k}$. In Algorithm 3.1, we chose $i_k = \text{mod}(k + 1, n_{\text{blocks}})$. (Preliminary experiments with a random choice of i_k, with a discrete uniform distribution in $\{1, 2, \ldots, n_{\text{blocks}}\}$, showed similar results.) Based on the theoretical results, we stop the method at iteration k, if $x^k = x^{k-1} = \cdots = x^{k-n_{\text{blocks}}+1}$. In the numerical experiments, following [2, 5, 6], we consider $\alpha = 10^{-8}$. In all our experiments the required conditions at Step 2 were satisfied using Alternative 2.

All methods were implemented in Fortran 90. Tests were conducted on a computer with a 3.4 GHz Intel Core i5 processor and 8GB 1600 MHz DDR3 RAM memory, running macOS Mojave (version 10.14.6). Code was compiled by the GFortran compiler of GCC (version 8.2.0) with the -O3 optimization directive enabled.

The city of São Paulo, with more than 15 million square kilometers of extension and more than 12 million inhabitants, is the most populous city in Brazil, the American continent, the Portuguese-speaking countries and the entire southern hemisphere. It is administratively divided into thirty-two regions, each of which, in turn, is divided into districts, the latter sometimes subdivided into subdistricts (popularly called neighborhoods); see https://pt.wikipedia.org/wiki/S%C3%A3o_Paulo. The city has a total of 96 neighborhoods. The considered problem consists in finding a shortest route to visit all of them.

The construction of the problem started by downloading a political-administrative map of the city from the city hall website; see http://geosampa.prefeitura.sp.gov.br/. The map describes each neighborhood as a polygon. The polygon with more vertices has 5,691 vertices and, all together, the polygons have 156,852 vertices. To turn the problem into something more tractable, we redefine the polygons with number of vertices $n_v > 100$ (all of them in fact) by considering only the vertices with indices of the form $\text{form } 1 + \lfloor 50/n_v \rfloor j$ for $j = 0, 1, 2, \ldots$. This way, all polygons were left with a number of vertices between 51 and 57, totaling 4,966 vertices. Moreover, for artistic reasons related to the graphical representation of the problem, we shrunk each polygon by 20%. The shrinkage consisted in replacing each vertex v_i by $o_i + 0.8(v_i - o_i)$, where the offset $o_i = \frac{1}{2}(x_{\min} + x_{\max}, y_{\min} + y_{\max})^T$, and $(x_{\min}, y_{\min})^T$ and $(x_{\max}, y_{\max})^T$ correspond to the lower-left and upper-right corners of the smallest rectangle that encloses the polygon. With this procedure we ended up with the $p = 96$ polygons Ω_i for $i = 1, \ldots, p$ that determine the problem (35) under consideration; see Fig. 3.

Table 1 shows the details of the optimization process. The table shows, for each iteration, the length of the current route. It also shows, for each iteration, how many neighbors had to be evaluated to find one that improves the current route. Naturally, each evaluation of a neighbor corresponds to a call to the BCD method. Therefore, the next two columns show the number of calls to the BCD method per iteration and the number of cycles these calls used. The last two columns of the table show these two values accumulated over the iterations. It can be noted from the table that the BCD method is used to solve more than 200,000 subproblems and that this requires, altogether, the execution of more than 3 million cycles, i.e. an average of 15 cycles per problem. The instance under consideration has $p = 96$ points. The constructive
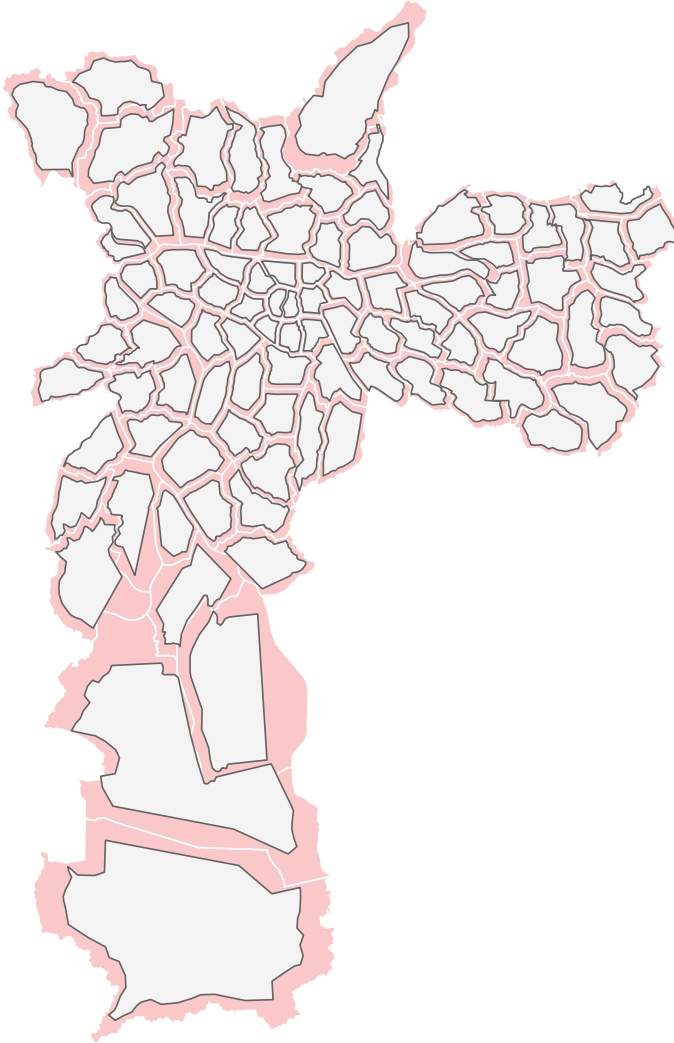
Fig. 3 Representation of the $p = 96$ polygons that determine problem (35). The considered polygons appear in gray, while the original polygons appear in coral on the background merely to improve the artistic appearance of the drawing
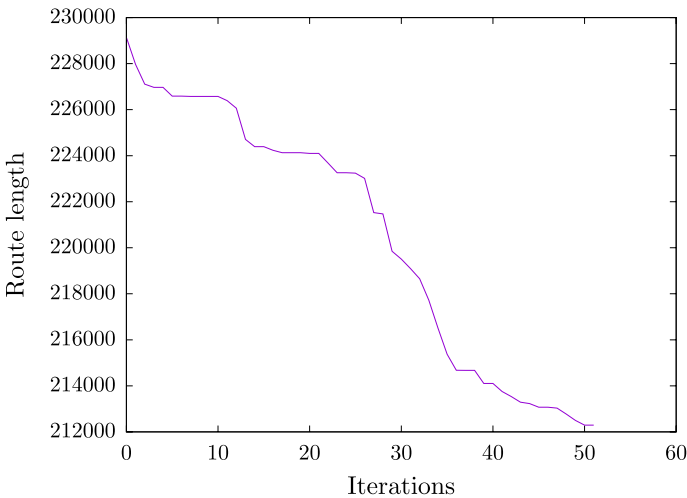
heuristic used to generate the initial point evaluates (by calling the BCD method) $O(\frac{1}{2}p^2)$ permutations; while the reinsertion neighborhood evaluates, in the worst case, $O(p^2)$ neighbors. The first line of the table shows the cost of the constructive heuristic, and is consistent with what we have just mentioned. The remaining lines show that in the first 4 iterations and in a few intermediate iterations the method quickly finds a neighbor that improves the current solution. On the other hand, the average number of neighbors evaluated per iteration is 4,164, which corresponds to approximately 45% of the neighbors. The running time of the algorithm is directly

Table 1 Performance of the heuristic method applied to solve the considered instance of the continuous version of the TSP problem

iter	Route length	Usage of BCD method per iter		Accumulated usage of BCD method	
		# calls	# cycles	# calls	# cycles
0	229,139.65	4,653	40,018	4,653	40,018
1	227,965.59	2	26	4,655	40,044
2	227,110.10	1	10	4,656	40,054
3	226,970.07	3	56	4,659	40,110
4	226,970.07	2	24	4,661	40,134
5	226,588.08	4,125	60,955	8,786	101,089
6	226,586.52	4,418	66,454	13,204	167,543
7	226,575.05	4,220	64,793	17,424	232,336
8	226,575.05	95	1,655	17,519	233,991
9	226,573.77	4,510	66,294	22,029	300,285
10	226,573.77	189	2,658	22,218	302,943
11	226,391.25	4,708	69,757	26,926	372,700
12	226,063.02	4,789	70,987	31,715	443,687
13	224,708.13	3,653	57,797	35,368	501,484
14	224,391.45	4,889	73,541	40,257	575,025
15	224,391.45	3,747	59,159	44,004	634,184
16	224,236.40	4,801	70,928	48,805	705,112
17	224,128.38	4,983	74,453	53,788	779,565
18	224,128.38	95	1,761	53,883	781,326
19	224,128.38	95	908	53,978	782,234
20	224,100.64	3,851	58,075	57,829	840,309
21	224,100.64	3,838	57,462	61,667	897,771
22	223,681.92	4,887	75,693	66,554	973,464
23	223,261.18	3,947	66,182	70,501	1,039,646
24	223,261.18	95	847	70,596	1,040,493
25	223,242.32	5,076	77,901	75,672	1,118,394
26	223,013.14	5,572	92,531	81,244	1,210,925
27	221,526.66	5,701	91,366	86,945	1,302,291
28	221,469.25	94	900	87,039	1,303,191
29	219,846.99	5,606	89,792	92,645	1,392,983
30	219,505.59	1	4	92,646	1,392,987
31	219,096.08	5,761	92,924	98,407	1,485,911
32	218,652.50	6,164	96,849	104,571	1,582,760
33	217,719.39	6,146	97,136	110,717	1,679,896
34	216,509.74	6,256	98,403	116,973	1,778,299
35	215,372.59	6,350	99,083	123,323	1,877,382
36	214,676.14	5,289	87,276	128,612	1,964,658
37	214,674.57	5,077	82,038	133,689	2,046,696
38	214,674.57	1,345	29,796	135,034	2,076,492
39	214,102.74	3,937	71,897	138,971	2,148,389
40	214,102.74	95	1,033	139,066	2,149,422

Table 1 (continued)	iter	Route length	Usage of BCD method per iter		Accumulated usage of BCD method	
			# calls	# cycles	# calls	# cycles
	41	213,754.48	5,190	92,893	144,256	2,242,315
	42	213,533.35	6,049	98,937	150,305	2,341,252
	43	213,290.76	6,241	102,045	156,546	2,443,297
	44	213,231.26	7,195	115,060	163,741	2,558,357
	45	213,071.95	6,338	99,378	170,079	2,657,735
	46	213,070.90	6,625	103,442	176,704	2,761,177
	47	213,032.17	7,297	111,568	184,001	2,872,745
	48	212,773.09	7,681	116,393	191,682	2,989,138
	49	212,499.29	7,585	115,190	199,267	3,104,328
	50	212,292.01	8,161	125,796	207,428	3,230,124
	51	212,292.01	9,120	154,558	216,548	3,384,682

Fig. 4 Route length as a function of the iteration number

proportional and totally dependent on the cost of computing $x_{i_k}^{trial}$. The constructive heuristic used to compute the initial point x^0 used 29.58 seconds of CPU time; while the method as a whole consumed practically one hour of CPU time, exactly 3,589.31 seconds.

Figure 4 shows the evolution of the route length over the iterations of the method; while Fig. 5 shows some of the generated routes. Figure 6 shows the final iterate in detail.

Up to this point, we have described and illustrated in detail the way in which the discrete heuristic method described in Sect. 6.2 made intensive use of the BCD
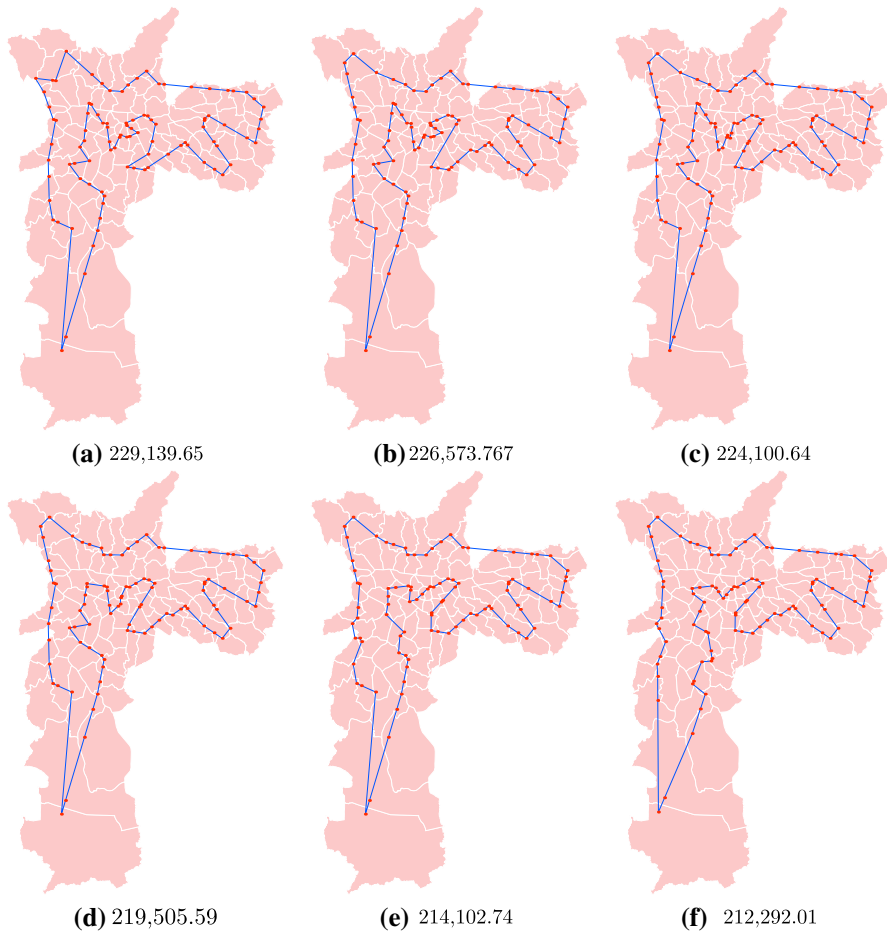
(a) 229,139.65   **(b)** 226,573.767   **(c)** 224,100.64

(d) 219,505.59   **(e)** 214,102.74   **(f)** 212,292.01

Fig. 5 Sample of the routes that are built throughout the iterative optimization process. The length of each route appears near to the route. The map of São Paulo city appears in the background for artistic purposes, but the polygons representing the districts are being omitted for the sake of clarity. (a) Represents the initial guess given by the constructive heuristic; (f) Represents the final iterate (obtained at iteration 51); and (b)–(e) Represent the iterands of the iterations 10, 20, 30 and 40, respectively. It is worth noting that the red dots, each always within its respective polygon that is not being displayed, move from one iteration to another

method to solve the problem (35). We close the numerical experiments section by showing in Fig. 7, with a graphic and a table, the iterands of the BCD method for a specific fixed permutation. To make this figure, we considered the permutation given by the constructive heuristic used to generate x^0 and we randomly draw points inside each of the polygons. The graph and table show the iterations for 13 complete cycles. The method actually uses 44 cycles, but the functional value varies from 229,139.66 at the end of cycle 13 to 229,139.65 at the end of cycle 44, when it stops because all the variables' blocks are repeated from cycle 43 to cycle 44.
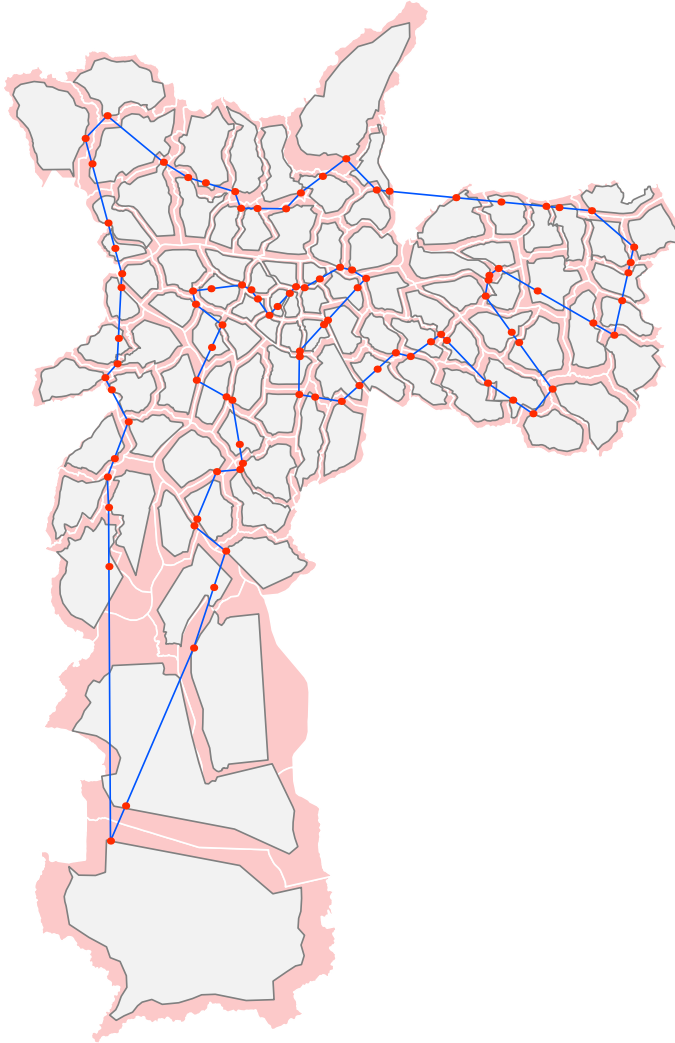
Fig. 6 Final iterate with route length equal to 212,292.01

The initial points are in yellow or light orange and the color changes to red at cycle 13. The evolution of each point is marked with a dotted line whose color changes along with the color of the point. Independently of that, the route determined by the points of each cycle is marked in blue. The route with the lightest blue corresponds to the route given by the initial points (yellow or light orange) and the color of the route gets darker and darker until it reaches the route of cycle 13, marked with the strongest blue. Roughly speaking, the points move a little more in the first 3 cycles, in which the objective function decreases the most, and then there are only small accommodations of the points until the method converges. The authors are aware of
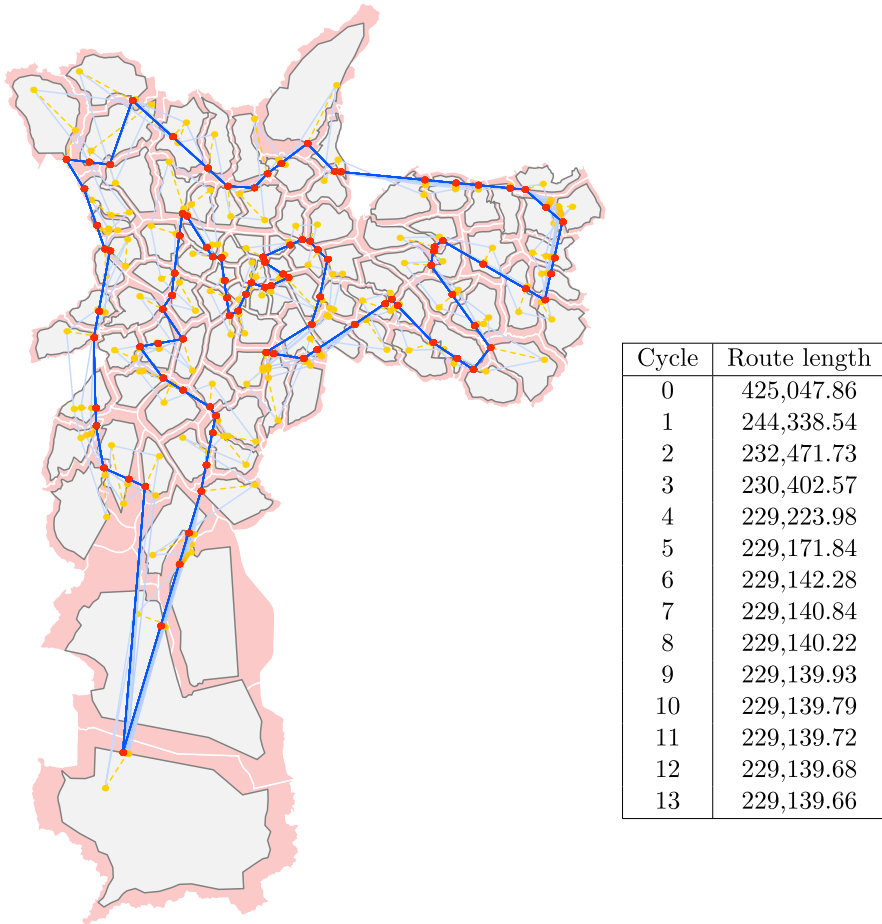
Cycle	Route length
0	425,047.86
1	244,338.54
2	232,471.73
3	230,402.57
4	229,223.98
5	229,171.84
6	229,142.28
7	229,140.84
8	229,140.22
9	229,139.93
10	229,139.79
11	229,139.72
12	229,139.68
13	229,139.66

Fig. 7 Sequence of iterands of the BCD method when applied to random initial points within the polygons and with the order given by the constructive heuristic used to generate the initial point

the difficulty to see the figure clearly; a zoom in the image is recommended to see the details of the evolution of the iterands. In particular, the middle left part clearly shows how the route is being modified as the points move.

7 Conclusions

The framework presented in the present work could be extended in order to consider Taylor-like high-order models [4] satisfying well-established regularity assumptions, as it has been done in [2] for the case of box constraints. However, theoretical results in [2] reveal that using high-order models associated with Coordinate Descent methods is not worthwhile. The reason is that overall computed work is dominated by

the necessity of obtaining fast decrease of the distance between consecutive iterates, whereas high-order models do not help for achieving such purpose.

More interesting, from the practical point of view, is to exploit the particular case in which the constraints that define each Ω_i may be expressed in the form of global inequalities and equalities. (Of course, this is a particular case of the one addressed in this paper that corresponds to set $n_{ops}(i) = 1$ for all $i = 1, \ldots, n_{blocks}$.) In this case the obvious choice for solving subproblems consists of using some well established constrained optimization software. From the theoretical point of view there is nothing to be added, since practical optimization methods for constrained optimization may fail for different reasons, leading the abrupt interruption of the overall optimization process. However, we have no doubts that in many practical problems the standard constrained optimization approach associated with block coordinate descent should be useful.

The reason why, in this paper, we considered feasible sets Ω_i with the local constrained structure defined by open covering sets and constitutive constraints is not strictly related to block coordinate methods. In fact, in contact with several practical problems (an example of which is the one presented in Sect. 6) we observed that the non-global structure of constraints is not unusual and needs specific ways to be handled properly. We believe that different approaches than the one suggested in this paper are possible, most of them motivated by the particular structure of the practical problems at hand. Further research is expected in the following years with respect to this subject.

In the present work, a variant of the classical TSP problem was introduced. In this variant, each city can move freely within a given feasible set. In that context, the proposed BCD method was used to evaluate the merit function: the tour length for a given ordering of the cities. (Recall that the computation of the tour length requires the determination of the optimal position of each city within its own feasible region.) To illustrate the practical use of the BCD, a basic strategy for solving the TSP (local search with neighborhood determined by relocation moves) was implemented. As future work, state-of-the-art methods for solving the classical TSP could be adapted to solve the problem introduced in this work.

Data availability The authors confirm that all data generated or analyzed in the development of this work are adequately included or referenced in the article itself.

References

1. Aarts, E., Lenstra, J. K., (eds.): Local search in combinatorial optimization, Princeton University Press, (2003)
2. Amaral, V.A., Andreani, R., Birgin, E.G., Marcondes, D.S., Martínez, J.M.: On complexity and convergence of high-order coordinate descent algorithms for smooth nonconvex box-constrained minimization. J. Glob. Optim. https://doi.org/10.1007/s10898-022-0.1168-6 (to appear)

3. Applegate, D. L., Bixby, R. E., Chvátal, V., Cook, W. J.: The traveling salesman problem: a computational study, Princeton University Press, (2006)

4. Birgin, E.G., Gardenghi, J.L., Martínez, J.M., Santos, S.A., Toint, Ph.L.: Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. Mathematical Programming **163**, 359–368 (2017)

5. Birgin, E.G., Martínez, J.M.: On regularization and active-set methods with complexity for constrained optimization. SIAM Journal on Optimization **28**, 1367–1395 (2018)

6. Birgin, E.G., Martínez, J.M.: A Newton-like method with mixed factorizations and cubic regularization for unconstrained minimization. Computational Optimization and Applications **73**, 707–753 (2019)

7. Bonettini, S., Prato, M., Begegoldi, S.: A cyclic block coordinate descent method with generalized gradient projections. Applied Mathematics and Computation **286**, 288–300 (2016)

8. Gendreau, M., Hertz, A., Laporte, G.: New insertion and post optimization procedures for the traveling salesman problem. Operations Research **40**, 1086–1094 (1992)

9. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York, NY (2006)

10. Powell, M.J.D.: On search directions for minimization algorithms. Mathematical Programming **4**, 193–201 (1973)

11. Wright, S.J.: Coordinate descent methods. Mathematical Programming **151**, 3–34 (2015)