A UNIFYING FRAMEWORK FOR SPARSITY CONSTRAINED OPTIMIZATION*

MATTEO LAPUCCI†, TOMMASO LEVATO†, FRANCESCO RINALDI‡, AND MARCO SCIANDRONE†

Abstract. In this paper, we consider the optimization problem of minimizing a continuously differentiable function subject to both convex constraints and sparsity constraints. By exploiting a mixed-integer reformulation from the literature, we define a necessary optimality condition based on a tailored neighborhood that allows to take into account potential changes of the support set. We then propose an algorithmic framework to tackle the considered class of problems and prove its convergence to points satisfying the newly introduced concept of stationarity. We further show that, by suitably choosing the neighborhood, other well-known optimality conditions from the literature can be recovered at the limit points of the sequence produced by the algorithm. Finally, we analyze the computational impact of the neighborhood size within our framework and in the comparison with some state-of-the-art algorithms, namely, the Penalty Decomposition method and the Greedy Sparse-Simplex method. The algorithms have been tested using a benchmark related to sparse logistic regression problems.

Key words. sparsity constrained problems, optimality conditions, stationarity, numerical methods, asymptotic convergence, sparse logistic regression

AMS subject classifications. 90C30, 90C46, 65K05

1. Introduction.

We consider the following sparsity constrained problem:

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{s.t.} & \|x\|_0 \leq s, \\ & x \in X, \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, $X \subseteq \mathbb{R}^n$ is a closed and convex set, and $s < n$ is a properly chosen integer value. We further use $\mathcal{X}$ to indicate the overall feasible set $X \cap \{x \in \mathbb{R}^n \mid \|x\|_0 \leq s\}$.

Problem (1.1) has a wide range of applications, from subset selection in regression [25] and the compressed sensing technique used in signal processing [12] to portfolio optimization [7, 26]. Such a problem can be reformulated into equivalent different mixed-integer problems and is known to be $\mathcal{NP}$-hard [7, 27, 28].

The approaches proposed in the literature for the solution of problem (1.1) include: exact methods (see, e.g., [6, 7, 28, 29]) typically based on branch-and-bound or branch-and-cut strategies; methods that handle suitable reformulations of the problem based on orthogonality constraints (see, e.g., [9, 10, 11, 13]); penalty decomposition methods, where penalty subproblems are solved by a block coordinate descent method [20, 23]; methods that identify points satisfying tailored optimality conditions related to the problem [3, 4]; heuristics like evolutionary algorithms [1], particle swarm methods [8, 15], genetic algorithms, tabu search and simulated annealing [14], and also neural networks [18].

We observe that problem (1.1) is generally hard to solve because both the objective function and the feasible set (due to the combinatorial nature of the sparsity

*Submitted to the editors DATE.

†Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Via di Santa Marta 3, 50139 Firenze, Italy (matteo.lapucci@unifi.it, tommaso.levato@unifi.it, marco.sciandrone@unifi.it).

‡Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Via Trieste 63, 35121 Padova, Italy (rinaldi@math.unipd.it).

constraint) are nonconvex. The inherently combinatorial flavor of the given problem makes the definition of proper optimality conditions and, consequently, the development of algorithms that generate points satisfying those conditions a challenging task. A number of ways to address these issues are proposed in the literature (see, e.g., [3, 4, 11, 20, 23]). However, some of the optimality conditions proposed do not fully take into account the combinatorial nature of the problem, whereas some of the corresponding algorithms [3, 23] require to exactly solve a sequence of nonconvex subproblems and this may be practically prohibitive. Moreover, due to the theoretical tools involved in the analysis, it is anyway not easy to relate the different approaches with each other.

In this paper, we hence give a unifying view on this matter. More specifically, we consider the mixed-integer reformulation of problem (1.1) described in [11] and use it to define a suitable optimality condition. This condition is then embedded into an algorithmic framework aimed at finding points satisfying the resulting optimality criterion. The algorithm combines inexact minimizations with a strategy that explores tailored neighborhoods of a given feasible point. Those features make it easy to handle the nonconvexity in both the objective function and the feasible set also from a practical point of view. We prove the convergence of the algorithmic scheme, establishing that its limit points satisfy the specific optimality condition. We then show that different conditions proposed in the literature (see, e.g., [3, 11, 23]) can be easily derived from ours. We finally perform some numerical tests on sparse logistic regression in order to show that the devised method is also computationally viable.

The paper is organized as follows: in section 2, we provide basic definitions and preliminary results related to optimality conditions of problem (1.1). In section 3, we describe our proposed algorithmic framework and show (subsection 3.1) the convergence analysis without constraint qualifications. In section 4, we analyze the asymptotic convergence properties of the algorithm when constraint qualifications hold. Finally, we report numerical experiments in section 5 and give some concluding remarks in section 6. We also provide in Appendix A some insights on the relationship between classical stationarity conditions for convex problems with and without constraints qualifications.

2. Basic definitions and preliminary results. Even though problem (1.1) is a continuous optimization problem, it has an intrinsic combinatorial nature and in applications the interest often lies in finding a good, possibly globally optimal configuration of active variables. Being (1.1) a continuous problem, $x^* \in \mathcal{X}$ is a local minimizer if there exists an open ball $\mathcal{B}(x^*, \epsilon)$ such that $f(x^*) = \min\{f(x) \mid x \in \mathcal{X} \cap \mathcal{B}(x^*, \epsilon)\}$. In some works from the literature (e.g., [11, 23]) necessary conditions of local optimality have been proposed. However, for this particular problem every local minimizer for a fixed active set of s variables is a local minimizer of the given problem. Hence the number of local minimizers grows as fast as $\binom{n}{s}$ and is thus of low practical usefulness.

In [3, 4], the authors propose necessary conditions for global optimality that go beyond the concept of local minimum described above, thus allowing to consider possible changes to the structure of the support set, and reducing the pool of optimal candidates. However, these conditions are either tailored to the "unconstrained case", or limited to moderate changes in the support, or involve hard operations, such as exact minimizations or projections onto nonconvex sets.

In order to introduce a general and affordable necessary optimality condition that also takes into account the combinatorial nature of the problem, we consider in our

analysis the equivalent reformulation of problem (1.1) described in [11]:

$$\begin{aligned} \min_{x,y}\ & f(x) \\ \text{s.t. }\ & e^\top y \geq n - s, \\ & x_i y_i = 0, \quad \forall\, i = 1, \dots, n, \\ & x \in X, \\ & y \in \{0,1\}^n. \end{aligned} \tag{2.1}$$

From here onwards, we will use the following notation:

$$\begin{aligned} \mathcal{Y} &= \{y \mid y \in \{0,1\}^n, e^\top y \geq n - s\}, \\ \mathcal{X}(y) &= \{x \in X \mid x_i y_i = 0\ \forall\, i = 1, \dots, n\}. \end{aligned}$$

We further define the support set of a vector z by

$$I_1(z) = \{i \mid z_i \neq 0\},$$

while its complement is defined by

$$I_0(z) = \{i \mid z_i = 0\}.$$

Moreover, we recall the concept of super support set [4]:

DEFINITION 2.1. *Let us consider a feasible point z for problem (1.1). A set $J \subset \{1, \dots, n\}$ is called super support of z if it is such that $|J| = s$ and $I_1(z) \subseteq J$.*

We denote by z_I the subvector of z identified by the components contained in an index set I. We also denote by Π_C the orthogonal projection operator over the closed convex set C. We notice that given a feasible point (x, y) of problem (2.1), the components $I_0(y)$ give an *active subspace* for x, i.e., those components identify the subspace where the nonzero components of x lay. We thus have that $I_1(x) \subseteq I_0(y)$.

Nonlinear mixed-integer programs have been characterized exploiting the notion of *neighborhood* [21, 24]. Given a feasible point (x, y), a discrete neighborhood $\mathcal{N}(x, y)$ is a set of feasible points that are close, to some extent, to (x, y) and that contains (x, y) itself.

We introduce here an example of tailored neighborhood for problem (2.1) that can be implemented at a reasonable computational cost. Such a neighborhood will also help us to relate our analysis to the other theoretical tools available in the literature.

DEFINITION 2.2. *Let $d_H : \{0,1\}^n \times \{0,1\}^n \to \mathbb{N}$ denote the Hamming distance. Moreover, let $J(y, \hat{y}) = \{i \mid y_i \neq \hat{y}_i\}$ and let $H_{J(y,\hat{y})}(\cdot)$ be a function such that $\hat{x} = H_{J(y,\hat{y})}(x)$ is defined as*

$$\begin{cases} \hat{x}_h = 0 & \text{if } h \in J(y, \hat{y}) \\ \hat{x}_h = x_h & \text{otherwise} \end{cases}$$

Then, given $\rho \in \mathbb{N}$, the neighborhood is

$$\mathcal{N}_\rho(x, y) = \left\{(\hat{x}, \hat{y}) \mid e^\top \hat{y} \geq n - s,\ d_H(\hat{y}, y) \leq \rho,\ \hat{x} = H_{J(y,\hat{y})}(x)\right\}. \tag{2.2}$$

We notice that this particular definition of neighborhood allows to take into account the potential "change of status" of up to ρ variables in the vector $\hat{y}$ defining an active subspace.

EXAMPLE 1. *Consider the problem (2.1) with $n = 3$ and $s = 2$ and let $\rho = 2$. Let (x, y) be a feasible point defined as follows*

$$(x, y) = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

The neighborhood $\mathcal{N}_\rho(x, y)$ is given by

$$\mathcal{N}_2(x,y) = \left\{ \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$$

Now, a notion of local optimality for problem (2.1), depending on the neighborhood $\mathcal{N}(x, y)$, can be introduced:

DEFINITION 2.3. *A point $(x^*, y^*) \in \mathcal{X}(y^*) \times \mathcal{Y}$ is a local minimizer of problem (2.1) if there exists an $\epsilon > 0$ such that and for all $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$ it holds*

$$f(x^*) \leq f(x) \quad \forall\, x \in \mathcal{B}(\hat{x}, \epsilon) \cap X(\hat{y}).$$

Note that in the above definition the continuous nature of the problem, expressed by the variables x, is taken into account by means of the standard ball $\mathcal{B}(\hat{x}, \epsilon)$. The given definition clearly depends on the choice of the discrete neighborhoods. A larger neighborhood $\mathcal{N}(x^*, y^*)$ should give a better local minimizer, but the computational effort needed to locate the solution may increase.

Inspired by the definition of local optimality for problem (2.1), we introduce a necessary condition of global optimality for problem (1.1) that allows to take into account possible, beneficial changes of the support and that hence properly captures, from an applied point of view, the essence of the problem.

Such a condition relies on the use of stationary points related to continuous problems obtained by fixing the binary variables in problem (2.1), i.e., for a fixed $\bar{y} \in \mathcal{Y}$,

$$\begin{aligned} &\min\ f(x) \\ &\text{s.t. } x \in \mathcal{X}(\bar{y}). \end{aligned} \tag{2.3}$$

DEFINITION 2.4. *A point $x^* \in \mathcal{X}$ is called an $\mathcal{N}$-stationary point, if there exists an $y^* \in \mathcal{Y}$ such that*

(i) (x^*, y^*) is feasible for problem (2.1);

(ii) the point x^* is a stationary point of the continuous problem

$$\begin{aligned} &\min\ f(x) \\ &s.t.\ x \in \mathcal{X}(y^*); \end{aligned}$$

(iii) every $(\hat{x}, \hat{y}) \in \mathcal{N}_\rho(x^*, y^*)$ satisfies $f(\hat{x}) \geq f(x^*)$ and if $f(\hat{x}) = f(x^*)$, the point $\hat{x}$ is a stationary point of the continuous problem

$$\begin{aligned} &\min\ f(x) \\ &s.t.\ x \in \mathcal{X}(\hat{y}). \end{aligned}$$

It is easy to see that the following result stands:

THEOREM 2.5. *Let x^* be a minimum point of problem (1.1). Then x^* is an $\mathcal{N}$-stationary point.*

We will show later in this work that the definition of $\mathcal{N}$-stationariy allows to retrieve in a unified view most of the known optimality conditions, if a suitable neighborhood $\mathcal{N}$ is employed.

In Definition 2.4 we generically refer to stationary points of problem (2.3), namely, to points satisfying suitable optimality conditions. Then, concerning the assumptions on the feasible set $\mathcal{X}(\bar{y})$, we distinguish the two cases:

(i) no constraint qualifications hold;
(ii) constraint qualifications are satisfied and the usual KKT theory can be applied.

In case (i), we will refer to the following definition of stationary point of problem (2.3).

DEFINITION 2.6. *Given $\bar{y} \in \mathcal{Y}$ and $\bar{x} \in \mathcal{X}(\bar{y})$, we say that $\bar{x}$ is a stationary point of problem (2.3) if and only if*

$$\bar{x} = \Pi_{\mathcal{X}(\bar{y})} \left[\bar{x} - \nabla f(\bar{x}) \right].$$

We notice that $\mathcal{X}(\bar{y})$ is a convex set when X is convex, then the condition given above is a classic stationarity condition for the problem (2.3). Case (ii) will be considered later.

3. Algorithmic framework. Here, we discuss an algorithmic framework for the solution of problem (1.1) that exploits the reformulation given in problem (2.1). The proposed approach is somehow related to classic methods for mixed variable programming proposed in the literature (see, e.g., [21, 24]). Roughly speaking, the approach is based at each iteration on the definition of a suitable neighborhood $\mathcal{N}(x^k, y^k)$ of the current point (x^k, y^k) and on exploratory moves with respect to the continuous variables around the points of the neighborhood.

Concerning the exploration move, it is a local search performed by an Armijo-type line search along the projected gradient direction. The procedure is formalized in Algorithm 3.1.

For any point $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(\tilde{x}^k, y^k)$ that is not significantly worse (in terms of the objective value) than the current candidate, we perform a local continuous search around $\hat{x}^k$; we skip to the following iteration as soon as a point providing a sufficient decrease of the objective value is found. The algorithm, which we refer to as Sparse Neighborhood Search (SNS) is formally defined in Algorithm 3.2.

Algorithm 3.1 Projected-Gradient Line Search (PGLS)

input: $y \in \mathcal{Y}, x \in \mathcal{X}(y), \gamma \in (0, \frac{1}{2}), \delta \in (0,1), \alpha = 1$.
Step 1: Set $\hat{x} = \Pi_{\mathcal{X}(y)} \left[x - \nabla f(x) \right]$, $d = \hat{x} - x$.
Step 2: If

$$f(x + \alpha d) \leq f(x) + \gamma \alpha \nabla f(x)^\top d,$$

set $\tilde{x} = x + \alpha d$ and exit.
Step 3: Set $\alpha = \delta \alpha$ and go to Step 2.

Algorithm 3.2 Sparse Neighborhood Search (SNS)

input: $y^0 \in \mathcal{Y}, x^0 \in \mathcal{X}(y^0), \xi \geq 0, \theta \in (0,1), \eta_0 > 0, \mu_0 > 0, \delta \in (0,1)$.
Step 0: Set $k = 0$.
Step 1: Compute $\tilde{x}^k$ by PGLS(x^k, y^k).
Step 2: Define $W_k = \{(x,y) \in \mathcal{N}(\tilde{x}^k, y^k) \mid f(x) \leq f(\tilde{x}^k) + \xi\}$.
2.1: If $W_k \neq \emptyset$, choose $(x', y') \in W_k$, set $j = 1, x^j = x'$. Otherwise, go to Step 3.
2.2: Compute x^{j+1} by PGLS(x^j, y').
2.3: If $f(x^{j+1}) \leq f(\tilde{x}^k) - \eta_k$, set $x^{k+1} = x^{j+1}, y^{k+1} = y', \eta_{k+1} = \eta_k$ and go to Step 4.
2.4: If $\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\| > \left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\| + \mu_k$, set $j = j + 1$ and go to 2.2. Otherwise, set $W_k = W_k \setminus \{(x', y')\}$ and go to 2.1.
Step 3: Set $x^{k+1} = \tilde{x}^k, y^{k+1} = y^k$. If $f(x^{k+1}) \leq f(x^k) - \eta_k$, set $\eta_{k+1} = \eta_k$. Otherwise set $\eta_{k+1} = \theta\eta_k$.
Step 4: Set $\mu_{k+1} = \delta\mu_k$, $k = k + 1$ and go to Step 1.

3.1. Convergence analysis. In this section, we prove a set of results concerning the properties of the sequences produced by Algorithm 3.2. Note that in this Section we employ the concept of stationarity (A.2). First, we state some suitable assumptions.

ASSUMPTION 1. *The gradient $\nabla f(x)$ is Lipschitz-continuous, i.e., there exists a constant $L > 0$ such that*

$$\|\nabla f(x) - \nabla f(\bar{x})\| \leq L \|x - \bar{x}\|$$

for all $x, \bar{x} \in \mathbb{R}^n$.

ASSUMPTION 2. *Given $y^0 \in \mathcal{Y}$, $x^0 \in \mathcal{X}(y^0)$ and a scalar $\xi > 0$, the level set*

$$\mathcal{L}(x^0, y^0) = \{(x,y) \in \mathcal{X}(y) \times \mathcal{Y} \mid f(x) \leq f(x^0) + \xi\}$$

is compact.

The crucial point in the proposed framework is choosing suitable discrete neighborhoods. First, note that when we deal with both continuous and integer variables, the usual notion of convergence to a point needs to be tweaked. In particular, we have the following definition.

DEFINITION 3.1. *A sequence $\{(x^k, y^k)\}$ converges to a point $(\bar{x}, \bar{y})$ if for any $\epsilon > 0$ there exists an index k_ϵ such that for all $k \geq k_\epsilon$ we have that $y^k = \bar{y}$ and $\|x^k - \bar{x}\| < \epsilon$.*

To ensure convergence to meaningful points, we need a "continuity" assumption on the discrete neighborhoods we explore.

ASSUMPTION 3. *Let $\{(x^k, y^k)\}$ be a sequence converging to $(\bar{x}, \bar{y})$. Then, for any $(\hat{x}, \hat{y}) \in \mathcal{N}(\bar{x}, \bar{y})$, there exists a sequence $\{(\hat{x}^k, \hat{y}^k)\}$ converging to $(\hat{x}, \hat{y})$ such that $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$.*

The assumption above is a mild continuity assumption on the discrete neighborhoods and is equivalent to the lower semicontinuity of a point-to-set function as defined in [5]. Next, we properly define the discrete neighborhood used in our algorithmic framework.

Now, a discrete neighborhood, by definition, is a set of feasible points. In the case when $\mathcal{X} \subset \mathbb{R}^n$, zeroing variables may result in points that are not feasible. For

this reason, we initially consider an easier version of problem (2.1) where $\mathcal{X} = \mathbb{R}^n$. In this case the neighborhood $\mathcal{N}_\rho$ defined in (2.2) contains feasible points. Moreover, it satisfies Assumption 3, as stated here below.

PROPOSITION 3.2. *The point-to-set map $\mathcal{N}_\rho(x, y)$ defined in Definition 2.2 satisfies Assumption 3.*

Proof. Let $\{x^k, y^k\}$ be a sequence convergent to $\{\bar{x}, \bar{y}\}$. Then, for any $\epsilon > 0$, there exists k_ϵ such that $y^k = \bar{y}$ and $\|x^k - \bar{x}\| \leq \epsilon$ for all $k > k_\epsilon$. Let $(\hat{x}, \hat{y}) \in \mathcal{N}_\rho(\bar{x}, \bar{y})$. Since $y^k = \bar{y}$ for k sufficiently large, $\{y \mid e^\top y \geq n - s,\ d_H(y, y^k) \leq \rho\} = \{y \mid e^\top y \geq n - s,\ d_H(y, \bar{y}) \leq \rho\}$, hence $\hat{y} \in \{y \mid d_H(y, y^k) \leq \rho\}$ for all k.

Let us then consider the sequence $\{\hat{x}^k, \hat{y}^k\}$ where $\hat{y}^k = \hat{y}$ and $\hat{x}^k = H_{J(y^k, \hat{y})}(x^k)$. We can observe that $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}_\rho(x^k, y^k)$. Now, let $j \in \{1, \ldots, n\}$. The set $J(y^k, \hat{y}^k) = J(\bar{y}, \hat{y}) = J$ is constant for k sufficiently large.

If $j \notin J$, we have

$$\lim_{k\to\infty} \hat{x}_j^k = \lim_{k\to\infty} x_j^k = \bar{x}_j = \hat{x}_j.$$

On the other hand, if $j \in J$, $\hat{x}_j^k = 0$ and $\hat{x}_j = 0$. Hence

$$\lim_{k\to\infty} \hat{x}^k = \hat{x}$$

and we thus get the thesis. ∎

To generalize the previous proposition to the case where $\mathcal{X} \subset \mathbb{R}^n$, we can replace each $(\tilde{x}, \tilde{y}) \in \mathcal{N}_\rho(x, y)$ with the point $(\hat{x}, \hat{y})$, where $\hat{y} = \tilde{y}$ and $\hat{x} = \Pi_{\mathcal{X}(\hat{y})}(\tilde{x})$. In other words, first we change the structure of the active set, then we project the x part onto $\mathcal{X}(\hat{y})$, which is a convex set. In the following, we will refer to this new discrete neighborhood with $\mathcal{N}_{\mathcal{C}\rho}(x, y)$.

PROPOSITION 3.3. *Let $\{(x^k, y^k)\}$ be a sequence converging to $(\bar{x}, \bar{y})$. Then, the neighborhood $\mathcal{N}_{\mathcal{C}\rho}(\bar{x}, \bar{y})$ satisfies Assumption 3.*

Proof. The proof follows exactly as in Proposition 3.2, recalling the continuity of the projection operator $\Pi_{\mathcal{X}(\hat{y})}$. ∎

Before turning to the convergence analysis of the algorithm, we prove a further useful preliminary result concerning the neighborhood $\mathcal{N}_\rho$. Notice that this result can be easily extended to $\mathcal{N}_{\mathcal{C}\rho}$. In order to avoid getting a too much cumbersome notation, we will always refer to $\mathcal{N}_\rho$ from now on, even when dealing with additional constraints.

LEMMA 3.4. *Let $y \in \mathcal{Y}$ and $x \in \mathcal{X}(y)$ with $\delta = \|x\|_0$. Let us consider the set*

$$\bar{\mathcal{N}}(x) = \{(\hat{x}, \hat{y}) \mid \hat{x} = x,\ e^\top \hat{y} = n - s,\ I_0(\hat{y}) \supseteq I_1(x)\ \}.$$

We have that

$$\bar{\mathcal{N}}(x) \subseteq \mathcal{N}_\rho(x, y),$$

when $\rho \geq 2(s - \delta)$.

Proof. Let $(\hat{x}, \hat{y})$ be any point in $\bar{\mathcal{N}}(x)$. From the feasibility of (x, y) we have

$$\delta \leq I_0(y) \leq s \qquad n - s \leq I_1(y) \leq n - \delta. \tag{3.1}$$

Moreover, from the definition of $\bar{\mathcal{N}}(x)$, we have

$$I_0(\hat{y}) = s \qquad I_1(\hat{y}) = n - s.$$

Now, it is easy to see that

$$d_H(y,\hat{y}) = n - |I_0(y) \cap I_0(\hat{y})| - |I_1(y) \cap I_1(\hat{y})|. \tag{3.2}$$

We can note that, since $I_0(y) \supseteq I_1(x)$ and $I_0(\hat{y}) \supseteq I_1(x)$, it has to be $I_0(y) \cap I_0(\hat{y}) \supseteq I_1(x)$. Therefore

$$|I_0(y) \cap I_0(\hat{y})| \geq |I_1(x)| = \delta. \tag{3.3}$$

We can now turn to $I_1(y) \cap I_1(\hat{y})$. Since the latter set can be equivalently written, by De Morgan's law, as $\{1,\dots,n\} \setminus (I_0(y) \cup I_0(\hat{y}))$, we can obtain

$$\begin{aligned}
|I_1(y) \cap I_1(\hat{y})| &= |\{1,\dots,n\} \setminus (I_0(y) \cup I_0(\hat{y}))| \\
&= n - |I_0(y) \cup I_0(\hat{y})| \\
&= n - (|I_0(y)| + |I_0(\hat{y})| - |I_0(y) \cap I_0(\hat{y})|) \\
&= n - |I_0(y)| - s + |I_0(y) \cap I_0(\hat{y})| \\
&\geq n - s - s + \delta \\
&= n - 2s + \delta,
\end{aligned}$$

where the second last inequality comes from (3.1) and (3.3). Putting everything together back in (3.2), we get

$$d_H(y,\hat{y}) \leq n - \delta - n + 2s - \delta = 2(s-\delta).$$

Taking into account that $\rho \geq 2(s-\delta)$ in the definition of $\mathcal{N}_\rho(x,y)$, we obtain

$$(\hat{x},\hat{y}) \in \mathcal{N}_\rho(x,y),$$

thus getting the desired result. ∎

We can now focus on the algorithms. First, we prove a property of Algorithm 3.1 that will play an important role in the convergence analysis of Algorithm 3.2.

PROPOSITION 3.5. *Given a feasible point (x,y), Algorithm 3.1 produces a feasible point $(\tilde{x},y)$ such that*

$$f(\tilde{x}) \leq f(x) - \sigma\left(\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|\right),$$

where the function $\sigma(\cdot) \geq 0$ is such that if $\sigma\left(t^h\right) \to 0$ then $t^h \to 0$.

Proof. By definition, $d = \hat{x} - x$, where $\hat{x} = \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]$. By the properties of the projection operator, we can write

$$(x - \nabla f(x) - \hat{x})^\top (x - \hat{x}) \leq 0,$$

which, with simple manipulations, implies that

$$\nabla f(x)^\top d \leq -\|d\|^2 = -\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|^2. \tag{3.4}$$

By the instruction of the algorithm, either $\alpha = 1$ or $\alpha < 1$.
If $\alpha = 1$, then $\tilde{x} = x + d$ satisfies

$$f(\tilde{x}) \leq f(x) + \gamma \nabla f(x)^\top d \leq f(x) - \gamma\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|^2. \tag{3.5}$$

If $\alpha < 1$, we must have that

$$f(x + \alpha d) \leq f(x) + \gamma\alpha\nabla f(x)^\top d, \tag{3.6}$$

$$f\left(x + \frac{\alpha}{\delta}d\right) > f(x) + \gamma\frac{\alpha}{\delta}\nabla f(x)^\top d. \tag{3.7}$$

Applying the mean value theorem to equation (3.7), we get

$$\nabla f\left(x + \theta\frac{\alpha}{\delta}d\right)^\top d > \gamma\nabla f(x)^\top d,$$

where $\theta \in (0, 1)$. Adding and subtracting $\nabla f(x)^\top d$, and rearranging, we get

$$(1-\gamma)\nabla f(x)^\top d > \left[\nabla f(x) - \nabla f\left(x + \theta\frac{\alpha}{\delta}d\right)\right]^\top d.$$

By the Lipschitz-continuity of $\nabla f(x)$, we can write

$$\left[\nabla f(x) - \nabla f\left(x + \theta\frac{\alpha}{\delta}d\right)\right]^\top d \geq -L\frac{\alpha}{\delta}\|d\|^2,$$

which means that

$$(1-\gamma)\nabla f(x)^\top d > -L\frac{\alpha}{\delta}\|d\|^2,$$

Rearranging, we get

$$\frac{\delta}{L}(1-\gamma)\nabla f(x)^\top d > -\alpha\|d\|^2.$$

This last inequality, together with (3.4), yields

$$\frac{\delta}{L}(1-\gamma)\nabla f(x)^\top d > \alpha\nabla f(x)^\top d,$$

and substituting in equation (3.6) we finally get

$$f(\tilde{x}) < f(x) + \gamma\frac{\delta}{L}(1-\gamma)\nabla f(x)^\top d \leq f(x) - \gamma\frac{\delta}{L}(1-\gamma)\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|^2.$$

This last inequality, together with (3.5), implies that

$$f(\tilde{x}) \leq f(x) - \sigma\left(\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|\right)$$

where

$$\sigma(t) = \gamma\min\left\{1, \frac{\delta}{L}(1-\gamma)\right\}t^2.$$

∎

We can now state a couple of preliminary theoretical results. We first show that Algorithm 3.2 is well-posed.

PROPOSITION 3.6. *For each iteration k, Step 2 of Algorithm 3.2 terminates in a finite number of steps.*

Proof. Suppose by contradiction that Steps 2.1-2.4 generate an infinite loop, so that an infinite sequence of points $\{x^j\}$ is produced for which

$$\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\| > \left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\| + \mu_k > 0 \quad \forall j. \tag{3.8}$$

By Proposition 3.5, for each j we have that

$$f(x^{j+1}) - f(x^j) \leq -\sigma\left(\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\|\right), \tag{3.9}$$

where $\sigma(\cdot) \geq 0$. The sequence $\{f(x^j)\}$ is therefore nonincreasing. Moreover, (3.9) implies that

$$\left|f(x^{j+1}) - f(x^j)\right| \geq \sigma\left(\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\|\right). \tag{3.10}$$

By Assumption 2, $\{f(x^j)\}$ is lower bounded. Therefore, recalling that $\{f(x^j)\}$ is nonincreasing, we get that $\{f(x^j)\}$ converges, which implies that

$$\left|f(x^{j+1}) - f(x^j)\right| \to 0.$$

By (3.10), we get that $\sigma\left(\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\|\right) \to 0$, and, by the properties of $\sigma(\cdot)$, we finally get that $\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\| \to 0$, and this contradicts (3.8). ∎

The next proposition shows some properties of the sequences generated by the algorithm, which will play an important role in the subsequent analysis.

PROPOSITION 3.7. *Let $\{(x^k, y^k)\}$, $\{\mu_k\}$ and $\{\eta_k\}$ be the sequences produced by the algorithm. Then:*

(i) *the sequence $\{f(x^k)\}$ is nonincreasing and convergent;*
(ii) *the sequence $\{(x^k, y^k)\}$ is bounded;*
(iii) *the set $K_u = \{k \mid \eta_k < \eta_{k-1}\}$ of unsuccessful iterates is infinite;*
(iv) $\lim_{k\to\infty} \mu_k = 0$;
(v) $\lim_{k\to\infty} \eta_k = 0$;
(vi) $\lim_{k\to\infty} \left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\| = 0$.

Proof.
(i) The instructions of the algorithm and Proposition 3.5 imply that $\{f(x^k)\}$ is nonincreasing, and Assumption 2 implies that $\{f(x^k)\}$ is lower bounded. Hence, $\{f(x^k)\}$ converges.
(ii) The instructions of the algorithm imply that each point (x^k, y^k) belongs to the level set $\mathcal{L}(x^0, y^0)$, which is compact by Assumption 2. Therefore, $\{(x^k, y^k)\}$ is bounded.
(iii) Suppose that K_u is finite. Then there exists $\bar{k} > 0$ such that all iterates satisfying $k > \bar{k}$ are successful, i.e.,

$$f(x^k) \leq f(x^{k-1}) - \eta_{k-1},$$

and $\eta_k = \eta_{k-1} = \eta > 0$ for all $k \geq \bar{k}$. Since $\eta > 0$, this implies that $\{f(x^k)\}$ diverges to $-\infty$, in contradiction with Item (i).
(iv) Since, for all k, $\mu_{k+1} = \delta\mu_k$, where $\delta \in (0, 1)$, the claim holds.
(v) If $k \in K_u$, then $\eta_{k+1} = \theta\eta_k$, where $\theta \in (0, 1)$. Since K_u is infinite and $\eta_{k+1} = \eta_k$ if $k \notin K_u$, the claim holds.
(vi) By Proposition 3.5, we have that

$$f(\tilde{x}^k) - f(x^k) \leq -\sigma\left(\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\|\right).$$

By the instructions of the algorithm, $f(x^{k+1}) \leq f(\tilde{x}^k)$, and so we can write

$$f(x^{k+1}) - f(x^k) \leq -\sigma\left(\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\|\right),$$

i.e.,

$$\left|f(x^{k+1}) - f(x^k)\right| \geq \sigma\left(\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\|\right).$$

Since $\{f(x^k)\}$ converges, we get that $\sigma\left(\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\|\right) \to 0$. By the properties of $\sigma(\cdot)$, we get that $\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\| \to 0$. ∎

Before stating the main theorem of this section, it is useful to summarize some theoretical properties of the subsequence $\{(x^k, y^k)\}_{K_u}$ of the unsuccessful iterates. As the proof shows, the next proposition follows easily from the theoretical results we have shown above.

PROPOSITION 3.8. *Let $\{(x^k, y^k)\}$ be the sequence of iterates generated by Algorithm 3.2, and let $K_u = \{k \mid \eta_k < \eta_{k-1}\}$. Then:*

(i) *$\{(x^k, y^k)\}_{K_u}$ admits accumulation points;*
(ii) *for any accumulation point (x^*, y^*) of the sequence $\{(x^k, y^k)\}_{K_u}$, every $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$ is an accumulation point of a sequence $\{(\hat{x}^k, \hat{y}^k)\}_{K_u}$ where $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$.*

Proof. (i) By Proposition 3.7, Item (ii), $\{(x^k, y^k)\}$ is bounded. Therefore, $\{(x^k, y^k)\}_{K_u}$ is also bounded, and so it admits accumulation points.
(ii) Proposition 3.3 implies that every $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$ is an accumulation point of a sequence $\{(\hat{x}^k, \hat{y}^k)\}_{K_u}$, where $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$. ∎

We can now prove the main theoretical result of this section.

THEOREM 3.9. *Let $\{(x^k, y^k)\}$ be the sequence generated by Algorithm 3.2. Every accumulation point (x^*, y^*) of $\{(x^k, y^k)\}_{K_u}$ is such that x^* is an $\mathcal{N}$-stationary point of problem (1.1).*

Proof. Let (x^*, y^*) be an accumulation point of $\{(x^k, y^k)\}_{K_u}$. We must show that conditions (i)-(iii) of Definition 2.4 are satisfied.

(i) From the instructions of Algorithm 3.2 the iterates (x^k, y^k) belong to the set $\mathcal{L}(x^0, y^0)$, which is closed from Assumption 2. Any limit point (x^*, y^*) belongs to $\mathcal{L}(x^0, y^0)$ and is thus feasible for problem (2.1).
(ii) The result follows from Proposition 3.7, Item (vi).
(iii) Since K_u is an infinite subset of unsuccessful iterations, recalling that $x^k = \tilde{x}^{k-1}$, $y^k = y^{k-1}$, and setting $\hat{x}^k = \hat{x}^{k-1}$, $\hat{y}^k = \hat{y}^{k-1}$ for all $(\hat{x}^{k-1}, \hat{y}^{k-1}) \in \mathcal{N}(\tilde{x}^{k-1}, y^{k-1})$, the test at Step 3 fails at iteration k, and therefore

$$f(\hat{x}^k) > f(x^k) - \eta_{k-1}$$

for all $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$. Since the sequence $\{f(x^k)\}$ is nonincreasing (Proposition 3.7, Item (i)), we can write

$$f(x^*) \leq f(x^k) < f(\hat{x}^k) + \eta_{k-1}.$$

for all $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$. Taking limits, we get from Proposition 3.7, Item (v), Proposition 3.2, and by the continuity of f that $f(x^*) \leq f(\hat{x})$ for all $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$.

Now, note that Item (i) of Proposition 3.7 ensures the existence of $f^* \in \mathbb{R}$ satisfying

$$\lim_{k\to\infty} f(x^k) = f(x^*) = f^*. \tag{3.11}$$

Consider any $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$ such that

$$f(\hat{x}) = f^*. \tag{3.12}$$

Proposition 3.8 implies that $(\hat{x}, \hat{y})$ is an accumulation point of a sequence $\{(\hat{x}^k, \hat{y}^k)\}_{K_u}$, where $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$. Since $k \in K_u$, we have that $x^k = \tilde{x}^{k-1}$, $y^k = y^{k-1}$. Setting $\hat{x}^k = \hat{x}^{k-1}$, $\hat{y}^k = \hat{y}^{k-1}$ for all $(\hat{x}^{k-1}, \hat{y}^{k-1}) \in \mathcal{N}(\tilde{x}^{k-1}, y^{k-1})$, by (3.11) and (3.12) we get, for k sufficiently large,

$$f(\hat{x}^k) < f(x^k) + \xi.$$

Therefore, for such values of k, $(\hat{x}^k, \hat{y}^k) \in W_k$, and Steps 3.2-3.4 produce the points $x_k^2, \ldots, x_k^{j_k^*}$ (where j_k^* is the finite number of iterations of Steps 2.2-2.4 until the test at Step 2.4 fails), which, by the instructions at Step 2.2 and by Proposition 3.5, satisfy

$$f(\hat{x}^k) \geq f(x_k^2) \geq \ldots \geq f(x_k^{j_k^*}). \tag{3.13}$$

Since $k \in K_u$, Step 2.3 fails, and we can write

$$f(x_k^{j_k^*}) > f(\tilde{x}^k) - \eta_k \geq f(x^k) - \eta_{k-1}. \tag{3.14}$$

Moreover, as the sequence $\{(\hat{x}^k, \hat{y}^k)\}_{K_u}$ converges to the point $(\hat{x}, \hat{y})$, by (3.11), (3.12), (3.13), (3.14), and by Item (v) of Proposition 3.7, we obtain

$$f^* = \lim_{k\to\infty, k\in K_u} f(\hat{x}^k) = \lim_{k\to\infty, k\in K_u} f(x_k^2) = \lim_{k\to\infty, k\in K_u} f(x^k) = f^*.$$

By Proposition 3.5, we have that

$$f(x_k^2) \leq f(\hat{x}^k) - \sigma\left(\left\|\hat{x}^k - \Pi_{\mathcal{X}(\hat{y}^k)}\left[\hat{x}^k - \nabla f(\hat{x}^k)\right]\right\|\right),$$

which can be rewritten as

$$\left|f(x_k^2) - f(\hat{x}^k)\right| \geq \sigma\left(\left\|\hat{x}^k - \Pi_{\mathcal{X}(\hat{y}^k)}\left[\hat{x}^k - \nabla f(\hat{x}^k)\right]\right\|\right).$$

Taking limits for $k \to \infty, k \in K_u$, we finally get

$$\left\|\hat{x} - \Pi_{\mathcal{X}(\hat{y})}\left[\hat{x} - \nabla f(\hat{x})\right]\right\| = 0,$$

and the claim holds. ∎

In [4], the concept of *basic feasibility* (BF) introduced in [3] is extended to problem (1.1):

DEFINITION 3.10. *A feasible point x^* of problem (1.1) is referred to as basic feasible if, for any super support set J, letting $y_J \in \{0,1\}^n$ such that $y_i = 0$ if $i \in J$ and $y_i = 1$ otherwise, there exists $L > 0$ such that*

$$x^* = \Pi_{\mathcal{X}(y_J)}\left(x^* + d\right),$$

where $d_i = -\frac{1}{L}\nabla_i f(x^*)$ if $i \in J$ and $d_i = 0$ otherwise.

Note that BF stationarity requires that, for any y_J defining a super support set, $x^* = \Pi_{\mathcal{X}(y_J)}[x^* + d]$, where $d_J = -\frac{1}{L}\nabla_J f(x^*)$ and $d_{\bar{J}} = 0$, whereas the condition in Definition 2.6 requires $x^* = \Pi_{\mathcal{X}(y_J)}[x^* - \nabla f(x^*)]$. In fact, in the case of our problem the two conditions are equivalent, as we show below.

LEMMA 3.11. *Let $y \in \mathcal{Y}$ and $x^* \in \mathcal{X}(y)$. Then x^* satisfies*

$$x^* = \Pi_{\mathcal{X}(y)}(x^* + d),$$

where $d_{I_0(y)} = -\frac{1}{L}\nabla_J f(x^*)$ and $d_{I_1(y)} = 0$, if and only if it satisfies

$$x^* = \Pi_{\mathcal{X}(y)}(x^* - \nabla f(x^*)).$$

Proof. Let us consider

$$\hat{x} = \Pi_{\mathcal{X}(y)}[x^* - \nabla f(x^*)], \qquad \tilde{x} = \Pi_{\mathcal{X}(y)}[x^* + d].$$

Let us denote $\hat{x}^p = x^* - \nabla f(x^k)$ and $\tilde{x}^p = x^* + d$ Since both $\hat{x}$ and $\tilde{x}$ belong to $\mathcal{X}(y)$, we have $\hat{x}_{I_1(y)} = 0$ and $\tilde{x}_{I_1(y)} = 0$. From the well-known properties of the projection operator on a convex set, we get:

$$(\hat{x} - \hat{x}^p)^\top(\hat{x} - x) \leq 0 \quad \forall\, x \in X(y),$$

$$(\tilde{x} - \tilde{x}^p)^\top(\tilde{x} - x) \leq 0 \quad \forall\, x \in X(y),$$

hence

$$(\hat{x} - \hat{x}^p)^\top(\hat{x} - \tilde{x}) \leq 0, \qquad (\tilde{x} - \tilde{x}^p)^\top(\tilde{x} - \hat{x}) \leq 0.$$

Taking into account that $\tilde{x}_{I_1(y)} = \hat{x}_{I_1(y)} = 0$ and $\hat{x}^p_{I_0(y)} = \tilde{x}^p_{I_0(y)} = x^* - \nabla_{I_0(y)} f(x^*)$, we get

$$(\hat{x}_{I_0(y)} - \hat{x}^p_{I_0(y)})^\top(\hat{x}_{I_0(y)} - \tilde{x}_{I_0(y)}) \leq 0, \qquad (\tilde{x}_{I_0(y)} - \hat{x}^p_{I_0(y)})^\top(\tilde{x}_{I_0(y)} - \hat{x}_{I_0(y)}) \leq 0,$$

i.e.,

$$\|\hat{x}_{I_0(y)}\|^2 - \tilde{x}^\top_{I_0(y)}\hat{x}_{I_0(y)} - \hat{x}^\top_{I_0(y)}\hat{x}^p_{I_0(y)} + \tilde{x}^\top_{I_0(y)}\hat{x}^p_{I_0(y)} \leq 0,$$

and

$$\|\tilde{x}_{I_0(y)}\|^2 - \tilde{x}^\top_{I_0(y)}\hat{x}_{I_0(y)} - \tilde{x}^\top_{I_0(y)}\hat{x}^p_{I_0(y)} + \hat{x}^\top_{I_0(y)}\hat{x}^p_J \leq 0$$

Summing up the two inequalities, we get

$$\|\hat{x}_{I_0(y)}\|^2 + \|\tilde{x}_{I_0(y)}\|^2 - 2\hat{x}^\top_{I_0(y)}\tilde{x}_{I_0(y)} \leq 0,$$

i.e.,

$$\|\hat{x}_{I_0(y)} - \tilde{x}_{I_0(y)}\| \leq 0,$$

from which we obtain $\tilde{x}_{I_0(y)} = \hat{x}_{I_0(y)}$ and hence $\hat{x} = \tilde{x}$. ∎

We can hence show that, provided that $\mathcal{N}_\rho$ is employed as neighborhood in 3.2, with a sufficiently large value of ρ, the SNS procedure converges to basic feasible solutions.

THEOREM 3.12. *Let $\{(x^k, y^k)\}$ be the sequence of iterates generated by Algorithm 3.2 equipped with $\mathcal{N}_\rho$ as neighborhood and $\mathcal{A}^*$ the set of the accumulation points of the sequence $\{(x^k, y^k)\}_{K_u}$ of unsuccessful iterates. If $\rho \geq 2(s - \delta^*)$, in the definition of the set $\mathcal{N}_\rho(x, y)$, and $\delta^* = \min\{\|x^*\|_0 \mid (x^*, y^*) \in \mathcal{A}^*\}$, then given a point $(x^*, y^*) \in \mathcal{A}^*$, x^* is basic feasible for problem (1.1).*

Proof. Let $J \subset \{1,\dots,n\}$ be any super support set for x^*, and consider the vector $\hat{y}$ such that $\hat{y}_j = 1 \quad \forall j \notin J$ and zero otherwise. As $|J| = s$, we have $e^\top \hat{y} = n - s$, and, taking into account that $i \notin J$ implies $x_i^* = 0$ and $i \in J$ implies $\hat{y}_i = 0$, it follows

$$x_i^* \hat{y}_i = 0 \quad i = 1,\dots,n.$$

Then, we have $I_1(x^*) \subseteq I_0(\hat{y})$ and $(x^*, \hat{y}) \in \bar{\mathcal{N}}(x^*) \subseteq \mathcal{N}_\rho(x^*, y^*)$, where we used Lemma 3.4. By taking into account Theorem 3.9, we finally get that x^* is an $\mathcal{N}_\rho$-stationary point of problem (1.1) and that it is also a stationary point of

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } x \in \mathcal{X}(\hat{y}), \end{aligned}$$

that is

$$x^* = \Pi_{\mathcal{X}(\hat{y})}(x^* - \nabla f(x^*)).$$

Then, by Lemma 3.11, recalling that $\hat{y}_i = 0$ if and only if $i \in J$, we obtain that x^* is basic feasible. ∎

4. Convergence results under constraint qualifications. In this section, we show that, under constraint qualifications and by choosing suitable neighborhoods, it is possible to state convergence results similar to those considered in important works of the related literature [11, 23]. Here, we assume that $X = \{x \in \mathbb{R}^n \mid g(x) \leq 0,\ h(x) = 0\}$, where h_i, $i = 1,\dots,p$ are affine functions and g_i, $i = 1,\dots,m$, are convex functions. First we state the following assumption which implicitly involves constraint qualifications.

ASSUMPTION 4. *Given $\bar{y} \in \mathcal{Y}$ and $\bar{x} \in \mathcal{X}(\bar{y})$, we have that $\bar{x}$ is a stationary point of problem (2.3) if and only if there exist multipliers $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^n$ such that*

$$\begin{gathered} \nabla f(\bar{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\bar{x}) + \sum_{i=1}^{p} \mu_i \nabla h_i(\bar{x}) + \sum_{i=1}^{n} \gamma_i e_i = 0, \\ \lambda_i \geq 0,\ \lambda_i g_i(\bar{x}) = 0,\ \forall i = 1,\dots,m, \\ \gamma_i = 0,\ \forall\, i \text{ such that } \bar{y}_i = 0. \end{gathered}$$

The above assumption states that $\bar{x}$ is a stationary point of problem (2.3) if and only if it is a KKT point of the following problem

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } h_i(x) = 0, \quad \forall i = 1,\dots,p, \\ &\quad\ g_i(x) \leq 0, \quad \forall i = 1,\dots,m, \\ &\quad\ x_i \bar{y}_i = 0, \quad \forall i = 1,\dots,n, \end{aligned}$$

which can be equivalenty rewritten as follows

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } h_i(x) = 0, \quad \forall i = 1,\dots,p, \\ &\quad\ g_i(x) \leq 0, \quad \forall i = 1,\dots,m, \\ &\quad\ x_i = 0, \quad\ \forall i : \bar{y}_i = 1. \end{aligned}$$

Remark 4.1. As shown in Appendix A, Assumption 4 holds when, e.g., the functions g_i are strongly convex with constant $\mu_i > 0$, for $i = 1, \dots, m$, the functions h_j, for $j = 1, \dots, p$ are affine, and some Cardinality Constraint-Constraint Qualification (CC-CQ) is satisfied. For instance, a standard CC-CQ is the Cardinality Constraint-Linear Independence Constraint Qualification (CC-LICQ), requiring that the gradients

$$\begin{aligned} &\nabla g_i(\bar{x}) && \text{for all } i : g_i(\bar{x}) = 0 \\ &\nabla h_i(\bar{x}) && \text{for all } i = 1, \dots, p \\ &e_i && \text{for all } i : \bar{y}_i = 1 \end{aligned}$$

are linearly independent.

From Theorem 3.9 and Assumption 4 we immediately get the following result.

THEOREM 4.2. *Let $\{(x^k, y^k)\}$ be the sequence generated by Algorithm 3.2. Every accumulation point (x^*, y^*) of the sequence of unsuccessful iterates $\{(x^k, y^k)\}_{K_u}$ is such that there exist multipliers $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^n$ such that*

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i \nabla h_i(x^*) + \sum_{i=1}^{n} \gamma_i e_i = 0, \\ \lambda_i \geq 0,\ \lambda_i g_i(x^*) = 0,\ \forall i = 1, \dots, m, \\ \gamma_i = 0,\ \forall\, i \text{ such that } y_i^* = 0. \end{aligned} \tag{4.1}$$

Remark 4.3. Condition (4.1) is the *S-stationarity* concept introduced in [11]. Basically, the limit points of the sequence $\{(x^k, y^k)\}_{K_u}$ produced by Algorithm 3.2 are always guaranteed to be S-stationary. This implies, by the results in [11], that x^* is also Mordukhovich-stationary for problem (1.1). In fact, under Assumption 4, it is easy to see that $\mathcal{N}$-stationarity is a stronger condition than M-stationarity, from points (i)-(ii) of Definition 2.4.

In order to state stronger convergence results, we need to use suitable neighborhoods (e.g., $\mathcal{N}_\rho$ with a sufficiently large value of ρ) in the algorithm.

THEOREM 4.4. *Let $\{(x^k, y^k)\}$ be the sequence generated by Algorithm 3.2 equipped with $\mathcal{N}_\rho$ as neighborhood and $\mathcal{A}^*$ the set of the accumulation points of the sequence $\{(x^k, y^k)\}_{K_u}$ of unsuccessful iterates. If $\rho \geq 2(s - \delta^*)$, in the definition of the set $\mathcal{N}_\rho(x, y)$, and $\delta^* = \min\{\|x^*\|_0 \mid (x^*, y^*) \in \mathcal{A}^*\}$, then given a point $(x^*, y^*) \in \mathcal{A}^*$ and for every super support set $J \subset \{1, \dots, n\}$, we have that there exist multipliers $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^n$ such that*

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i \nabla h_i(x^*) + \sum_{i=1}^{n} \gamma_i e_i = 0, \\ \lambda_i \geq 0,\ \lambda_i g_i(x^*) = 0,\ \forall i = 1, \dots, m, \\ \gamma_i = 0,\ \forall\, i \in J. \end{aligned} \tag{4.2}$$

Proof. Let $J \subset \{1, \dots, n\}$ be any super support set for x^*, and consider the vector $\hat{y}$ such that $\hat{y}_j = 1 \quad \forall j \notin J$ and zero otherwise. As $|J| = s$, we have $e^\top \hat{y} = n - s$, and, taking into account that $i \notin J$ implies $x_i^* = 0$ and $i \in J$ implies $\hat{y}_i = 0$, it follows

$$x_i^* \hat{y}_i = 0 \quad i = 1, \dots, n.$$

Then, we have $I_1(x^*) \subseteq I_0(\hat{y})$ and $(x^*, \hat{y}) \in \bar{\mathcal{N}}(x^*) \subseteq \mathcal{N}_\rho(x^*, y^*)$, where we used Lemma 3.4. By taking into account Theorem 3.9, we finally get that x^* is an $\mathcal{N}_\rho$-stationary point of problem (1.1) and that it is also a stationary point of

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } x \in \mathcal{X}(\hat{y}). \end{aligned}$$

Then, by Assumption 4, recalling that $\hat{y}_i = 0$ if and only if $i \in J$, we obtain that (4.2) holds. ∎

Remark 4.5. Condition (4.2) is the necessary optimality condition first defined in [23]. It is interesting to note that the Penalty Decomposition algorithm proposed in the referenced work in fact is not guaranteed to converge to a point satisfying such conditions, that are guaranteed to hold only if the limit point has full support. In the general case, the PD method generates points satisfying (4.2) for at least one super support set. Our SNS algorithm would have the same exact convergence results if we used the neighborhood

$$\mathcal{N}(x^k, y^k) = \{(x, y) \mid x = x^k,\ e^\top y = n - s,\ y_i x_i^k = 0 \,\forall\, i\}.$$

The above neighborhood basically checks all the super support sets at the current iterate x^k, but it does not satisfy the continuity Assumption 3, hence failing to guarantee that condition (4.2) is satisfied by all super support sets at the limit point.

5. Numerical Experiments. From a computational point of view, we are particularly interested in studying two relevant aspects. Specifically, here we want to:

- analyze the benefits and the costs of increasing the size of the neighborhood;
- assess the performance of the proposed approach, compared to the the Greedy Sparse-Simplex (GSS) method proposed in [3] and the Penalty Decomposition (PD) approach [23].

To these aims, we considered the problem of sparse logistic regression, where the objective function is continuously differentiable and convex, but the solution of the problem for a fixed support set requires the adoption of an iterative method. Note that we preferred to consider a problem without other constraints in addition to the sparsity one, in order to simplify the analysis of the behavior of the proposed algorithm.

The problem of *sparse logistic regression* [19] has important applications, for instance, in machine learning [2, 30]. Given a dataset having N samples $\{z^1, \dots, z^N\}$, with n features and N corresponding labels $\{t_1, \dots, t_N\}$ belonging to $\{-1, 1\}$, the problem of sparse maximum likelihood estimation of a logistic regression model can be formulated as follows

$$\begin{aligned} &\min_w L(w) = \sum_{i=1}^{N} \log\left(1 + \exp\left(-t_i (w^\top z^i)\right)\right) \\ &\text{s.t. } \|w\|_0 \le s. \end{aligned} \tag{5.1}$$

The benchmark for this experiment is made up of problems of the form (5.1), obtained as described hereafter. We employed 6 binary classification datasets, listed in Table 1. All the datasets are from the UCI Machine Learning Repository [17]. 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 different values of s, as specified later in this section.

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`

5.1. Implementation details. Algorithms SNS, PD and GSS have been implemented in Python 3.7, mainly exploiting libraries `numpy` and `scipy`. The convex subproblems of both PD and GSS have been solved up to global optimality by using the L-BFGS algorithm (in the implementation from [22], provided by `scipy`). We also employed L-BFGS for the local optimization steps in SNS. All algorithms start from the feasible initial point $x^0 = 0 \in \mathbb{R}^n$. For the PD algorithm, we set the starting penalty parameter to 1 and its growth rate to 1.05. The algorithm stops when $\|x^k - y^k\| < 0.0001$, as suggested in [23]. AS for the GSS, we stop the algorithm as soon as $\|x^{k+1} - x^k\| \leq 0.0001$.

Concerning our proposed Algorithm 3.2, the parameters have been set as follows:

- $\xi = 10^3$,
- $\theta = 0.5$,
- $\eta_0 = 10^{-5}$.

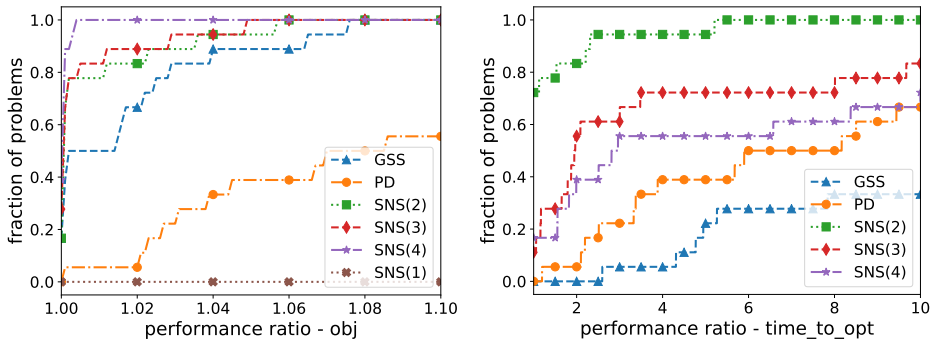
For what concerns μ_0 and δ, we actually keep the value of μ fixed to 10^{-6}. We again employ the stopping criterion $\|x^{k+1} - x^k\| \leq 0.0001$.

For all the algorithms, we have also set a time limit of 10^4 seconds. All the experiments have been carried out on an Intel(R) Xeon E5-2430 v2 @2.50GHz CPU machine with 6 physical cores (12 threads) and 16 GB RAM.

As benchmark for our experiments, we considered 18 problems, obtained from the 6 datasets in Table 1 and setting s to 3, 5 and 8 in (5.1). For SNS and GSS we consider the computational time employed to find the best solution. We take into account four versions of Algorithm 3.2, with neighborhood radius $\rho \in \{1, 2, 3, 4\}$.

In Figure 1 the performance profiles [16] w.r.t. the objective function values and the runtimes (intended as the time to find the best solution) attained by the different algorithms are shown. We do not report the runtime profile of SNS(1) since it is much faster than all the other methods and thus would dominate the plot, making it poorly informative. We can however note that unfortunately its speed is outweighed by the very poor quality of the solutions. We can observe that increasing the size of the neighborhood consistently leads to higher quality solutions, even though the computational cost grows. We can see that SNS (with a sufficiently large neighborhood) has better performances than the other algorithms known from the literature; in particular, while the neighborhood radius $\rho = 1$ only allows to perform forward selection, with poor outcomes, $\rho \geq 2$ makes swap operations possible, with a significant impact on the exploration capabilities. The GSS has worse quality performance than SNS(2), which is reasonable, since its move set is actually smaller and optimization is always carried out w.r.t. a single variable and not the entire active set. However, it proved to also be slower than the SNS, mostly because of two reasons: it always tries all feasible moves, not necessarily accepting the first one that provides an objective decrease, and

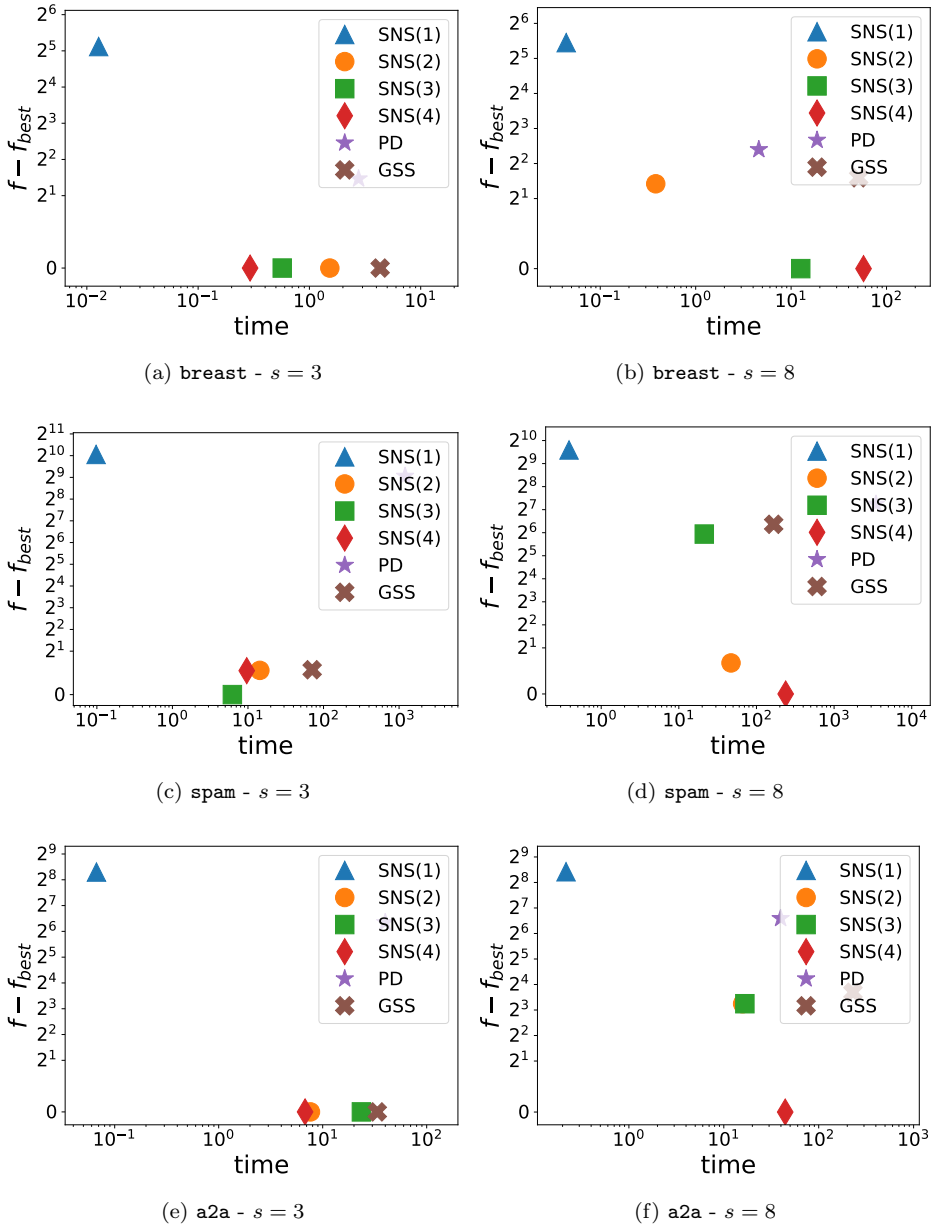
it requires many more iterations to converge, since it considers one variable at a time. Finally, the PD method appears not to be competitive from both points of view: it is slow at converging to a feasible point and it has substantially no global optimization features that could guide to globally good solutions.



(a) objective value (b) time

FIG. 1. *Performance profiles for the considered algorithms on 18 sparse logistic regression problems.*

It is interesting to remark how considering larger neighborhoods appears to be particularly useful in problems where the sparsity constraint is less strict and thus combinatorially more challenging. As an example, we show the runtime-objective tradeoff for the `breast`, `spam` and `a2a` problems for $s = 3$ and $s = 8$ in Figure 2. We can observe that for $s = 3$, SNS finds good, similar solutions for either $\rho = 2, 3$ or 4, with a similar computational cost. On the other hand, as s grows to 8, using $\rho = 4$ allows to significantly improve the quality of the solution without a significant increase in terms of runtime.



(a) `breast` - $s = 3$ (b) `breast` - $s = 8$

(c) `spam` - $s = 3$ (d) `spam` - $s = 8$

(e) `a2a` - $s = 3$ (f) `a2a` - $s = 8$

FIG. 2. *Quality/cost trade-off for the algorithms on sparse logistic regression problems from datasets* `breast`, `spam` *and* `a2a`.

6. Conclusions. In this paper we have analyzed sparsity constrained optimization problems. For this class of problems, we have defined a necessary optimality condition, namely, $\mathcal{N}$-stationarity, exploiting the concept of discrete neighborhood associated with a well-known mixed integer equivalent reformulation, that allows to take into account potentially advantageous changes on the set of active variables.

We have afterwards proposed an algorithmic framework to tackle the family of

problems under analysis. Our SNS method alternates continuous local search steps and neighborhood exploration steps; the algorithm is then proved to produce a sequence of iterates whose cluster points are $\mathcal{N}$-stationary. Moreover, we proved that, by suitably employing a tailored neighborhood, the limit points also satisfy other optimality conditions from the literature, based on both gradient projection and Lagrange multipliers, thus providing stronger optimality guarantees than other state-of-the-art approaches.

Finally, we studied the features and the benefits of our proposed procedure from a computational perspective. Specifically, we compared the performance of the SNS as the size of the neighborhood increases, observing that using wider neighborhoods consistently provides higher quality solutions with a reasonable increase of the computational cost, especially when the required cardinality is not that small. Moreover, when comparing SNS with the Penalty Decomposition method and the Greedy Sparse-Simplex method, we observed that our method has higher exploration capability, thus getting a nice match between theory and practice, and it is affordable in terms of computational cost, being even faster than the other considered methods.

Appendix A. On the relationship between stationarity conditions and KKT conditions. Consider the continuous optimization problem

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } x \in X, \end{aligned} \tag{A.1}$$

where $X = \{x \in \mathbb{R}^n \mid h(x) = 0,\ g(x) \leq 0\}$ is a convex set ($h_i,\ i = 1, \dots, p$ are affine functions, $g_i,\ i = 1, \dots, m$, are convex functions). We assume f and g to be continuously differentiable; h is differentiable, being affine.

DEFINITION A.1. *A point $x^* \in X$ is a stationary point for problem (A.1) if, for any direction d feasible at x^*, we have*

$$\nabla f(x^*)^\top d \geq 0.$$

It can be shown that a point x^* is stationary for problem (A.1) if and only if

$$x^* = \Pi_X[x^* - \nabla f(x^*)], \tag{A.2}$$

where Π_X denotes the orthogonal projection operator. Stationarity is a necessary condition of optimality for problem (A.1). It is possible to show that a point satisfying the KKT conditions is always a stationary point. Viceversa is true by stronger assumptions on the set of feasible directions.

PROPOSITION A.2. *Let $x^* \in X$ satisfy KKT conditions for problem (A.1). Then, x^* is stationary for problem (A.1).*

Proof. Assume x^* satisfies KKT conditions with multipliers λ and μ. Let d be any feasible direction at x^*. Since X is convex, we know that:

$$\nabla h_i(x^*)^\top d = 0 \quad \forall i = 1, \dots, p, \tag{A.3}$$

$$\nabla g_i(x^*)^\top d \leq 0 \quad \forall i : g_i(x^*) = 0. \tag{A.4}$$

Moreover, from KKT conditions we know that

$$\lambda_i = 0 \quad \forall\, i : g_i(x^*) < 0. \tag{A.5}$$

We know that

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{m} \mu_i \nabla h_i(x^*) = 0,$$

hence

$$\left(\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i \nabla h_i(x^*) = 0 \right)^\top d = 0,$$

and then

$$\nabla f(x^*)^\top d + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*)^\top d + \sum_{i=1}^{m} \mu_i \nabla h_i(x^*)^\top d = 0.$$

From equations (A.3) and (A.5), we get

$$\nabla f(x^*)^\top d + \sum_{i:g_i(x^*)=0} \lambda_i \nabla g_i(x^*)^\top d = 0,$$

thus, recalling (A.4) and $\lambda \geq 0$,

$$\nabla f(x^*)^\top d = - \sum_{i:g_i(x^*)=0} \lambda_i \nabla g_i(x^*)^\top d \geq 0.$$

Since d is an arbitrary feasible direction, we get the thesis. ∎

PROPOSITION A.3. *Let $x^* \in X$ be a stationary point for problem (A.1). Assume that one of the following conditions holds:*

(i) the set of feasible direction $D(x^*)$ is such that

$$D(x^*) = \{d \in \mathbb{R}^n : \nabla g_i(x^*)^\top d \leq 0,\ i \in I(x^*), \nabla h_i(x^*)^\top d = 0, i = 1, \ldots, p\}$$

(ii) the set of feasible direction $D(x^*)$ is such that

$$D(x^*) = \{d \in \mathbb{R}^n \mid \nabla g_i(x^*)^\top d < 0,\ i : g_i(x^*) = 0,\ \nabla h_j(x^*)^\top d = 0, j = 1, \ldots, p\},$$

and a constraint qualification holds.

Then, x^* is a KKT point.

Proof. Assertion (i). Let x^* be a stationary point. Then, there does not exist a direction $d \in D(x^*)$ such that

$$\nabla f(x^*)^\top d < 0.$$

This implies that the system

$$\begin{array}{lll} \nabla f(x^*)^\top d & < 0 & \\ \nabla g_i(x^*)^\top d & \leq 0 & i : g_i(x^*) = 0 \\ \nabla h_i(x^*)^\top d & \leq 0 & i = 1, \ldots, p \\ -\nabla h_i(x^*)^\top d & \leq 0 & i = 1, \ldots, p \end{array}$$

does not admit solution. By Farkas' Lemma we get the thesis.

Assertion (ii). Let x^* be a stationary point. Then, there does not exist a direction $d \in D(x^*)$ such that

$$\nabla f(x^*)^\top d < 0.$$

This implies that the system

$$\begin{array}{lll} \nabla f(x^*)^\top d & < 0 & \\ \nabla g_i(x^*)^\top d & < 0 & i : g_i(x^*) = 0 \\ \nabla h_i(x^*)^\top d & = 0 & i = 1, \dots, p \end{array}$$

does not admit solution. By Motzkin's theorem we get that x^* satisfies the Fritz-John conditions and hence, by assuming a constraint qualification, the thesis is proved. ∎

Condition (i) of Proposition A.3 holds if the functions g_i, $i = 1, \dots, m$, h_j, $j = 1, \dots, p$ are affine.

Condition (ii) of Proposition A.3 holds by assuming that the convex functions g_i, for $i = 1, \dots, m$ are such that

$$g_i(x + td) \geq g_i(x) + t\nabla g_i(x)^\top d + \frac{1}{2}\gamma t^2 \|d\|^2 \tag{A.6}$$

with $\gamma > 0$. Indeed, in this case it is easy to see that a direction d is a feasible direction at x^* if and only if

$$\nabla g_i(x^*)^\top d < 0 \qquad i : g_i(x^*) = 0 \qquad \nabla h_j(x^*)^\top d = 0 \quad i = 1, \dots, p$$

Condition (A.6) is satisfied by assuming that the functions g_i are twice continuosly differentiable and the Hessian matrix is positive definite.

Condition (A.6) holds also for continuously differentiable functions g_i assuming that they are *strongly convex with constant* $\mu_i > 0$, i.e., that for $i = 1, \dots, m$ the functions

$$g_i(y) \geq g_i(x) + \nabla g_i(x)^\top (y - x) + \frac{\mu_i}{2}\|y - x\|^2, \quad \forall\, x, y.$$

REFERENCES

[1] K.P. Anagnostopoulos and G. Mamanis. A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research*, 37(7):1285–1297, 2010.

[2] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

[3] A. Beck and Y. Eldar. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization*, 23(3):1480–1509, 2013.

[4] Amir Beck and Nadav Hallak. On the minimization over sparse symmetric sets: projections, optimality conditions, and algorithms. *Mathematics of Operations Research*, 41(1):196–223, 2016.

[5] C. Berge. *Topological Spaces: Including a Treatment of Multi-valued Functions, Vector Spaces and Convexity*. Macmillan, 1963.

[6] Dimitris Bertsimas and Romy Shioda. Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1):1–22, 2009.

[7] Daniel Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74(2):121–140, 1996.

[8] Kris Boudt and Chunlin Wan. The effect of velocity sparsity on the performance of cardinality constrained particle swarm optimization. *Optimization Letters*, 2019.

[9] Martin Branda, Max Bucher, Michal Červinka, and Alexandra Schwartz. Convergence of a scholtes-type regularization method for cardinality-constrained optimization problems with an application in sparse robust portfolio optimization. *Computational Optimization and Applications*, 70(2):503–530, 2018.

[10] Max Bucher and Alexandra Schwartz. Second-order optimality conditions and improved convergence results for regularization methods for cardinality-constrained optimization problems. *Journal of Optimization Theory and Applications*, 178(2):383–410, 2018.

[11] O. Burdakov, C. Kanzow, and A. Schwartz. Mathematical programs with cardinality constraints: Reformulation by complementarity-type conditions and a regularization method. *SIAM Journal on Optimization*, 26(1):397–425, 2016.

[12] E.J. Candès and M.B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[13] Michal Červinka, Christian Kanzow, and Alexandra Schwartz. Constraint qualifications and optimality conditions for optimization problems with cardinality constraints. *Mathematical Programming*, 160(1):353–377, 2016.

[14] T.-J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, 2000.

[15] Guang-Feng Deng, Woo-Tsong Lin, and Chih-Chung Lo. Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications*, 39(4):4558–4566, 2012.

[16] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[17] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[18] Alberto Fernández and Sergio Gómez. Portfolio selection using neural networks. *Computers & Operations Research*, 34(4):1177–1191, 2007.

[19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[20] Matteo Lapucci, Tommaso Levato, and Marco Sciandrone. Convergent inexact penalty decomposition methods for cardinality-constrained problems. *Journal of Optimization Theory and Applications*, 188(2):473–496, 2021.

[21] Duan Li and Xiaoling Sun. *Nonlinear integer programming*, volume 84. Springer Science & Business Media, 2006.

[22] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[23] Z. Lu and Y. Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization*, 23(4):2448–2478, 2013.

[24] S. Lucidi, V. Piccialli, and M. Sciandrone. An algorithm model for mixed variable programming. *SIAM Journal on Optimization*, 15(4):1057–1084, 2005.

[25] A. Miller. *Subset Selection in Regression*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2002.

[26] Purity Mutunge and Dag Haugland. Minimizing the tracking error of cardinality constrained portfolios. *Computers & Operations Research*, 90:33–41, 2018.

[27] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[28] Dong X. Shaw, Shucheng Liu, and Leonid Kopman. Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. *Optimization Methods and Software*, 23(3):411–420, 2008.

[29] Juan Pablo Vielma, Shabbir Ahmed, and George L. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

[30] Jason Weston, André Elisseeff, Bernhard Schölkopf, and Mike Tipping. Use of the zero norm with linear models and kernel methods. *The Journal of Machine Learning Research*, 3:1439–1461, 2003.