



# Convergent Inexact Penalty Decomposition Methods for Cardinality-Constrained Problems

Matteo Lapucci<sup>1</sup> · Tommaso Levato<sup>1</sup> · Marco Sciandrone<sup>1</sup>

Received: 21 May 2020 / Accepted: 28 November 2020  
© The Author(s) 2020

## Abstract

In this manuscript, we consider the problem of minimizing a smooth function with cardinality constraint, i.e., the constraint requiring that the  $\ell_0$ -norm of the vector of variables cannot exceed a given threshold value. A well-known approach of the literature is represented by the class of penalty decomposition methods, where a sequence of penalty subproblems, depending on the original variables and new variables, are inexactly solved by a two-block decomposition method. The inner iterates of the decomposition method require to perform exact minimizations with respect to the two blocks of variables. The computation of the global minimum with respect to the original variables may be prohibitive in the case of nonconvex objective function. In order to overcome this nontrivial issue, we propose a modified penalty decomposition method, where the exact minimizations with respect to the original variables are replaced by suitable line searches along gradient-related directions. We also present a derivative-free penalty decomposition algorithm for black-box optimization. We state convergence results of the proposed methods, and we report the results of preliminary computational experiments.

**Keywords** Cardinality constraint · Penalty decomposition method · Nonconvex optimization · Global convergence · Derivative-free optimization

**Mathematics Subject Classification** 90C26 · 90C30 · 65K05

---

Communicated by Benoit Chachuat.

---

✉ Tommaso Levato  
tommaso.levato@unifi.it

<sup>1</sup> Department of Information Engineering, Università degli Studi di Firenze, Via di Santa Marta 3, 50139 Firenze, Italy

## 1 Introduction

In this work, we consider the problem of minimizing a smooth function with a sparsity constraint (*cardinality constraint*). Optimization problems where sparse solutions are sought arise frequently in modern science and engineering. Just as examples, applications of sparse optimization regard compressed sensing in signal processing [1,2], best subset selection [3–6] and sparse inverse covariance estimation [7,8] in statistics, sparse portfolio selection [9] in decision science, neural networks compression in machine learning [10,11].

It is well known that optimization problems involving the  $\ell_0$  norm are  $\mathcal{NP}$ -hard [12,13]. Hence, classes of algorithms have been proposed through the years to approximately solve cardinality-constrained problems. Examples of effective methods are given by the Lasso [14] and other  $\ell_p$ -reformulation approaches [15,16].

Particularly useful algorithms designed to deal with cardinality-constrained optimization problems are the greedy sparse simplex method [17] and the class of penalty decomposition (PD) methods [18]. The former is specifically designed for problems of the form (1), and the latter has been defined to deal with cardinality-constrained problems characterized by the presence of further standard constraints. These methods, based on different approaches, present theoretical convergence properties and are computationally efficient in the solution of cardinality-constrained problems. However, they require to exactly solve at each iteration suitable subproblems (of dimension 1 in the case of the greedy sparse simplex method, and of dimension  $n$  for PD methods). This may be prohibitive when either the objective function is nonconvex or the finite termination of an algorithm applied to a convex subproblem cannot be guaranteed. This latter issue typically occurs when the convex function is not quadratic. Note that there are several applications of sparse optimization involving nonconvex objective functions (see, e.g., [10]).

The aim of the present work is to tackle cardinality-constrained problems by defining convergent algorithms that do not require to compute the exact solution of (possibly nonconvex) subproblems. To this aim, we focus on the approach of the PD methods and we present two contributions:

- (a) the definition of a PD algorithm performing inexact minimizations by an Armijo-type line search [19] along *gradient-related* directions;
- (b) the definition of a derivative-free PD method for sparse black-box optimization.

The two algorithms share the penalty decomposition approach, but differ significantly in the inexact minimization steps and in the definition of the inner stopping criterion. We perform a theoretical analysis of the proposed methods, and we state convergence results that are equivalent to those of the original PD methods [18] but, in general, weaker than those of the greedy sparse simplex method [17]. Finally, we remark that, to our knowledge, convergent derivative-free methods for cardinality-constrained problems were not known, and this makes the derivative-free algorithm proposed in the present work particularly attractive. The paper is organized as follows. In Sect. 2, we address optimality conditions for problem (1); we also describe the PD method originally introduced in [18]. In Sect. 3, we propose a modified version of the PD algorithm and we state global convergence results. In Sect. 4, we present a derivative-

free PD method for black-box optimization and we prove the global convergence of the proposed method. The results of preliminary computational experiments, limited to a class of convex problems, are reported in Sect. 5 and show the validity of the proposed approach. Finally, Sect. 6 contains some concluding remarks.

## 2 Background

In this work, we consider the following optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \|x\|_0 \leq s, \quad (1)$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuously differentiable function,  $\|x\|_0$  is the  $\ell_0$  norm of  $x$ , i.e., the number of its nonzero components, and  $s$  is an integer such that  $0 < s < n$ . Throughout the paper, we make the following assumption.

**Assumption 2.1** The function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuously differentiable and coercive on  $\mathbb{R}^n$ , i.e., for all sequences  $\{x^k\}$  such that  $x^k \in \mathbb{R}^n$  and  $\lim_{k \rightarrow \infty} \|x^k\| = \infty$  we have  $\lim_{k \rightarrow \infty} f(x^k) = \infty$ .

The above assumption implies that problem (1) admits solution. Necessary optimality conditions for problem (1) have been stated in [17], where the basic feasible (BF) property has been introduced. We recall this notion hereafter.

**Definition 2.1** We say that a point  $\bar{x} \in \mathbb{R}^n$  is a *BF-vector*, if:

- when  $\|\bar{x}\|_0 = s$ , it holds  $\nabla_i f(\bar{x}) = 0$  for all  $i$  s.t.  $\bar{x}_i \neq 0$ ;
- when  $\|\bar{x}\|_0 < s$ , it holds  $\nabla_i f(\bar{x}) = 0$  for all  $i = 1, \dots, n$ .

It can be easily shown that if  $x^*$  is an optimal solution of problem (1), then  $x^*$  is a BF-vector.

Necessary optimality conditions for cardinality-constrained problems with additional nonlinear constraints have been studied in [18]. Such conditions have been used to study the convergence of the PD method proposed in the same work. In the case of problem (1), the aforementioned necessary optimality conditions result simplified. In particular, on the basis of the convergence analysis performed in [18], we introduce the following definition.

**Definition 2.2** We say that a point  $\bar{x} \in \mathbb{R}^n$  satisfies *Lu–Zhang first-order optimality conditions* if there exists a set  $I \subseteq \{1, \dots, n\}$  such that  $|I| = s$ ,  $\bar{x}_i = 0$  for all  $i \in \bar{I} = \{1, \dots, n\} \setminus I$  and  $\nabla_i f(\bar{x}) = 0$  for all  $i \in I$ .

If  $x^*$  is an optimal solution, then  $x^*$  satisfies Lu–Zhang first-order optimality conditions [18]. It can be easily verified that a BF-vector satisfies Lu–Zhang conditions. The converse is not necessarily true, i.e., Lu–Zhang conditions are weaker than the optimality conditions expressed by BF property. We show this with the following example.

**Example 2.1** Consider problem (1), letting

$$f(x) = (x_1 - 1)^2 + x_2^2 + (x_3 - 1)^2$$

and  $s = 2$ . The point  $\bar{x} = [1 \ 0 \ 0]$  satisfies Lu–Zhang conditions, but it is not a BF-vector. Indeed, let  $J = \{1, 2\}$ . We have that  $\bar{x}_j = 0$  for all  $j \in \bar{J}$  and  $\nabla_i f(\bar{x}) = 0$  for all  $i \in J$ . Thus,  $\bar{x}$  satisfies Lu–Zhang conditions. On the other hand,  $\|\bar{x}\|_0 < 2$ , and  $\nabla_3 f(\bar{x}) \neq 0$ , i.e., it is not a BF-vector.

## 2.1 The Projection onto the Feasible Set

Consider the problem of computing the orthogonal projection of a vector  $\bar{x}$  onto the feasible set, i.e., the problem

$$\min_x \|x - \bar{x}\|^2 \quad \text{s.t.} \quad \|x\|_0 \leq s. \quad (2)$$

Since the feasible set is not convex, the solution of (2) is not unique. A globally optimal solution can be computed in closed form taking the  $s$  components of  $\bar{x}$  with the largest absolute value [17]. To formally characterize the solution, let us define the index set  $I(x)$  of the largest nonzero variables (in absolute value) at a generic point  $x \in \mathbb{R}^n$ , satisfying the following properties:

$$\begin{aligned} I(x) &\in \arg \max_{S \subseteq \{1, \dots, n\}} |S| \\ \text{s.t.} \quad &|S| \leq s, \quad i \in S \Rightarrow x_i \neq 0, \\ &|x_i| \geq |x_j| \quad \forall i \in S, \forall j \notin S. \end{aligned} \quad (3)$$

In general, the index set  $I(x)$  is not uniquely defined. Also, note that  $I(x) = \{i \in \{1, \dots, n\} : x_i \neq 0\}$  if  $\|x\|_0 \leq s$ .

Then, the solution  $x^*$  of problem (2) is such that

$$x_i^* = \bar{x}_i \text{ for } i \in I(\bar{x}), \quad x_i^* = 0 \text{ for } i \notin I(\bar{x}). \quad (4)$$

## 2.2 The Penalty Decomposition Method

Applying the classical variable splitting technique [20], Problem (1) can be equivalently expressed as

$$\min_{x, y \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \|y\|_0 \leq s, \quad x = y. \quad (5)$$

For simplicity, in the following, we will denote  $Y = \{y \in \mathbb{R}^n : \|y\|_0 \leq s\}$ .

The quadratic penalty function associated with (5) is

$$q_\tau(x, y) = f(x) + \frac{\tau}{2} \|x - y\|^2,$$

where  $\tau > 0$  is the penalty parameter.

In [18], the penalty decomposition (PD) method (see Algorithm 1) was proposed to solve Problem (5). In particular, the approach is that of approximately solving a sequence of penalty subproblems by a two-block decomposition method. The algorithm starts from a point  $(x^0, y^0)$  that is feasible for problem (5). At every iteration, the algorithm performs the block coordinate descent (BCD) method [21,22] w.r.t. the two blocks of variables  $x$  and  $y$ , until an approximate stationary point of the penalty function w.r.t. the  $x$  block is attained. Then, the penalty parameter  $\tau_k$  is increased for the successive iteration, where a higher degree of accuracy is required to approximate a stationary point.

Note that, as discussed in Sect. 2.1, the  $y$ -update step can be performed by computing the closed-form solution of the related subproblem. At the beginning of each iteration, before starting the BCD loop, a test is performed to ensure that the points of the generated sequence belong to a compact level set. This is done in order to guarantee that the sequence generated by the PD method is bounded, so that it admits limit points. In [18] it is proved that each limit point is feasible and satisfies Lu–Zhang conditions.

---

**Algorithm 1:** PenaltyDecomposition

---

```

1 Input:  $\tau_0 > 0, \theta > 1, x^0 = y^0 \in \mathbb{R}^n$  s.t.  $\|x^0\|_0 \leq s$ , a sequence  $\{\varepsilon_k\}$  s.t.  $\varepsilon_k \rightarrow 0$ ,
    $\Gamma \geq \max\{f(x^0), \min_x q_{\tau_0}(x, y^0)\}$ .
2 for  $k = 0, 1, \dots$  do
3    $\ell = 0$ 
4    $u^0 = x^k$ 
5   if  $\min_x q_{\tau_k}(x, y^k) \leq \Gamma$  then
6      $v^0 = y^k$ 
7   else
8      $v^0 = y^0$ 
9   while  $\|\nabla_x q_{\tau_k}(u^\ell, v^\ell)\| > \varepsilon_k$  do
10     $u^{\ell+1} \in \arg \min_u q_{\tau_k}(u, v^\ell)$ 
11     $v^{\ell+1} \in \arg \min_{v \in Y} q_{\tau_k}(u^{\ell+1}, v)$ 
12     $\ell = \ell + 1$ 
13    $\tau_{k+1} = \theta \tau_k$ 
14    $x^{k+1}, y^{k+1} = u^\ell, v^\ell$ 
15 Output: The sequence  $\{x^k\}$ .

```

---

### 3 An Inexact Penalty Decomposition Method

Algorithm 1 has been shown to be effective in practice [18]. However, it requires to compute, in the inner iterations of the block decomposition method, the exact solution of a sequence of subproblems in the  $x$  variables (see steps 5 and 10). This may be

prohibitive when either the objective function is nonconvex or the finite termination of an algorithm applied to a convex subproblem cannot be guaranteed. On the other hand, the convergence analysis performed in [18] is strongly based on the assumption that the global minima of the subproblems in the  $x$  variables are determined. In order to overcome this nontrivial issue by preserving global convergence properties, we propose a modified version of the algorithm, suitable even for problems with nonconvex objective function.

---

**Algorithm 2:** InexactPenaltyDecomposition

---

```

1 Input:  $\tau_0 > 0, \theta > 1, x^0 = y^0 \in \mathbb{R}^n$  s.t.  $\|x^0\|_0 \leq s$ , a sequence  $\{\varepsilon_k\}$  s.t.  $\varepsilon_k \rightarrow 0, \gamma \in ]0, 1[$ ,
    $\beta \in ]0, 1[$ .
2 for  $k = 0, 1, \dots$  do
3    $\ell = 0$ 
4    $\alpha = \text{ArmijoLineSearch}(q_{\tau_k}, x^k, y^k, -\nabla_x q_{\tau_k}(x^k, y^k), \gamma, \beta)$ 
5    $x^{\text{trial}} = x^k - \alpha \nabla_x q_{\tau_k}(x^k, y^k)$ 
6   if  $q_{\tau_k}(x^{\text{trial}}, y^k) \leq f(x^0)$  then
7      $u^0, v^0 = x^k, y^k$ 
8   else
9      $u^0, v^0 = x^0, y^0$ 
10  while  $\|\nabla_x q_{\tau_k}(u^\ell, v^\ell)\| > \varepsilon_k$  do
11     $\alpha_\ell = \text{ArmijoLineSearch}(q_{\tau_k}, u^\ell, v^\ell, -\nabla_x q_{\tau_k}(u^\ell, v^\ell), \gamma, \beta)$ 
12     $u^{\ell+1} = u^\ell - \alpha_\ell \nabla_x q_{\tau_k}(u^\ell, v^\ell)$ 
13     $v^{\ell+1} \in \arg \min_{v \in Y} q_{\tau_k}(u^{\ell+1}, v)$ 
14     $\ell = \ell + 1$ 
15   $\tau_{k+1} = \theta \tau_k$ 
16   $x^{k+1} = u^\ell$ 
17   $y^{k+1} = v^\ell$ 
18 Output: The sequence  $\{x^k\}$ .
```

---

The proposed procedure is described in Algorithm 2. The exact minimization with respect to the  $x$  variables is replaced by an Armijo-type line search along the steepest descent direction of the penalty function. The line search procedure along a descent direction  $d$  is shown in Algorithm 3.

We recall some well-known properties for the Armijo-type line search, later used in the convergence analysis. These results can be found, for instance, in [19].

---

**Algorithm 3:** ArmijoLineSearch

---

```

1 Input:  $g : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}, x, y \in \mathbb{R}^n, d \in \mathbb{R}^n, \gamma \in ]0, 1[, \beta \in ]0, 1[$ .
2 Compute
```

$$\alpha = \max_{j \in \mathbb{N}} \{\beta^j : g(x + \beta^j d, y) \leq g(x, y) + \gamma \beta^j \nabla_x g(x, y)^T d\} \tag{6}$$

```

3 return  $\alpha$ 
```

---

It can be easily seen that the algorithm is well defined, i.e., there exists a finite integer  $j$  such that  $\beta^j$  satisfies the acceptability condition (6). Moreover the following result holds.

**Proposition 3.1** Let  $g : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  be a continuously differentiable function and  $\{x^t, y^t\} \subseteq \mathbb{R}^n \times \mathbb{R}^n$ . Let  $T \subseteq \{0, 1, \dots\}$  be an infinite subset such that

$$\lim_{\substack{t \rightarrow \infty \\ t \in T}} (x^t, y^t) = (\bar{x}, \bar{y}).$$

Let  $\{d^t\}$  be a sequence of directions such that  $\nabla_x g(x^t, y^t)^T d^t < 0$  and assume that  $\|d^t\| \leq M$  for some  $M > 0$  and for all  $t \in T$ . If

$$\lim_{\substack{t \rightarrow \infty \\ t \in T}} g(x^t, y^t) - g(x^t + \alpha_t d^t, y^t) = 0,$$

then we have

$$\lim_{\substack{t \rightarrow \infty \\ t \in T}} \nabla_x g(x^t, y^t)^T d^t = 0.$$

**Remark 3.1** Step 12 of Algorithm 2 can be modified in order to make the algorithm more general. More specifically, the steepest descent direction  $-\nabla_x q_{\tau_k}(u^\ell, v^\ell)$  could be replaced by any gradient-related direction  $d^\ell$ . In this sense, we have the possibility of arbitrarily defining the updated point  $u^{\ell+1}$ , provided that  $q_{\tau_k}(u^{\ell+1}, v^\ell) \leq q_{\tau_k}(u^\ell + \alpha_\ell d^\ell, v^\ell)$ , where  $\alpha_\ell$  is computed by an Armijo line search along the descent direction  $d^\ell$  that, in particular, may be  $-\nabla_x q_{\tau_k}(u^\ell, v^\ell)$ . It can be easily seen that this modification does not spoil the theoretical analysis we are going to carry out hereafter, while it may bring significant benefits from a computational perspective.

**Remark 3.2** As outlined by [18], the stopping condition at line 10 of Algorithm 2 is useful for establishing the convergence properties of the algorithm, but, in practice, different rules could be employed with benefits in terms of efficiency. For example, the progress of the decreasing sequence  $\{q_{\tau_k}(u^\ell, v^\ell)\}$  might be taken into account. As for the main loop, the whole algorithm can be stopped in practice as soon as  $x^k$  and  $y^k$  are sufficiently close.

We now address the properties of the inexact penalty decomposition method. Let us introduce the level set

$$\mathcal{L}_0(f) = \{x : f(x) \leq f(x^0)\}.$$

Note that  $\mathcal{L}_0(f)$  is compact, being  $f$  continuous and coercive on  $\mathbb{R}^n$ . First we show that also  $q_\tau(x, y)$  is a coercive function.

**Lemma 3.1** The function  $q_\tau(x, y)$  is coercive on  $\mathbb{R}^n \times \mathbb{R}^n$ .

**Proof** Let us consider any pair of sequences  $\{x^k\}$  and  $\{y^k\}$  such that at least one of the following conditions holds

$$\lim_{k \rightarrow \infty} \|x^k\| = \infty, \tag{7}$$

$$\lim_{k \rightarrow \infty} \|y^k\| = \infty. \tag{8}$$

Assume by contradiction that there exists an infinite subset  $K \subseteq \{0, 1, \dots\}$  such that

$$\limsup_{\substack{k \rightarrow \infty \\ k \in K}} q_\tau(x^k, y^k) \neq \infty. \tag{9}$$

Suppose first that there exists an infinite subset  $K_1 \subseteq K$  such that

$$\|x^k - y^k\| \leq M, \tag{10}$$

for some  $M > 0$  and for all  $k \in K_1$ . Recalling that  $f$  is coercive on  $\mathbb{R}^n$ , from (7), (8) we have that  $f(x^k) \rightarrow \infty$  for  $k \rightarrow \infty, k \in K_1$ . From (10) we obtain

$$\lim_{\substack{k \rightarrow \infty \\ k \in K_1}} q_\tau(x^k, y^k) = \lim_{\substack{k \rightarrow \infty \\ k \in K_1}} f(x^k) + \frac{\tau}{2} \|x^k - y^k\|^2 = \infty,$$

and this contradicts (9). Then, we must have

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} \|x^k - y^k\| = \infty.$$

As  $f$  is coercive and continuous, it admits minimum over  $\mathbb{R}^n$ . Let  $f^*$  be the minimum value of  $f$ . Thus, we have

$$q_\tau(x^k, y^k) \geq f^* + \frac{\tau}{2} \|x^k - y^k\|^2,$$

which implies that  $q_\tau(x^k, y^k) \rightarrow \infty$  for  $k \rightarrow \infty, k \in K$ .

Then, we can conclude that, for any infinite set  $K$ , we have

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} q_\tau(x^k, y^k) = \infty,$$

and this contradicts (9). □

Now, we can prove that Algorithm 2 is well defined, i.e., that the cycle between step 10 and step 14 terminates in a finite number of inner iterations.

**Proposition 3.2** *Algorithm 2 cannot infinitely cycle between step 10 and step 14, i.e., for each outer iteration  $k \geq 0$ , the algorithm determines in a finite number of inner iterations a point  $(x^{k+1}, y^{k+1})$  such that*

$$\|\nabla_x q_{\tau_k}(x^{k+1}, y^{k+1})\| \leq \epsilon. \tag{11}$$



**Proof** Suppose by contradiction that, at a certain iteration  $k$ , the sequence  $\{u^\ell, v^\ell\}$  is infinite. From the instructions of the algorithm, we have

$$q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq q_{\tau_k}(u^0, v^0).$$

Hence, for all  $\ell \geq 0$ , the point  $\{u^\ell, v^\ell\}$  belongs to the level set

$$\mathcal{L}_0(q_{\tau_k}) = \{(u, v) \in \mathbb{R}^n \times \mathbb{R}^n : q_{\tau_k}(u, v) \leq q_{\tau_k}(u^0, v^0)\}.$$

Lemma 3.1 implies that  $\mathcal{L}_0(q_{\tau_k})$  is a compact set. Therefore, the sequence  $\{u^\ell, v^\ell\}$  admits cluster points. Let  $K \subseteq \{0, 1, \dots\}$  be an infinite subset such that

$$\lim_{\substack{\ell \rightarrow \infty \\ \ell \in K}} (u^\ell, v^\ell) = (\bar{u}, \bar{v}).$$

Recalling the continuity of the gradient, we have

$$\lim_{\substack{\ell \rightarrow \infty \\ \ell \in K}} \nabla_x q_{\tau_k}(u^\ell, v^\ell) = \nabla_x q_{\tau_k}(\bar{u}, \bar{v}).$$

We now show that  $\nabla_x q_{\tau_k}(\bar{u}, \bar{v}) = 0$ . Setting  $d^\ell = -\nabla_x q_{\tau_k}(u^\ell, v^\ell)$  and taking into account the instructions of the algorithm we can write

$$q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq q_{\tau_k}(u^{\ell+1}, v^\ell) = q_{\tau_k}(u^\ell + \alpha_\ell d^\ell, v^\ell) < q_{\tau_k}(u^\ell, v^\ell). \tag{12}$$

Recalling again the continuity of the gradient, we have that  $d^\ell \rightarrow \nabla_x q_{\tau_k}(\bar{u}, \bar{v})$  for  $\ell \in K$  and  $\ell \rightarrow \infty$ , and hence  $\|d^\ell\| \leq M$  for some  $M > 0$  and for all  $\ell \in K$ .

The sequence  $\{q_{\tau_k}(u^\ell, v^\ell)\}$  is monotone decreasing,  $q_{\tau_k}(u, v)$  is continuous, and hence, we have that

$$\lim_{\ell \rightarrow \infty} q_{\tau_k}(u^\ell, v^\ell) = q_{\tau_k}(\bar{u}, \bar{v}).$$

From (12), it follows  $\lim_{\ell \rightarrow \infty} q_{\tau_k}(u^\ell, v^\ell) - q_{\tau_k}(u^\ell + \alpha_\ell d^\ell, v^\ell) = 0$ . Then, the hypotheses of Proposition 3.1 are satisfied and we can write

$$\lim_{\substack{\ell \rightarrow \infty \\ \ell \in K}} \nabla_x q_{\tau_k}(u^\ell, v^\ell)^T d^\ell = \lim_{\substack{\ell \rightarrow \infty \\ \ell \in K}} -\|\nabla_x q_{\tau_k}(u^\ell, v^\ell)\|^2 = 0,$$

which implies that, for  $\ell \in K$  sufficiently large, we have  $\|\nabla_x q_{\tau_k}(u^\ell, v^\ell)\| \leq \epsilon$ , i.e., the stopping criterion of step 10 is satisfied in a finite number of iterations, and this contradicts the fact that  $\{u^\ell, v^\ell\}$  is an infinite sequence.  $\square$

Before stating the global convergence result, we prove that the sequence generated by the algorithm admits limit points and that every limit point  $(\bar{x}, \bar{y})$  is such that  $\bar{x}$  is feasible for the original problem (1).

**Proposition 3.3** *Let  $\{x^k, y^k\}$  be the sequence generated by Algorithm 2. Then,  $\{x^k, y^k\}$  admits cluster points and every cluster point  $(\bar{x}, \bar{y})$  is such that  $\bar{x} = \bar{y}$ , and  $\|\bar{x}\|_0 \leq s$ .*

**Proof** Consider a generic iteration  $k$ . The instructions of the algorithm imply for all  $\ell \geq 0$

$$q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq q_{\tau_k}(u^{\ell+1}, v^\ell) = q_{\tau_k}(u^\ell - \alpha_\ell \nabla_x q_{\tau_k}(u^\ell, v^\ell), v^\ell) \leq q_{\tau_k}(u^\ell, v^\ell),$$

and hence we can write

$$q_{\tau_k}(x^{k+1}, y^{k+1}) \leq q_{\tau_k}(u^0 - \alpha_0 \nabla_x q_{\tau_k}(u^0, v^0), v^0). \tag{13}$$

From the definition of  $(u^0, v^0)$ , we either have  $(u^0, v^0) = (x^k, y^k)$  or  $(u^0, v^0) = (x^0, y^0)$ . In the former case, we have, by the definition of  $x_{\text{trial}}$ , that

$$q_{\tau_k}(u^0 - \alpha_0 \nabla_x q_{\tau_k}(u^0, v^0), v^0) = q_{\tau_k}(x_{\text{trial}}, y^k) \leq f(x^0),$$

where the last inequality holds, as in this case the condition at line 6 is satisfied. In the latter case, we have

$$\begin{aligned} q_{\tau_k}(u^0 - \alpha_0 \nabla_x q_{\tau_k}(u^0, v^0), v^0) &\leq q_{\tau_k}(u^0, v^0) = q_{\tau_k}(x^0, y^0) \\ &= f(x^0) + \frac{\tau_k}{2} \|x^0 - y^0\|^2 = f(x^0). \end{aligned}$$

Then, in both cases from (13) it follows

$$q_{\tau_k}(x^{k+1}, y^{k+1}) \leq f(x^0). \tag{14}$$

We also have

$$f(x^{k+1}) \leq q_{\tau_k}(x^{k+1}, y^{k+1}) = f(x^{k+1}) + \frac{\tau_k}{2} \|x^{k+1} - y^{k+1}\|^2 \leq f(x^0), \tag{15}$$

and hence we can conclude that for all  $k \geq 0$  we have  $f(x^{k+1}) \leq f(x^0)$ . Therefore, the points of the sequence  $\{x^k\}$  belong to the compact set  $\mathcal{L}_0(f)$ , and this implies that  $\{x^k\}$  is a bounded sequence and that, for all  $k \geq 0$ ,  $f(x^k) \geq f^* > -\infty$ ,  $f^*$  being the minimum value of  $f$  over  $\mathbb{R}^n$ .

From (15), dividing by  $\tau_k$ , we get

$$\|x^{k+1} - y^{k+1}\|^2 \leq 2 \frac{f(x^0) - f(x^{k+1})}{\tau_k} \leq 2 \frac{f(x^0) - f^*}{\tau_k}.$$

Taking limits for  $k \rightarrow \infty$ , recalling that  $\tau_k \rightarrow \infty$  for  $k \rightarrow \infty$ , we obtain

$$\lim_{k \rightarrow \infty} \|x^{k+1} - y^{k+1}\| = 0. \tag{16}$$

Therefore, since  $\{x^k\}$  is a bounded sequence, from (16), it follows that  $\{y^k\}$  is bounded, and hence the sequence  $\{(x^k, y^k)\}$  admits cluster points. Let  $(\bar{x}, \bar{y})$  be any cluster point

of  $\{(x^k, y^k)\}$ , i.e., there exists an infinite subset  $K \subseteq \{0, 1, \dots\}$  such that

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} (x^k, y^k) = (\bar{x}, \bar{y}).$$

Again from (16) it follows  $\bar{x} = \bar{y}$ .

Finally, as  $\|y^k\|_0 \leq s$  for all  $k$ , recalling the lower semicontinuity of the  $\ell_0$ -norm  $\|\cdot\|_0$ , we can conclude that  $\|\bar{x}\|_0 = \|\bar{y}\|_0 \leq s$ . □

We are ready to state the global convergence result.

**Theorem 3.1** *Let  $\{x^k, y^k\}$  be the sequence generated by Algorithm 2. Then,  $\{x^k, y^k\}$  admits cluster points and every cluster point  $(\bar{x}, \bar{y})$  is such that  $\bar{x}$  satisfies the Lu–Zhang conditions for problem (1).*

**Proof** Proposition 3.3 implies that the sequence  $\{x^k, y^k\}$  admits cluster points. Let  $K \subseteq \{0, 1, \dots\}$  be an infinite subsequence such that

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} (x^{k+1}, y^{k+1}) = (\bar{x}, \bar{y}).$$

From Proposition 3.3, it follows  $\bar{x} = \bar{y}$  and

$$\|\bar{x}\|_0 \leq s. \tag{17}$$

Using (11) of Proposition 3.2, for all  $k \geq 0$ , we have

$$\|\nabla f(x^{k+1}) + \tau_k(x^{k+1} - y^{k+1})\| \leq \varepsilon_k,$$

so that, taking limits for  $k \in K$  and  $k \rightarrow \infty$ , as  $\varepsilon_k \rightarrow 0$ , we can write

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} \|\nabla f(x^{k+1}) + \tau_k(x^{k+1} - y^{k+1})\| = 0. \tag{18}$$

From the instructions of the algorithm, we have  $y^{k+1} \in \arg \min_{y \in Y} q_{\tau_k}(x^{k+1}, y)$ , i.e.,  $y^{k+1}$  is a solution of the problem

$$\min_y \|y - x^{k+1}\|^2 \quad \text{s.t.} \quad \|y\|_0 \leq s.$$

From (4) it follows

$$y_i^{k+1} = x_i^{k+1} \quad \text{for } i \in I(x^{k+1}), \quad y_i^{k+1} = 0 \quad \text{for } i \notin I(x^{k+1}),$$

where we recall that the index set  $I(x^{k+1})$  contains at most  $s$  elements, those corresponding to the not null components of  $x^{k+1}$  with the largest absolute value.

Note that  $|I(x^{k+1})| < s$  implies  $\|x^{k+1}\|_0 < s$  and hence  $y^{k+1} = x^{k+1}$ . Therefore, we can write

$$-\tau_k(x_i^{k+1} - y_i^{k+1}) = 0 \begin{cases} \forall i \in I(x^{k+1}), & \text{if } |I(x^{k+1})| = s, \\ \forall i \in \{1, \dots, n\}, & \text{if } |I(x^{k+1})| < s. \end{cases} \tag{19}$$

The index set  $I(x^{k+1})$  belongs to the finite set  $\{1, \dots, n\}$ ; therefore, there exists an infinite subset  $K_1 \subseteq K$  such that  $I(x^{k+1}) = I$  for all  $k \in K_1$ .

Let  $I^* = I(\bar{x})$ . We show that  $I^* \subseteq I$ . Indeed, assume by contradiction that there exists  $i \in I^*$  such that  $i \notin I$ . Hence,  $\bar{y}_i = \bar{x}_i \neq 0$ , while  $y_i^{k+1} = 0$  for all  $k \in K$ . This is a contradiction, since  $y^{k+1} \rightarrow \bar{y}$  for  $k \rightarrow \infty, k \in K$ .

Therefore, we have the following possible cases:

$$(i) |I| = s, I = I^*; \quad (ii) |I| < s; \quad (iii) |I| = s, I \supset I^*.$$

We now prove each case separately:

(i) Let  $i \in I = I^*$ ; from (18) we have

$$\lim_{\substack{k \rightarrow \infty \\ k \in K_1}} \nabla_i f(x^{k+1}) + \tau_k(x_i^{k+1} - y_i^{k+1}) = 0,$$

and, using the first condition of (19), it follows  $\tau_k(x_i^{k+1} - y_i^{k+1}) = 0$  for all  $k \in K_1$ . Therefore, recalling the continuity of the gradient, we can write

$$\lim_{\substack{k \rightarrow \infty \\ k \in K_1}} \nabla_i f(x^{k+1}) = \nabla_i f(\bar{x}) = 0 \quad \forall i \in I^*,$$

i.e., Lu–Zhang conditions hold with the set  $I = I^*$ .

(ii) Let  $i \in \{1, \dots, n\}$ ; similarly to the previous case, we have that

$$\lim_{\substack{k \rightarrow \infty \\ k \in K_1}} \nabla_i f(x^{k+1}) + \tau_k(x_i^{k+1} - y_i^{k+1}) = 0,$$

and using the second condition of (19) it follows  $\tau_k(x_i^{k+1} - y_i^{k+1}) = 0$  for all  $k \in K_1$ . Therefore, we obtain

$$\lim_{\substack{k \rightarrow \infty \\ k \in K_1}} \nabla_i f(x^{k+1}) = \nabla_i f(\bar{x}) = 0 \quad \forall i \in \{1, \dots, n\},$$

i.e., Lu–Zhang conditions hold taking any subset of  $\{1, \dots, n\}$  of cardinality  $s$  that contains  $I^*$ .

(iii) Let  $i \in I$ . By the same reasonings of case (i), we can write

$$\lim_{\substack{k \rightarrow \infty \\ k \in K_1}} \nabla_i f(x^{k+1}) = \nabla_i f(\bar{x}) = 0 \quad \forall i \in I,$$

i.e., Lu–Zhang conditions hold with the set  $I$ .

Putting everything together, we have from (i), (ii) and (iii) that Lu–Zhang conditions are always satisfied.  $\square$

As we can see, the proposed inexact version of the algorithm enjoys the same convergence properties as the original, exact one. We also provide, in the following remark, a better characterization of the algorithm, showing that the limit points are often BF-vectors.

**Remark 3.3** We note that, in both case (i) and case (ii) we have that  $\bar{x}$  satisfies the BF optimality conditions. Moreover, note also that:

- If there exists a subsequence  $\hat{K} \subseteq K$  s.t.  $\|x^k\|_0 = \|\bar{x}\|_0$  for all  $k \in \hat{K}$ , the only possible cases are cases (i) and (ii). Indeed, let us consider a further subsequence  $K_2 \subseteq \hat{K}$ , such that  $I(x^{k+1}) = I$  for every  $k \in K_2$ , for some  $I \subset \{1, \dots, n\}$ . We know that  $K_2$  exists and that  $I \supseteq I^*$ . Since  $\|x^{k+1}\|_0 = \|\bar{x}\|_0 \leq s$  for every  $k \in K_2$ ,  $I$  and  $I^*$  are the index sets of nonzero variables of  $x^{k+1}$  and  $\bar{x}$ , respectively, which have the same cardinality. Therefore, it cannot be  $I \supset I^*$ . It follows that  $I = I^*$ , so we fall into either case (i) or case (ii), and thus,  $\bar{x}$  satisfies BF conditions.
- If there exists a subsequence  $\hat{K} \subseteq K$  such that  $\|x^{k+1}\|_0 < s$  for all  $k \in \hat{K}$ , we can again define  $K_2 \subseteq \hat{K}$  such that  $I(x^{k+1}) = I$  for every  $k \in K_2$ , for some  $I \subset \{1, \dots, n\}$ . In this case, we have  $|I| = \|x^{k+1}\|_0 < s$  and case (ii) applies. It follows that  $\bar{x}$  is a BF-vector.

## 4 A Derivative-Free Penalty Decomposition Method

First-order information about the objective function is fundamental for the PD methods we have considered thus far. However, there are applications where the objective function is obtained by direct measurements or it is the result of a complex system of calculations, so that its analytical expression is not available and the computation of its values may be affected by the presence of noise. Hence, in these cases the gradient cannot be explicitly calculated or approximated.

Such lack of information has an impact on the applicability of Algorithm 2. In particular, the  $x$  update step and the inner loop stopping criterion are no more employable as they are.

In this section, we propose a derivative-free modification of Algorithm 2 that, similarly to [23–25], updates  $x$  by line search steps along the coordinate axes and employs a stopping criterion based on the length of such steps.

The derivative-free PD method is described by Algorithm 4. At the  $x$  update step, we employ as search directions the coordinate directions and their opposites. A tentative step length  $\tilde{\alpha}_i$  is associated with each of these directions. At every iteration, all search directions are considered one at a time; a derivative-free line search is performed along each direction, according to Algorithm 5. If the tentative step size does not provide a sufficient decrease, it will be reduced for the next iteration. If, on the other hand, the tentative step size is of sufficient decrease, an extrapolation procedure is carried out;

the tentative step size for that same direction at the successive iteration will be the longest one tried in the extrapolation phase that provides a sufficient decrease. That same step length is also used to move along the considered direction, provided it is large at least  $\varepsilon_k$ ; otherwise, no movement is done along the direction. The inner loop then stops when all tentative step sizes have become smaller than  $\varepsilon_k$ .

---

**Algorithm 4:** DerivativeFreeInexactPenaltyDecomposition

---

```

1 Input:  $\tau_0 > 0, \theta > 1, \delta \in ]0, 1[, \gamma \in ]0, 1[, \sigma > 1, x^0 = y^0 \in \mathbb{R}^n$  s.t.  $\|x^0\|_0 \leq s, \{\varepsilon_k\}$  s.t.  $\varepsilon_k < 1$ 
   for all  $k$  and  $\varepsilon_k \rightarrow 0, \mathcal{D} = \{d_1, \dots, d_{2n}\} = \{e_1, \dots, e_n, -e_1, \dots, -e_n\}$ .
2 for  $k = 0, 1, \dots$  do
3    $\tilde{\alpha}^0 = e \in \mathbb{R}^{2n}$ 
4    $\ell = 0$ 
5    $x^{\text{trial}} = x^k$ 
6   for  $i = 1, \dots, 2n$  do
7      $\hat{\alpha}_i = \text{LineSearch}(q_{\tau_k}(x, y^k), d_i, 1, x^k, \gamma, \sigma)$ 
8     if  $\hat{\alpha}_i > \varepsilon_k$  then
9        $x^{\text{trial}} = x^k + \hat{\alpha}_i d_i$ 
10      break
11  if  $q_{\tau_k}(x^{\text{trial}}, y^k) \leq f(x^0)$  then
12     $u^0, v^0 = x^k, y^k$ 
13  else
14     $u^0, v^0 = x^0, y^0$ 
15  while  $\max_{i=1, \dots, 2n} \{\tilde{\alpha}_i^\ell\} > \varepsilon_k$  do
16     $u^\ell(0) = u^\ell$ 
17    for  $i = 1, \dots, 2n$  do
18       $\alpha_i^\ell = \text{LineSearch}(q_{\tau_k}(u, v^\ell), d_i, \tilde{\alpha}_i^\ell, u^\ell(i-1), \gamma, \sigma)$ 
19      if  $\alpha_i^\ell = 0$  then
20         $\tilde{\alpha}_i^{\ell+1} = \delta \tilde{\alpha}_i^\ell$ 
21      else
22         $\tilde{\alpha}_i^{\ell+1} = \alpha_i^\ell$ 
23      if  $\alpha_i^\ell > \varepsilon_k$  then
24         $u^\ell(i) = u^\ell(i-1) + \alpha_i^\ell d_i^\ell$ 
25      else
26         $u^\ell(i) = u^\ell(i-1)$ 
27     $u^{\ell+1} = u^\ell(2n)$ 
28     $v^{\ell+1} \in \arg \min_{v \in Y} q_{\tau_k}(u^{\ell+1}, v)$ 
29     $\ell = \ell + 1$ 
30   $\tau_{k+1} = \theta \tau_k$ 
31   $x^{k+1} = u^\ell$ 
32   $y^{k+1} = v^\ell$ 
33 Output: The sequence  $\{x^k\}$ .

```

---

**Algorithm 5:** LineSearch

```

1 Input:  $f : \mathbb{R}^n \rightarrow \mathbb{R}, d \in \mathbb{R}^n, \alpha_0 \in \mathbb{R}^+, x \in \mathbb{R}^n, ]0, 1[, \gamma \in ]0, 1[, \sigma > 1.$ 
2  $\alpha = \alpha_0$ 
3 if  $f(x + \alpha d) \leq f(x) - \gamma \alpha^2 \|d\|^2$  then
4   | Let  $\beta = \alpha$ 
5   | repeat
6   |   | Set  $\alpha = \beta$ 
7   |   | Set  $\beta = \sigma \alpha$ 
8   |   until  $f(x + \beta d) > f(x) - \gamma \beta^2 \|d\|^2;$ 
9   |   return  $\alpha$ 
10 Set  $\alpha = 0;$ 
11 return  $\alpha$ 

```

Hereafter, we show that Algorithm 4 enjoys the same convergence properties as Algorithm 2. First, we prove that the line search procedure does not loop infinitely inside our procedure.

**Proposition 4.1** *Algorithm 5 cannot infinitely cycle between steps 5 and 8.*

**Proof** Assume by contradiction that Algorithm 5 does not terminate. Then, for  $j = 0, 1, \dots$ , we have  $f(x + \sigma^j \alpha_0 d) \leq f(x) - \gamma \sigma^{2j} \alpha_0^2 \|d\|^2$ . Taking limits for  $j \rightarrow \infty$ , we obtain that  $f(x + \sigma^j \alpha_0 d) \rightarrow -\infty$ , and this contradicts the fact that  $f$  is bounded below, being  $f$  continuous and coercive.  $\square$

Note that, as shown by Proposition 4.1,  $q_{\tau_k}$  is coercive on  $\mathbb{R}^n \times \mathbb{R}^n$ . We prove that Algorithm 4 is well defined, i.e., the inner loop terminates in finite number of iterations.

**Proposition 4.2** *Algorithm 4 cannot infinitely cycle between steps 15 and 29.*

**Proof** Assume by contradiction that the algorithm loops infinitely. Then, for every  $\ell = 0, 1, \dots$ , there exists  $i \in \{1, \dots, 2n\}$  such that  $\tilde{\alpha}_i^\ell > \varepsilon_k$ , i.e.,

$$\max_{i=1, \dots, 2n} \{\tilde{\alpha}_i^\ell\} > \varepsilon_k. \tag{20}$$

The instructions of the algorithm imply

$$q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq q_{\tau_k}(u^{\ell+1}, v^\ell) \leq q_{\tau_k}(u^\ell(i), v^\ell) \leq q_{\tau_k}(u^\ell(i-1), v^\ell) \leq q_{\tau_k}(u^\ell, v^\ell).$$

Then, the decreasing sequence  $\{q_{\tau_k}(u^\ell, v^\ell)\}$  tends to a finite value, being  $q_{\tau_k}$  continuous and coercive and hence bounded below. For any  $i \in \{1, \dots, 2n\}$ , we can split the sequence of iterations  $\{0, 1, \dots\}$  into two subsequences  $K_1$  and  $K_2$  such that  $K_1 \cup K_2 = \{0, 1, \dots\}$ ,  $K_1 \cap K_2 = \emptyset$ . In particular, we denote by:

- $K_1$  the set of iterations where  $\tilde{\alpha}_i^{\ell+1} = \alpha_i^\ell = \tilde{\alpha}_i^\ell \sigma^s > 0$  for some  $s \geq 0, s \in \mathbb{N}$ ;
- $K_2$  the set of iterations where  $\tilde{\alpha}_i^{\ell+1} = \delta \tilde{\alpha}_i^\ell$  and  $\alpha_i^\ell = 0$ .

Note that  $K_1$  and  $K_2$  cannot both be finite. Then, we analyze the following two cases,  $K_1$  infinite (Case I) and  $K_2$  infinite (Case II).

Case (I). We have

$$\begin{aligned} q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) &\leq q_{\tau_k}(u^{\ell+1}, v^\ell) \leq q_{\tau_k}(u^\ell(i), v^\ell) \leq q_{\tau_k}(u^\ell(i-1), v^\ell) - \gamma(\tilde{\alpha}_i^\ell \sigma^s)^2 \\ &\leq q_{\tau_k}(u^\ell(0), v^\ell) - \gamma(\tilde{\alpha}_i^\ell)^2 = q_{\tau_k}(u^\ell, v^\ell) - \gamma(\tilde{\alpha}_i^\ell)^2. \end{aligned}$$

Taking limits for  $\ell \in K_1, \ell \rightarrow \infty$ , recalling that  $\{q_{\tau_k}(u^\ell, v^\ell)\}$  tends to a finite limit, we get

$$\lim_{\substack{\ell \rightarrow \infty \\ \ell \in K_1}} \tilde{\alpha}_i^\ell = 0, \tag{21}$$

and hence, for  $\ell \in K_1$  sufficiently large, we have  $\tilde{\alpha}_i^\ell \leq \varepsilon_k$ .

Case (II). For every  $\ell \in K_2$ , let  $m_\ell$  be the maximum index on  $\{0, 1, \dots\}$  such that  $m_\ell \in K_1, m_\ell < \ell$  ( $m_\ell$  is the index of the last iteration in  $K_1$  preceding  $\ell$ ). We can assume  $m_\ell = 0$  if the index  $m_\ell$  does not exist, that is,  $K_1$  is empty. Then, we can write  $\tilde{\alpha}_i^\ell = \delta^{\ell-m_\ell} \alpha_i^{m_\ell}$ . As  $\ell \in K_2$  and  $\ell \rightarrow \infty$ , either  $m_\ell \rightarrow \infty$  (if  $K_1$  is an infinite subset) or  $\ell - m_\ell \rightarrow \infty$  (if  $K_1$  is finite). Therefore, (21) and the fact that  $\delta \in (0, 1)$  imply

$$\lim_{\substack{\ell \rightarrow \infty \\ \ell \in K_2}} \tilde{\alpha}_i^\ell = 0.$$

Thus, for  $\ell \in K_2$  sufficiently large, we have  $\tilde{\alpha}_i^\ell \leq \varepsilon_k$ .

We can conclude that  $\lim_{\ell \rightarrow \infty} \tilde{\alpha}_i^\ell = 0$ , so that, recalling that  $i$  is arbitrary, we get  $\max_{i=1, \dots, n} \{\tilde{\alpha}_i^\ell\} \leq \varepsilon_k$  for  $\ell$  sufficiently large, and this contradicts (20).  $\square$

Next, we prove a technical result used later.

**Proposition 4.3** *Assume that the initial step sizes  $\tilde{\alpha}_i^0$ , with  $i = 1, \dots, n$ , are such that  $\tilde{\alpha}_i^0 > \varepsilon_k$  for all  $k$ . Then, for every  $k$  and for every  $i = 1, \dots, 2n$ , there exists  $\rho_i^k \in ]0, c\varepsilon_k[$  such that*

$$\nabla_x q_{\tau_k}(x^{k+1} + \rho_i^k d_i, y^{k+1})^T d_i > -c\varepsilon_k,$$

with  $c = \max\{\sigma, 1/\delta\}$ .

**Proof** Given any iteration  $k$ , let  $\ell$  be the index of the last inner iteration. By definition of  $\ell$ , we must have that  $\tilde{\alpha}_i^{\ell+1} \leq \varepsilon_k$  for all  $i = 1, \dots, n$ . From the instructions of the algorithm this implies that we have  $u^{\ell+1} = u^\ell(2n) = \dots = u^\ell(0) = u^\ell$ , and consequently  $v^{\ell+1} = v^\ell$ . Consider any  $i \in \{1, \dots, 2n\}$ . We have two cases:

1.  $\tilde{\alpha}_i^{\ell+1} = \delta \tilde{\alpha}_i^\ell$ ; in this case,  $\tilde{\alpha}_i^\ell$  did not satisfy the sufficient decrease condition in the LineSearch procedure, i.e.,

$$q_{\tau_k}(u^\ell + \tilde{\alpha}_i^\ell d_i, v^\ell) - q_{\tau_k}(u^\ell, v^\ell) > -\gamma(\tilde{\alpha}_i^\ell)^2. \tag{22}$$



Using the mean value theorem, we can write

$$q_{\tau_k}(u^\ell + \tilde{\alpha}_i^\ell d_i, v^\ell) - q_{\tau_k}(u^\ell, v^\ell) = \tilde{\alpha}_i^\ell \nabla_x q_{\tau_k}(u^\ell + \rho_i^\ell d_i, v^\ell)^T d_i, \tag{23}$$

where  $\rho_i^\ell \in ]0, \tilde{\alpha}_i^\ell[$ . From (22) and (23), it follows:

$$\nabla_x q_{\tau_k}(u^\ell + \rho_i^\ell d_i, v^\ell)^T d_i > -\gamma \tilde{\alpha}_i^\ell = -\frac{\gamma}{\delta} \tilde{\alpha}_i^{\ell+1} \geq -\frac{\gamma}{\delta} \varepsilon_k.$$

Observe that  $\tilde{\alpha}_i^\ell \leq \varepsilon_k/\delta$  and hence  $\rho_i^\ell \in ]0, \varepsilon_k/\delta[$ .

2.  $\tilde{\alpha}_i^{\ell+1} = \alpha_i^\ell$ ; from the instructions of the LineSearch procedure, we get

$$q_{\tau_k}(u^\ell + \sigma \alpha_i^\ell d_i, v^\ell) - q_{\tau_k}(u^\ell, v^\ell) > -\gamma (\sigma \alpha_i^\ell)^2. \tag{24}$$

Using the mean value theorem, we can write

$$q_{\tau_k}(u^\ell + \sigma \alpha_i^\ell d_i, v^\ell) - q_{\tau_k}(u^\ell, v^\ell) = \sigma \alpha_i^\ell \nabla_x q_{\tau_k}(u^\ell + \rho_i^\ell d_i, v^\ell)^T d_i, \tag{25}$$

where  $\rho_i^\ell \in ]0, \sigma \alpha_i^\ell[$ . From (24) and (25), it follows

$$\nabla_x q_{\tau_k}(u^\ell + \rho_i^\ell d_i, v^\ell)^T d_i > -\gamma \sigma \alpha_i^\ell = -\gamma \sigma \tilde{\alpha}_i^{\ell+1} \geq -\gamma \sigma \varepsilon_k.$$

Observe that  $\sigma \alpha_i^\ell = \sigma \tilde{\alpha}_i^{\ell+1} \leq \sigma \varepsilon_k$  and hence  $\rho_i^\ell \in ]0, \sigma \varepsilon_k[$ .

Thus, in both cases we can write

$$\nabla_x q_{\tau_k}(u^\ell + \rho_i^\ell d_i, v^\ell)^T d_i > -c \varepsilon_k, \tag{26}$$

for some  $\rho_i^\ell \in ]0, c \varepsilon_k[$  and  $c = \max\{\sigma, 1/\delta\}$ .

Since  $\tilde{\alpha}_i^{\ell+1} \leq \varepsilon_k$  for all  $i = 1, \dots, 2n$ , from the instructions of the algorithm, we have  $u^{\ell+1} = u^\ell$  and consequently  $v^{\ell+1} = v^\ell$ . Hence, equation (26) holds with  $u^\ell = x^{k+1}$ , and  $v^\ell = y^{k+1}$ . □

Now, we prove that the sequence generated by the algorithm admits limit points and that every limit point is feasible for the original problem.

**Proposition 4.4** *Let  $\{x^k, y^k\}$  be the sequence generated by Algorithm 4. Then,  $\{x^k, y^k\}$  admits cluster points and every cluster point  $(\bar{x}, \bar{y})$  is such that  $\bar{x} = \bar{y}$ , and  $\|\bar{x}\|_0 \leq s$ .*

**Proof** Consider a generic iteration  $k$ . The instructions of the algorithm imply, for all  $\ell \geq 0$ ,

$$q_{\tau_k}(x^{k+1}, y^{k+1}) = q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq q_{\tau_k}(u^{\ell+1}, v^\ell) \leq q_{\tau_k}(u^\ell, v^\ell).$$

From the definition of  $(u^0, v^0)$ , we either have  $(u^0, v^0) = (x^k, y^k)$  or  $(u^0, v^0) = (x^0, y^0)$ . In the former case, for some  $i \in \{1, \dots, 2n\}$  we have, by the

definition of  $x_{\text{trial}}$ , that

$$q_{\tau_k}(u^1, v^0) \leq q_{\tau_k}(u^0 + \hat{\alpha}_i d_i, v^0) = q_{\tau_k}(x_{\text{trial}}, y^k) \leq f(x^0).$$

In the latter case, we have

$$q_{\tau_k}(u^0, v^0) = q_{\tau_k}(x^0, y^0) = f(x^0) + \frac{\tau_k}{2} \|x^0 - y^0\|^2 = f(x^0).$$

Then, in both cases it follows

$$q_{\tau_k}(x^{k+1}, y^{k+1}) \leq f(x^0). \tag{27}$$

The rest of the proof follows the same reasonings used in the proof of Proposition 3.3, starting from the condition corresponding to (27), i.e., condition (14).  $\square$

**Theorem 4.1** *Let  $\{x^k, y^k\}$  be the sequence generated by Algorithm 4. Then,  $\{x^k, y^k\}$  admits cluster points and every cluster point  $(\bar{x}, \bar{y})$  is such that  $\bar{x}$  satisfies the Lu–Zhang conditions for problem (1).*

**Proof** Proposition 4.4 implies that the sequence  $\{x^k, y^k\}$  admits cluster points. Let  $K \subseteq \{0, 1, \dots\}$  be an infinite subsequence such that

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} (x^{k+1}, y^{k+1}) = (\bar{x}, \bar{y}).$$

From Proposition 4.4, it follows  $\bar{x} = \bar{y}$  and  $\|\bar{x}\|_0 \leq s$ . From the instructions of the algorithm, we have  $y^{k+1} \in \arg \min_{y \in Y} q_{\tau_k}(x^{k+1}, y)$ , i.e.,  $y^{k+1}$  is a solution of the problem

$$\min_y \|y - x^{k+1}\|^2 \quad \text{s.t.} \quad \|y\|_0 \leq s.$$

From (4) it follows

$$y_i^{k+1} = x_i^{k+1} \quad \text{for } i \in I(x^{k+1}), \quad y_i^{k+1} = 0 \quad \text{for } i \notin I(x^{k+1}),$$

where we recall that the index set  $I(x^{k+1})$  contains at most  $s$  elements, those corresponding to the not null components of  $x^{k+1}$  with the largest absolute value.

Note that  $|I(x^{k+1})| < s$  implies  $\|x^{k+1}\|_0 < s$  and hence  $y^{k+1} = x^{k+1}$ . Therefore, we can write

$$-\tau_k(x_i^{k+1} - y_i^{k+1}) = 0 \begin{cases} \forall i \in I(x^{k+1}), & \text{if } |I(x^{k+1})| = s, \\ \forall i \in \{1, \dots, n\}, & \text{if } |I(x^{k+1})| < s. \end{cases} \tag{28}$$

The index set  $I(x^{k+1})$  belongs to the finite set  $\{1, \dots, n\}$ ; therefore, there exists an infinite subset  $K_1 \subseteq K$  such that  $I(x^{k+1}) = I$  for all  $k \in K_1$ .

Let  $I^* = I(\bar{x})$ . We have already shown in the proof of Theorem 3.1 that  $I^* \subseteq I$ . We consider the following possible cases:

$$(i) |I| = s, I = I^*; \quad (ii) |I| < s; \quad (iii) |I| = s, I \supset I^*.$$

We now prove each case separately:

- (i) Let  $i \in I = I^*$ ; using the first condition of (28), we get  $\tau_k(x_i^{k+1} - y_i^{k+1}) = 0$  for all  $k \in K_1$ . From Proposition 4.3, recalling that

$$\mathcal{D} = \{d_1, \dots, d_{2n}\} = \{e_1, \dots, e_n, -e_1, \dots, -e_n\},$$

we have that

$$\begin{aligned} \nabla f(x^{k+1} + \rho_i^k e_i)^T e_i &= \nabla_x q_{\tau_k}(x^{k+1} + \rho_i^k e_i, y^{k+1})^T e_i > -c\varepsilon_k, \\ -\nabla f(x^{k+1} + \rho_{i+n}^k e_i)^T e_i &= -\nabla_x q_{\tau_k}(x^{k+1} - \rho_{i+n}^k e_i, y^{k+1})^T e_i > -c\varepsilon_k, \end{aligned}$$

with  $c = \max\{\sigma, 1/\delta\}$ . Taking limits for  $k \rightarrow \infty, k \in K_1$ , recalling that  $\varepsilon_k \rightarrow 0, \rho_i^k, \rho_{i+n}^k \in ]0, c\varepsilon_k[$  and the continuity of the gradient, we get

$$\begin{aligned} \lim_{k \in K_1, k \rightarrow \infty} \nabla f(x^{k+1} + \rho_i^k e_i)^T e_i &= \nabla_i f(\bar{x}) \geq 0, \\ \lim_{k \in K_1, k \rightarrow \infty} -\nabla f(x^{k+1} - \rho_{i+n}^k e_i)^T e_i &= -\nabla_i f(\bar{x}) \geq 0, \end{aligned}$$

from which it follows that  $\nabla_i f(\bar{x}) = 0$  for all  $i \in I^*$ , i.e., Lu-Zhang conditions hold with the set  $I = I^*$ .

- (ii) Let  $i \in \{1, \dots, n\}$ ; the second condition of (28) implies  $\tau_k(x_i^{k+1} - y_i^{k+1}) = 0$  for all  $k \in K_1$ . Similarly to the previous case, we can write

$$\begin{aligned} \nabla f(x^{k+1} + \rho_i^k e_i)^T e_i &= \nabla_x q_{\tau_k}(x^{k+1} + \rho_i^k e_i, y^{k+1})^T e_i > -c\varepsilon_k, \\ -\nabla f(x^{k+1} + \rho_{i+n}^k e_i)^T e_i &= -\nabla_x q_{\tau_k}(x^{k+1} - \rho_{i+n}^k e_i, y^{k+1})^T e_i > -c\varepsilon_k, \end{aligned}$$

with  $c = \max\{\sigma, 1/\delta\}$ , and we can prove

$$\lim_{\substack{k \rightarrow \infty \\ k \in K_1}} \nabla_i f(x^{k+1}) = \nabla_i f(\bar{x}) = 0 \quad \forall i \in \{1, \dots, n\},$$

i.e., Lu-Zhang conditions hold taking any subset of  $\{1, \dots, n\}$  of cardinality  $s$  that contains  $I^*$ .

- (iii) Let  $i \in I$ . By the same reasonings of case (i), we can write

$$\lim_{\substack{k \rightarrow \infty \\ k \in K_1}} \nabla_i f(x^{k+1}) = \nabla_i f(\bar{x}) = 0 \quad \forall i \in I,$$

i.e., Lu-Zhang conditions hold with the set  $I$ .

**Table 1** List of datasets used for experiments on sparse logistic regression

| Dataset                              | N    | n   | Abbreviation |
|--------------------------------------|------|-----|--------------|
| Heart (Statlog)                      | 270  | 25  | heart        |
| Breast Cancer Wisconsin (Prognostic) | 194  | 33  | breast       |
| QSAR Biodegradation                  | 1055 | 41  | biodeg       |
| SPECTF Heart                         | 267  | 44  | spectf       |
| Spambase                             | 4601 | 57  | spam         |
| Adult a2a                            | 2265 | 123 | a2a          |

Putting everything together, we have, from (i), (ii) and (iii), that Lu–Zhang conditions are always satisfied.  $\square$

**Remark 4.1** As in Remark 3.3, if there exists a subsequence  $\hat{K} \subset K$  s.t.  $\|x^k\|_0 = \|\bar{x}\|_0$  for all  $k \in \hat{K}$  or  $\|x^k\|_0 < s$  for all  $k \in \hat{K}$ ,  $\bar{x}$  is a BF-vector.

## 5 Preliminary Computational Experiments

In this section, we show the results of preliminary computational experiments, performed to assess the validity of the proposed approach.

The purpose of these preliminary experiments is to evaluate the inexact minimization strategy of the proposed algorithm (in both its gradient-based and derivative-free versions), compared with the exact minimization approach of the original PD method. To this aim, we consider the problem of sparse logistic regression, where the objective function is convex, but the solution of the subproblems in the  $x$  variables cannot be obtained in closed form, i.e., it requires the adoption of an iterative method.

### Test Problems

The problem of *sparse logistic regression* [26] has important applications, for instance, in machine learning [27, 28]. Given a dataset having  $N$  samples  $\{z^1, \dots, z^N\}$ , with  $n$  features and  $N$  corresponding labels  $\{d_1, \dots, d_N\}$  belonging to  $\{-1, 1\}$ , the sparse logistic regression problem can be formulated as follows:

$$\min_w L(w) = \sum_{i=1}^N \log \left( 1 + \exp \left( -d_i (w^T z^i) \right) \right) \quad \text{s.t.} \quad \|w\|_0 \leq s. \quad (29)$$

The benchmark for this experiment is made up of 18 problems of the form (29), obtained as described hereafter. We employed 6 binary classification datasets, listed in Table 1. All the datasets are from the UCI Machine Learning Repository [29]. For each dataset, we removed data points with missing variables; moreover, we one-hot encoded the categorical variables and standardized the other ones to zero mean and unit standard deviation. For every dataset, we chose 3 different values of  $s$ , in order to define 3 different problems of the form (29). The considered values of  $s$  correspond to the 25%, 50% and 75% of the number  $n$  of features of the dataset.

### Implementation Details

Algorithms 1, 2 and 4 have been implemented in Python 3.6. The algorithms start from the feasible initial point  $x^0 = y^0 = 0 \in \mathbb{R}^n$ . Their common parameters have been set as follows:  $\tau_0 = 1$  and  $\theta = 1.1$ . The three algorithms differ only in the  $x$ -minimization step. Concerning the line search parameters of Algorithm 2, we set  $\gamma = 10^{-5}$  and  $\beta = 0.5$ . As for the derivative-free Algorithm 4, we set  $\delta = 0.5$ ,  $\gamma = 10^{-5}$ ,  $\sigma = 2$ .

The  $x$ -minimization step for Algorithm 1 has been performed by the BFGS [19] solver included in the SciPy library [30]. In particular, the inner iterates of the BFGS solver have been stopped whenever the current point  $u^{\ell+1}$  is such that  $\|\nabla_x q_{\tau_k}(u^{\ell+1}, v^\ell)\| \leq 10^{-5}$ , i.e., when the current point is a good approximation of a stationary point and hence, being the penalty function  $q_{\tau_k}$  strictly convex with respect to  $u$ , of the global minimizer.

For a fair comparison, we employ for the three PD procedures the same stopping criteria for the outer and the inner loop. Specifically, we used the practical stopping criteria proposed in [18]: the inner loop stops when the decrease of the value of the function  $q_{\tau_k}$  is sufficiently small, i.e., when  $q_{\tau_k}(u^\ell, v^\ell) - q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq \epsilon_{\text{in}}$ , where  $\epsilon_{\text{in}} = 10^{-4}$ ; the outer loop is stopped when  $x$  and  $y$  are sufficiently close, i.e., as soon as  $\|x^{k+1} - y^{k+1}\| \leq \epsilon_{\text{out}}$ , where  $\epsilon_{\text{out}} = 10^{-4}$ .

All the experiments have been carried out on an Intel(R) Core(TM) i7- 6700 CPU @ 3.40GHz machine with 4 physical cores (8 threads) and 16 GB RAM.

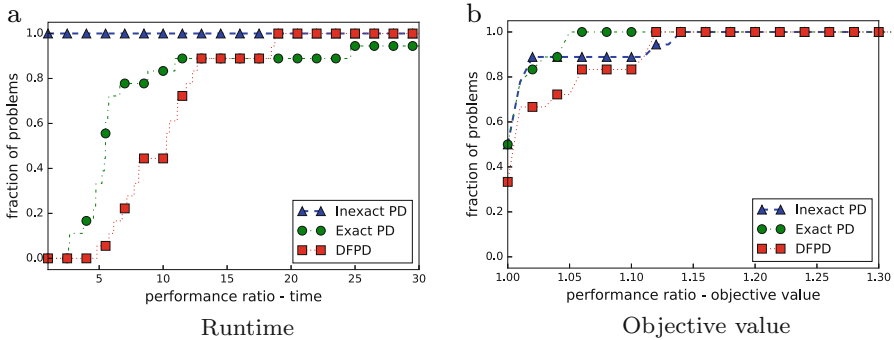
### Numerical Results

The three algorithms, Algorithm 1 called exact PD, Algorithm 2 called inexact PD, and Algorithm 4 called DFPD, have been compared using performance profiles [31]. We recall that, in performance profiles, each curve represents, given a performance metric, the cumulative distribution of the ratio between the result obtained by a solver on an instance of a problem and the best result obtained by any considered solver on that instance. The results of the comparison are shown in Figure 1.

From the results in Figure 1b, we can observe that the performances of the three algorithms, in terms of attained objective function values, are quite close, with rather slight fluctuations. It is worth remarking that different local minima can be attained by different algorithms, even for equal starting points, because of the nonconvex nature of problem (29).

On the other hand, as shown in Figure 1a, the inexact version of the PD algorithm clearly outperforms the other two algorithms in terms of efficiency. This aspect can be valuable in connection with a global optimization strategy, where many local minimizations have to be performed and the availability of an efficient local solver may be useful. The derivative-free algorithm is about an order of magnitude slower than its direct gradient-based counterpart, which is reasonable, considering that the size of the considered problems is quite large in the perspective of derivative-free optimization. In fact, the difference between the speed of gradient-based and derivative-free methods on problems with relatively large size is usually even larger; here, this gap is mitigated, since there is a large set of instructions shared by all the versions of the algorithm.

On the whole, the computational experience, although limited to a single class of problems, confirms the validity of the proposed approach. We remark that we tested



**Fig. 1** Performance profiles of runtime (a) and attained objective value (b) for the exact, inexact and derivative-free penalty decomposition algorithms, on the 18 sparse logistic regression problems

the simplest implementation of the proposed algorithm, that is, performing, in the  $x$ -minimization step, a single line search along the steepest direction. Benefits, in terms of attained function values, could be obtained by performing more iterations of a descent method and by introducing a suitable inner stopping criterion. As already observed, this can be done to improve the effectiveness of the algorithm preserving its global convergence properties.

## 6 Conclusions

In this paper, we have proposed two penalty decomposition-based methods for smooth cardinality-constrained problems. In the first method, based on gradient information, the exact minimization step of the original penalty decomposition method is replaced by line searches along gradient-related directions. Thus, the contribution related to this algorithm lies in the fact that it represents a viable technique, whenever a closed-form solution of the subproblems in the original variables is not available (in both the convex and nonconvex cases). The second method is a derivative-free algorithm for sparse black-box optimization. We remark that, to our knowledge, derivative-free algorithms for cardinality-constrained problems are not known, so that the presented method seems to yield an important contribution in the field of sparse optimization. We state global convergence results for the new penalty decomposition algorithms. We note that the theoretical analysis is quite different from that of the related literature and that it presents substantial differences for the two proposed algorithms. Although the main focus of the work is theoretical, we have reported also the results of preliminary computational experiments performed by the proposed penalty decomposition methods. The obtained results, although limited to a single class of problems, show the validity of the proposed approach. Further work will regard the extension of the presented algorithms to the case of problems with additional equality and inequality constraints, which, similarly to what is done by [18], might be handled by moving them into the quadratic penalty term. Another interesting theoretical investigation might concern the substitution of the line search step by a trust-region framework. Such a modification,

which we consider to be reasonable, would in fact require nontrivial changes to the convergence analysis. Finally, the application of the derivative-free algorithm to real sparse black-box problems would be of great interest.

**Acknowledgements** The authors would like to express their gratitude to the anonymous reviewers and to the associate editor for the very detailed reviews and the constructive and very on point comments that allowed to strongly improve this work.

**Funding** Open Access funding provided by Università degli Studi di Firenze.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Blumensath, T., Davies, M.: Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.* **27**(3), 265–274 (2009)
- Donoho, D.L.: Compressed sensing. *IEEE Trans. Inf. Theory* **52**(4), 1289–1306 (2006)
- Bertsimas, D., King, A., Mazumder, R.: Best subset selection via a modern optimization lens. *Ann. Statist.* **44**, 813–852 (2016)
- Di Gangi, L., Lapucci, M., Schoen, F., Sortino, A.: An efficient optimization approach for best subset selection in linear regression, with application to model selection and fitting in autoregressive time-series. *Comput. Optim. Appl.* (2019)
- Miyashiro, R., Takano, Y.: Mixed integer second-order cone programming formulations for variable selection in linear regression. *Eur. J. Oper. Res.* **247**(3), 721–731 (2015)
- Sato, T., Takano, Y., Miyashiro, R., Yoshise, A.: Feature subset selection for logistic regression via mixed integer optimization. *Comput. Optim. Appl.* **64**(3), 865–880 (2016)
- Rolfs, B., Rajaratnam, B., Guillot, D., Wong, I., Maleki, A.: Iterative thresholding algorithm for sparse inverse covariance estimation. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25*, pp. 1574–1582. Curran Associates, Inc., New York (2012)
- Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics* **9**(3), 432–441 (2008)
- Teng, Y., Yang, L., Yu, B., Song, X.: A penalty palm method for sparse portfolio selection problems. *Optim. Methods Softw.* **32**(1), 126–147 (2017)
- Carreira-Perpinan, M.A., Idelbayev, Y.: “Learning-compression” algorithms for neural net pruning. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8532–8541 (2018)
- Jin, X., Yuan, X., Feng, J., Yan, S.: Training skinny deep neural networks with iterative hard thresholding methods (2016)
- Bienstock, D.: Computational study of a family of mixed-integer quadratic programming problems. *Math. Program.* **74**(2), 121–140 (1996)
- Natarajan, B.K.: Sparse approximate solutions to linear systems. *SIAM J. Comput.* **24**(2), 227–234 (1995)
- Tibshirani, R.: Regression shrinkage and selection via the Lasso. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **58**(1), 267–288 (1996)
- Chartrand, R.: Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.* **14**(10), 707–710 (2007)
- Chen, X., Xu, F., Ye, Y.: Lower bound theory of nonzero entries in solutions of  $\ell_2$ - $\ell_p$  minimization. *SIAM J. Sci. Comput.* **32**(5), 2832–2852 (2010)

17. Beck, A., Eldar, Y.: Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM J. Optim.* **23**(3), 1480–1509 (2013)
18. Lu, Z., Zhang, Y.: Sparse approximation via penalty decomposition methods. *SIAM J. Optim.* **23**(4), 2448–2478 (2013)
19. Bertsekas, D.P.: Nonlinear programming. *J. Oper. Res. Soc.* **48**(3), 334–334 (1997)
20. Jörnsten, K., Näsberg, M., Smeds, P.: Variable Splitting: A New Lagrangean Relaxation Approach to Some Mathematical Programming Models. LiTH MAT R.: Matematiska Institutionen. University of Linköping, Department of Mathematics (1985)
21. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods, vol. 23. Prentice hall Englewood Cliffs, New Jersey (1989)
22. Beck, A., Tetrushvili, L.: On the convergence of block coordinate descent type methods. *SIAM J. Optim.* **23**(4), 2037–2060 (2013)
23. Fasano, G., Liuzzi, G., Lucidi, S., Rinaldi, F.: A linesearch-based derivative-free approach for nonsmooth constrained optimization. *SIAM J. Optim.* **24**(3), 959–992 (2014)
24. Liuzzi, G., Lucidi, S., Piccialli, V.: Exploiting derivative-free local searches in direct-type algorithms for global optimization. *Comput. Optim. Appl.* **65**(2), 449–475 (2016)
25. Lucidi, S., Sciandrone, M.: A derivative-free algorithm for bound constrained optimization. *Comput. Optim. Appl.* **21**(2), 119–142 (2002)
26. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media, New York (2009)
27. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of the zero-norm with linear models and kernel methods. *J. Mach. Learn. Res.* **3**(Mar), 1439–1461 (2003)
28. Bach, F., Jenatton, R., Mairal, J., Obozinski, G., et al.: Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn.* **4**(1), 1–106 (2012)
29. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
30. Virtanen, P., Gommers, R., Oliphant, T.E.: SciPy 1.0 Contributors: SciPy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods* **17**, 261–272 (2020). <https://doi.org/10.1038/s41592-019-0686-2>
31. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**(2), 201–213 (2002)

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