

An Active-Set Algorithmic Framework for Non-Convex Optimization Problems over the Simplex

Andrea Cristofari*, Marianna De Santis†, Stefano Lucidi†, Francesco Rinaldi*

*Dipartimento di Matematica “Tullio Levi-Civita”
Università di Padova

Via Trieste, 63, 35121 Padua, Italy

E-mail: andrea.cristofari@unipd.it, rinaldi@math.unipd.it

†Dipartimento di Ingegneria Informatica, Automatica e Gestionale
Sapienza Università di Roma

Via Ariosto, 25, 00185 Rome, Italy

E-mail: mdesantis@diag.uniroma1.it, lucidi@diag.uniroma1.it

Abstract. In this paper, we describe a new active-set algorithmic framework for minimizing a non-convex function over the unit simplex. At each iteration, the method makes use of a rule for identifying active variables (i.e., variables that are zero at a stationary point) and specific directions (that we name *active-set gradient related* directions) satisfying a new “nonorthogonality” type of condition. We prove global convergence to stationary points when using an Armijo line search in the given framework. We further describe three different examples of active-set gradient related directions that guarantee linear convergence rate (under suitable assumptions). Finally, we report numerical experiments showing the effectiveness of the approach.

Keywords. Active-set methods. Unit simplex. Non-convex optimization. Large-scale optimization.

MSC2000 subject classifications. 65K05. 90C06. 90C30.

1 Introduction

Many real-world applications can be modeled as optimization problems over structured feasible sets. In particular, the problem of minimizing a function over a simple polytope (such as the unit simplex) arises in different fields like, e.g., machine learning, statistics and economics. Examples of relevant applications include training of support vector machines, boosting (Adaboost), convex approximation in ℓ_p , mixture density estimation, finding maximum stable sets (maximum cliques) in graphs, portfolio optimization and population dynamics problems (see, e.g., [4, 5, 8, 10] and references therein).

The problem we address here can be stated as follows:

$$\min_{x \in \Delta} f(x) \tag{1}$$

where $\Delta = \{x \in \mathbb{R}^n : e^T x = 1, x \geq 0\}$ is the unit simplex, $e \in \mathbb{R}^n$ is the vector of all ones, $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable and its gradient $\nabla f(x)$ is Lipschitz continuous over the feasible set, with constant $L > 0$.

Note that minimizing an objective function $h(x)$ over a polytope P can be recast as problem (1). Indeed, since any point $x \in P$ can be expressed as a convex combination of the columns of $V = [v_1 \ \dots \ v_m] \in \mathbb{R}^{n \times m}$, with v_1, \dots, v_m vertices of P , the problem $\min\{h(x) : x \in P\}$ can be rewritten as $\min\{h(Vy) : e^T y = 1, y \geq 0\}$. Thus, each variable y_i represents the weight of the i th vertex in the convex combination.

In many different contexts, problems can have very sparse solutions (i.e., solutions with many zero components). Hence, developing methods that allow to quickly identify the set of zero components in the optimal solution is getting crucial to guarantee relevant savings in terms of CPU time. In our problem, estimating the set of zero components in the optimal solution, or in a stationary point when the objective function is non-convex, coincides with estimating the set of *active* (or binding) inequality constraints. This set of active constraints is often referred to as *active set* and the so called *active-set methods* are characterized by computing, at each iteration, an estimate of the binding constraints which is iteratively updated. Usually, only a single active constraint is added to or deleted from the estimated active set at each iteration (see, e.g., [29] and references therein). However, when dealing with simple constraints, more sophisticated active-set methods can be used, which can add to or delete from the current estimated active set more than one constraint at each iteration, and eventually find the active set in a finite number of steps if certain conditions hold. In particular, several active-set algorithms are based on the idea of combining an “identification” step (to identify the active constraints) with a minimization step in a reduced space (see, e.g., [1, 3, 6, 9, 14, 16, 17, 20, 21, 22, 27] and references therein).

Here, we propose an active-set algorithmic framework for solving problem (1), where $f(x)$ is a possibly non-convex objective function. Hereinafter, we call *active variables* those variables that are equal to zero at a stationary point. Indeed, those variables correspond to the active constraints when the feasible set is the unit simplex.

In the first part of the paper, we describe an active-set estimate to identify the active variables by extending some specific strategies proposed in the contexts of box-constrained problems (see [7, 9, 11, 12]) to the case of unit simplex. The main features of our active-set strategy are essentially two:

1. it does not only focus on the zero variables and keep them fixed, but rather tries to quickly identify as many active variables as possible (including nonzero variables) at a given point;
2. it gives a significant reduction in the objective function, while guaranteeing feasibility, when setting to zero those variables estimated active (and moving a suitably chosen variable estimated nonactive).

These features enable us to easily use this active-set strategy into any globally convergent algorithm, potentially improving its efficiency without compromising its theoretical properties.

In the second part of the paper, inspired by the “nonorthogonality” type of condition described in [2], we define a new class of directions, named *active-set gradient related*, that

we combine with the active-set strategy described above to devise a two-step algorithmic framework. In the first step, our method sets to zero the estimated active variables and suitably moves an estimated nonactive variable, producing a new feasible point with a smaller objective value. In the second step, an active-set gradient related direction, combined with a suitable line search, is used in the subspace of the estimated nonactive variables to generate the next iterate. We prove convergence of our framework to stationary points of problem (1) using Armijo line search. We would like to highlight that, since at each iteration we move to zero some of the variables and then approximately optimize in a subspace, guaranteeing convergence is not a straightforward task and requires a thorough theoretical analysis.

We further give three specific examples of active-set gradient related directions that can be used to implement our algorithm in practice. More specifically, we consider Frank-Wolfe [18], away-step Frank-Wolfe [31] and projected gradient directions (see, e.g., [2] and references therein). We then prove linear convergence rate of the framework (under suitable assumptions) when using these directions.

In the final part of the paper, we report numerical results on both convex and non-convex instances. The results seem to indicate that our active-set algorithm is very efficient when dealing with sparse optimization problems.

The paper is organized as follows. In Section 2, we report the notation and useful preliminary results. In Section 3, we describe in depth our active-set estimate and the related theoretical properties. In Section 4, we present our algorithmic framework and carry out the convergence analysis. We also give three specific examples of active-set gradient related directions that can be used in the framework. In Section 5, we analyze the convergence rate of the method when using those active-set gradient related directions. In Section 6, we report our numerical experience. Finally, in Section 7, we draw some conclusions.

2 Notation and Preliminaries

Throughout the paper, we indicate with $\|\cdot\|$ the Euclidean norm. Given a vector $v \in \mathbb{R}^n$ and an index set $I \subseteq \{1, \dots, n\}$, we denote with v_I the subvector with components v_i , $i \in I$. We indicate with e_i the i th unit vector. Given $x \in \mathbb{R}^n$ and a non-empty closed convex set $S \subseteq \mathbb{R}^n$, we denote by $P(x)_S$ the projection of x on S . Finally, given a subset $I \subseteq \{1, \dots, n\}$, we denote Δ_I the subset of points of Δ with $x_i = 0$ for all $i \notin I$, i.e.,

$$\Delta_I := \{x \in \Delta : x_i = 0, \forall i \notin I\}.$$

Definition 1. *A stationary point x^* of problem (1) is a feasible point that satisfies the following first-order necessary optimality conditions:*

$$\nabla f(x^*) - \lambda^* e - \mu^* = 0, \tag{2}$$

$$(\mu^*)^T x^* = 0, \tag{3}$$

$$\mu^* \geq 0. \tag{4}$$

where $\lambda^* \in \mathbb{R}$ and $\mu^* \in \mathbb{R}^n$ are the KKT multipliers.

It is easy to verify that conditions (2)–(4) are equivalent to the following:

$$\nabla_i f(x^*) \begin{cases} \geq \lambda^*, & x_i^* = 0, \\ = \lambda^*, & x_i^* > 0, \end{cases} \quad i = 1, \dots, n.$$

Remark 1. Note that a feasible point x^* of problem (1) is stationary if and only if $\nabla f(x^*)^T(e_i - x^*) \geq 0$ for all $i = 1, \dots, n$, or equivalently,

$$\max\{0, -\nabla f(x^*)^T(e_i - x^*)\} = 0, \quad i = 1, \dots, n.$$

3 Active-Set Estimate

Given a stationary point x^* of problem (1), the active set can be defined as the set of inequality constraints binding at x^* . As mentioned above, since there is a one-to-one correspondence between inequality constraints and variables, we equivalently define as active set the set of zero components at x^* .

Definition 2. Let $x^* \in \mathbb{R}^n$ be a stationary point of problem (1). We define as active set the following set:

$$\bar{A}(x^*) = \{i \in \{1, \dots, n\} : x_i^* = 0\}. \quad (5)$$

We further define the nonactive set $\bar{N}(x^*)$ as the complement of $\bar{A}(x^*)$:

$$\bar{N}(x^*) = \{1, \dots, n\} \setminus \bar{A}(x^*) = \{i \in \{1, \dots, n\} : x_i^* > 0\}. \quad (6)$$

Now, we describe how, at any feasible point x , we estimate the active set. Following the approach proposed in [13, 15], we use a strategy that requires proper approximation of the KKT multipliers by means of the so called *multiplier functions*. To compute these multiplier functions, let (λ^*, μ^*) be the KKT multipliers associated to a given stationary point x^* . By (2), we have

$$\mu^* = \nabla f(x^*) - \lambda^* e,$$

then, multiplying by x^* and taking into account complementarity condition (3), we get

$$0 = (\mu^*)^T x^* = (\nabla f(x^*) - \lambda^* e)^T x^*.$$

From the feasibility of x^* , we obtain the following expressions for the multipliers:

$$\lambda^* = \nabla f(x^*)^T x^*, \quad \mu^* = \nabla f(x^*) - \lambda^* e,$$

so that we can introduce the following multiplier functions:

$$\lambda(x) = \nabla f(x)^T x, \quad (7)$$

$$\mu_i(x) = \nabla_i f(x) - \lambda(x), \quad i = 1, \dots, n. \quad (8)$$

Now, we can define our estimate of the active set.

Definition 3. Let $x \in \mathbb{R}^n$ be a feasible point of problem (1). We define the active-set estimate $A(x)$ and the nonactive-set estimate $N(x)$ as

$$A(x) = \{i: x_i \leq \epsilon \mu_i(x)\} = \{i: x_i \leq \epsilon \nabla f(x)^T (e_i - x)\}, \quad (9)$$

$$N(x) = \{i: x_i > \epsilon \mu_i(x)\} = \{i: x_i > \epsilon \nabla f(x)^T (e_i - x)\}, \quad (10)$$

where ϵ is a positive scalar.

By adapting Theorem 2.1 in [15], we can ensure that, in a neighborhood of a stationary point x^* , all the estimated active variables are active at x^* and include all active variables at x^* satisfying strict complementarity. We state this result in the following theorem.

Theorem 1. If (x^*, λ^*, μ^*) satisfies KKT conditions for problem (1), then there exists an open ball $\mathcal{B}(x^*, \rho)$ with center x^* and radius $\rho > 0$ such that, for each $x \in \mathcal{B}(x^*, \rho)$ we have

$$\{i: x_i^* = 0, \mu_i(x^*) > 0\} \subseteq A(x) \subseteq \bar{A}(x^*).$$

Furthermore, if strict complementarity holds, then

$$\{i: x_i^* = 0, \mu_i(x^*) > 0\} = A(x) = \bar{A}(x^*),$$

for each $x \in \mathcal{B}(x^*, \rho)$.

3.1 A Global Property of the Active-Set Estimate

Here, we analyze a global property of our active-set estimate. In particular, we show how, given a point $x \in \mathbb{R}^n$ feasible for problem (1), we can obtain a sufficient decrease in the objective function by setting the estimated active variables to zero. In the context of box constrained problems, this property has been already proved in [7, 9, 11, 12]. When dealing with problems over the standard simplex, we have to handle the presence of an equality constraint: in order to maintain feasibility, we need to update at least one estimated nonactive variable, so that all variables sum up to 1. The next proposition gives us a hint on how to choose the estimated nonactive variable that will be updated when setting to zero the active variables.

Proposition 1. Let $x \in \mathbb{R}^n$ be a feasible non-stationary point of problem (1) and define

$$J(x) = \left\{ j: j \in \underset{i=1, \dots, n}{\operatorname{Argmin}} \{ \nabla_i f(x) \} \right\}. \quad (11)$$

Then, $J(x) \subseteq N(x)$.

Proof. Since x is non-stationary, we have $|J(x)| < n$. Moreover, an index i must exist such that $x_i > 0$ and $\nabla_i f(x) > \nabla_j f(x)$, $j \in J(x)$. It follows that

$$\nabla f(x)^T x > \nabla_j f(x) e^T x = \nabla_j f(x).$$

Now, we can choose any index $j \in J(x)$ and set $\nu = j$. Recalling the definition of multipliers (7)–(8), we obtain

$$\mu_\nu(x) = \nabla_\nu f(x) - \lambda(x) = \nabla_\nu f(x) - \nabla f(x)^T x < \nabla_\nu f(x) - \nabla_\nu f(x) = 0 \leq x_\nu.$$

Since $x_\nu \geq 0$ and $\mu_\nu(x) < 0$, we have that $x_\nu > \epsilon \mu_\nu(x)$, and then $\nu \in N(x)$. \square

Remark 2. Proposition 1 implies that for every feasible non-stationary point x , the estimated nonactive set $N(x)$ is non-empty.

The main result of this section, reported in Proposition 2, shows that it is possible to get a significant decrease in the objective function by setting to zero the estimated active variables and suitably updating a variable chosen in the set defined in Proposition 1. To obtain this result, we first need an assumption on the parameter ϵ appearing in Definition 3.

Assumption 1. Assume that the parameter ϵ appearing in the estimates (9)–(10) satisfies the following conditions:

$$0 < \epsilon \leq \frac{2}{nL(2C + 1)},$$

where $C > 0$ is a given constant.

Proposition 2. Let Assumption 1 hold. Given a feasible non-stationary point x of problem (1), let $j \in J(x)$ and $I = \{1, \dots, n\} \setminus \{j\}$. Let $\hat{A}(x)$ be a set of indices such that $\hat{A}(x) \subseteq A(x)$. Let \tilde{x} be the feasible point defined as follows:

$$\tilde{x}_{\hat{A}(x)} = 0; \quad \tilde{x}_{I \setminus \hat{A}(x)} = x_{I \setminus \hat{A}(x)}; \quad \tilde{x}_j = x_j + \sum_{i \in \hat{A}(x)} x_i.$$

Then,

$$f(\tilde{x}) - f(x) \leq -CL\|\tilde{x} - x\|^2,$$

where $C > 0$ is the constant appearing in Assumption 1.

Proof. Define

$$\hat{A}^+ = \hat{A}(x) \cap \{i : x_i > 0\}. \quad (12)$$

Using known results on functions with Lipschitz continuous gradient [28], we can write

$$f(\tilde{x}) \leq f(x) + \nabla f(x)^T(\tilde{x} - x) + \frac{L}{2}\|\tilde{x} - x\|^2.$$

Adding and subtracting $CL\|\tilde{x} - x\|^2$ in the right-hand side of the above inequality, we get

$$f(\tilde{x}) \leq f(x) + \nabla f(x)^T(\tilde{x} - x) + \frac{L(2C + 1)}{2}\|\tilde{x} - x\|^2 - CL\|\tilde{x} - x\|^2. \quad (13)$$

In order to prove the proposition, we need to show that

$$\nabla f(x)^T(\tilde{x} - x) + \frac{L(2C + 1)}{2}\|\tilde{x} - x\|^2 \leq 0. \quad (14)$$

From the definition of \tilde{x} , we get

$$\begin{aligned} \|\tilde{x} - x\|^2 &= \sum_{i \in \hat{A}^+} (x_i)^2 + \left(\sum_{i \in \hat{A}^+} x_i \right)^2 \leq \sum_{i \in \hat{A}^+} (x_i)^2 + |\hat{A}^+| \sum_{i \in \hat{A}^+} (x_i)^2 \\ &= (|\hat{A}^+| + 1)x_{\hat{A}^+}^T x_{\hat{A}^+} \end{aligned} \quad (15)$$

and

$$\begin{aligned}\nabla f(x)^T(\tilde{x} - x) &= -\nabla_{\hat{A}^+} f(x)^T x_{\hat{A}^+} + \nabla_j f(x) \sum_{i \in \hat{A}^+} x_i \\ &= x_{\hat{A}^+}^T \left(\nabla_j f(x) e_{\hat{A}^+} - \nabla_{\hat{A}^+} f(x) \right).\end{aligned}\tag{16}$$

From the definition of the index j , we have that $\nabla_i f(x) \geq \nabla_j f(x)$ for all $i \in \{1, \dots, n\}$. Therefore, we can write

$$\sum_{i=1}^n \nabla_i f(x) x_i \geq \sum_{i=1}^n \nabla_j f(x) x_i = \nabla_j f(x) \sum_{i=1}^n x_i = \nabla_j f(x).\tag{17}$$

Recalling the active-set estimate and using (17), we have that

$$x_i \leq \epsilon \left(\nabla_i f(x) - \sum_{i=1}^n \nabla_i f(x) x_i \right) \leq \epsilon (\nabla_i f(x) - \nabla_j f(x)), \quad \forall i \in \hat{A}^+,$$

so that, by (15), we can write

$$\|\tilde{x} - x\|^2 \leq \epsilon (|\hat{A}^+| + 1) x_{\hat{A}^+}^T \left(\nabla_{\hat{A}^+} f(x) - \nabla_j f(x) e_{\hat{A}^+} \right).\tag{18}$$

From (16) and (18), we get

$$\begin{aligned}\nabla f(x)^T(\tilde{x} - x) + \frac{L(2C+1)}{2} \|\tilde{x} - x\|^2 &\leq x_{\hat{A}^+}^T [\nabla_j f(x) e_{\hat{A}^+} - \nabla_{\hat{A}^+} f(x)] + \\ &\quad + \frac{L(2C+1)}{2} (|\hat{A}^+| + 1) \epsilon x_{\hat{A}^+}^T \left(\nabla_{\hat{A}^+} f(x) - \nabla_j f(x) e_{\hat{A}^+} \right) \\ &= \left(\frac{L(2C+1)}{2} (|\hat{A}^+| + 1) \epsilon - 1 \right) x_{\hat{A}^+}^T \left(\nabla_{\hat{A}^+} f(x) - \nabla_j f(x) e_{\hat{A}^+} \right) \\ &\leq \left(\frac{L(2C+1)}{2} n \epsilon - 1 \right) x_{\hat{A}^+}^T \left(\nabla_{\hat{A}^+} f(x) - \nabla_j f(x) e_{\hat{A}^+} \right),\end{aligned}$$

where the last inequality follows from the non-negativity of

$$x_{\hat{A}^+}^T \left(\nabla_{\hat{A}^+} f(x) - \nabla_j f(x) e_{\hat{A}^+} \right)$$

(implied by (18)) and from the fact that $|\hat{A}^+| + 1 \leq n$ (implied by Proposition 1, see Remark 2). Then, (14) follows from the assumption we made on ϵ . \square

Remark 3. *In Assumption 1, the upper bound of ϵ depends on n . Actually, Proposition 2 still holds by replacing the constant n by $|\hat{A}^+| + 1$ in the upper bound of ϵ , where \hat{A}^+ is defined as in (12). This follows from the fact that n is only used to upper bound $|\hat{A}^+| + 1$ in the proof of Proposition 2. Note that, in general, \hat{A}^+ might be considerably smaller than n , but it depends on both the specific point x and ϵ itself. So, for the sake of simplicity, in Assumption 1 we use the constant n , even if all the theoretical results of the paper would hold by using $|\hat{A}^+| + 1$ instead.*

Remark 4. *From Assumption 1 and Proposition 2, we see that there is a trade-off, depending on the constant C , between the magnitude of the upper bound of ϵ and the decrease in the objective function guaranteed by Proposition 2. Namely, for small values of C , large values of ϵ can be used, and then, from (5), a major number of variables might be estimated active. But the corresponding decrease in the objective function might be small. Vice versa, for large values of C , we have the opposite situation.*

The property described in the above proposition is crucial for the analysis of the algorithm framework that we carry out in the next section. We remark that there is no way to get the same result from [13, 15], where a similar approach is used to estimate the active set. Indeed, in those papers the authors deal with non-linear inequality constraints and there is no such a result like Proposition 2. As a consequence, they cannot get the same algorithmic framework we describe in the present paper.

4 An Active-Set Algorithmic Framework for Minimization over the Simplex

In this section, we describe in depth an algorithmic framework that embeds the active-set estimate explained in the previous section. The framework performs two different steps at each iteration: the first one to update the estimated active variables, and the second one to update the estimated nonactive variables. The aim is to exploit as much as possible the properties of our estimate: first, the ability to identify those active variables satisfying strict complementarity after a sufficiently large number of iterations, according to the results in Theorem 1; second, the ability to get a decrease in the objective function when moving the variables according to Proposition 2.

In particular, let x^k be the point given at the beginning of a generic iteration k . In the first step, we compute the active and nonactive-set estimates $A(x^k)$, $N(x^k)$, and we generate the new feasible point \tilde{x}^k as indicated in Proposition 2: we set $\tilde{x}_{A(x^k)}$ to zero and we update \tilde{x}_j^k , with $j \in J(x^k)$ (all the other variables are not moved). Then, in the second step, we compute a search direction $d_{N(x^k)}^k$ in the subspace of the estimated nonactive variables, and we eventually perform a line search to get a new iterate x^{k+1} (the computation of the search direction and the line search are described below).

The detailed scheme of our algorithmic framework, that we name **AS-SIMPLEX**, is reported in Algorithm 1.

4.1 Global Convergence Analysis

In this section, we show the global convergence of **AS-SIMPLEX** to stationary points. First, we need to describe how we compute the search direction d^k and the stepsize α^k (lines 7 and 9 in Algorithm 1, respectively).

From now on, given any feasible point x^k generated by the algorithm and a feasible direction d^k , we call α_{\max}^k the maximum feasible stepsize that can be used along this direction. Taking inspiration from [2], in our framework we require the search direction to be *active-set gradient related*, according to the following definition:

Algorithm 1 Active-Set algorithmic framework for minimization over the simplex (AS-SIMPLEX)

- 1 Choose a feasible point x^0
 - 2 For $k = 0, 1, \dots$
 - 3 If x^k is a stationary point, then STOP
 - 4 Compute $A^k = A(x^k)$ and $N^k = N(x^k)$
 - 5 Compute $J^k = J(x^k)$, choose $j \in J^k$ and define $\tilde{N}^k = N^k \setminus \{j\}$
 - 6 Set $\tilde{x}_{A^k}^k = 0$, $\tilde{x}_{\tilde{N}^k}^k = x_{\tilde{N}^k}^k$ and $\tilde{x}_j^k = x_j^k + \sum_{h \in A^k} x_h^k$
 - 7 Compute a feasible direction d^k such that $d_{A^k}^k = 0$
 - 8 If $\nabla f(\tilde{x}^k)^T d^k < 0$ then
 - 9 Compute a stepsize $\alpha^k \in (0, \alpha_{\max}^k]$ by means of a line search
 - 10 Else
 - 11 Set $\alpha^k = 0$
 - 12 End if
 - 13 Set $x^{k+1} = \tilde{x}^k + \alpha^k d^k$
 - 14 End for
-

Definition 4. Given an infinite sequence of points $\{x^k\}$ produced by AS-SIMPLEX, we say that the sequence of directions $\{d^k\}$ is active-set gradient related if, for any subsequence $\{x^k\}_K$ such that $N(x^k) = \tilde{N}$ for all $k \in K$ and $\lim_{k \rightarrow \infty, k \in K} x^k = x^*$, where x^* is non-stationary in $\Delta_{\tilde{N}}$, we have that

$$\{d_{\tilde{N}}^k\}_K \text{ is bounded,} \quad (19)$$

$$\limsup_{k \rightarrow \infty, k \in K} \nabla_{\tilde{N}} f(\tilde{x}^k)^T d_{\tilde{N}}^k < 0, \quad (20)$$

$$\liminf_{k \rightarrow \infty, k \in K} \alpha_{\max}^k > 0. \quad (21)$$

By a slight abuse of terminology, in the following we say that a given direction d^k is active-set gradient related if it is an element of a sequence $\{d^k\}$ satisfying Definition 4. Moreover, in Subsection 4.2, we will provide some examples of active-set gradient related directions.

For what concerns the computation of the stepsize, a possibility is that of considering the classical Armijo line search (see, e.g., [2] and references therein). This method, which basically performs a successive stepsize reduction, allows to avoid the often considerable computation associated with an exact line search. Indeed, when dealing with some non-convex problems, even finding an approximate local minimizer along the search direction generally requires too many evaluations of the objective function and possibly the gradient.

The detailed scheme of the Armijo line search is reported in Algorithm 2.

Now, we are ready to show that AS-SIMPLEX globally converges to stationary points when the sequence of directions is active-set gradient related and the Armijo line search is used to compute the stepsize.

Theorem 2. Let Assumption 1 hold. Let $\{x^k\}$ be the sequence of points produced by AS-SIMPLEX, where the sequence of directions $\{d^k\}$ is active-set gradient related and the stepsize α^k

Algorithm 2 Armijo line search

- 0 Choose $\delta \in (0, 1)$, $\gamma \in (0, 1)$
 - 1 Set initial stepsize $\alpha = \alpha_{\max}^k$
 - 2 While $f(\tilde{x}^k + \alpha d^k) > f(\tilde{x}^k) + \gamma \alpha \nabla f(\tilde{x}^k)^T d^k$
 - 3 Set $\alpha = \delta \alpha$
 - 4 End while
-

is computed using the Armijo line search. Then, either an integer $\bar{k} \geq 0$ exists such that $x^{\bar{k}}$ is a stationary point for problem (1), or the sequence $\{x^k\}$ is infinite and every limit point x^* of the sequence is a stationary point for problem (1).

Proof. Let $\{x^k\}$ be the sequence produced by AS-SIMPLEX and let us assume that a stationary point is not produced in a finite number of iterations. First note that, from the instructions of the algorithm and Proposition 2, we can write

$$f(x^{k+1}) \leq f(\tilde{x}^k) \leq f(x^k) - CL\|\tilde{x}^k - x^k\|^2.$$

Therefore, from the continuity of the objective function and the compactness of the feasible set, it follows that

$$\lim_{k \rightarrow \infty} [f(x^{k+1}) - f(x^k)] = 0, \quad (22)$$

$$\lim_{k \rightarrow \infty} \|\tilde{x}^k - x^k\| = 0. \quad (23)$$

Since the feasible set is compact, then the sequence $\{x^k\}$ attains a limit point x^* and, recalling (23), there exists $K \subseteq \mathbb{N}$ such that

$$\lim_{k \rightarrow \infty, k \in K} x^k = \lim_{k \rightarrow \infty, k \in K} \tilde{x}^k = x^*. \quad (24)$$

Taking into account the structure of the feasible set, we can characterize a stationary point x using the following condition (see Remark 1):

$$\nabla f(x)^T (e_i - x) \geq 0, \quad \forall i \in \{1, \dots, n\}.$$

Let $\Phi_i(x)$ be the continuous function defined as

$$\Phi_i(x) = \max\{0, -\nabla f(x)^T (e_i - x)\}, \quad i = 1, \dots, n,$$

that measures the violation of the stationarity conditions for a variable x_i , $i = 1, \dots, n$.

By contradiction, we assume that x^* is non-stationary, so that an index $\nu \in \{1, \dots, n\}$ exists such that

$$\Phi_\nu(x^*) > 0. \quad (25)$$

Taking into account that the number of possible different choices of A^k and N^k is finite, we can find a subset of iteration indices $\bar{K} \subseteq K$ such that $A^k = \hat{A}$ and $N^k = \hat{N}$ for all $k \in \bar{K}$.

First, suppose that $\nu \in \hat{A}$. Then, by Definition 3, we can write

$$0 \leq x_\nu^k \leq \epsilon \nabla f(x^k)^T (e_\nu - x^k),$$

so that $\Phi_\nu(x^k) = \max\{0, -\nabla f(x^k)^T(e_\nu - x^k)\} = 0$, for all $k \in \bar{K}$. Therefore, from (24) and the continuity of the function $\Phi_i(\cdot)$, we get a contradiction with (25).

Then, ν necessarily belongs to \hat{N} , that is, $x_{\hat{N}}^*$ is non-stationary in $\Delta_{\hat{N}}$, where $\Delta_{\hat{N}}$ is given as in Definition 4. From that definition and the fact that $d_{\hat{A}}^k = 0$, we also have that $\{d^k\}_K$ is bounded. Then, there exists an infinite subset of K (that we still denote by K without loss of generality) such that

$$\lim_{k \rightarrow \infty, k \in K} d^k = \bar{d}. \quad (26)$$

Moreover, exploiting again Definition 4, we have that $\eta > 0$ and $M > 0$ exist such that

$$\limsup_{k \rightarrow \infty, k \in K} \nabla f(\tilde{x}^k)^T d^k = -\eta, \quad (27)$$

$$\alpha_{\max}^k \geq M > 0, \quad \forall k \text{ sufficiently large, } k \in K. \quad (28)$$

From (27), it follows that $\hat{k} \in K$ exists such that $\nabla f(\tilde{x}^k)^T d^k < 0$, for $k \geq \hat{k}$, $k \in K$. Then, according to line 9 of Algorithm 1, the Armijo line search computes a value $\alpha^k \in (0, \alpha_{\max}^k]$ for all $k \geq \hat{k}$, such that

$$f(x^{k+1}) \leq f(\tilde{x}^k) + \gamma \alpha^k \nabla f(\tilde{x}^k)^T d^k, \quad \forall k \geq \hat{k}, k \in K, \quad (29)$$

or equivalently,

$$f(\tilde{x}^k) - f(x^{k+1}) \geq \gamma \alpha^k |\nabla f(\tilde{x}^k)^T d^k|, \quad \forall k \geq \hat{k}, k \in K.$$

From (22) and (23), we get that the left-hand side of the above inequality converges to zero for $k \rightarrow \infty$, hence

$$\lim_{k \rightarrow \infty, k \in K} \alpha^k |\nabla f(\tilde{x}^k)^T d^k| = 0. \quad (30)$$

Using (27), we obtain that $\lim_{k \rightarrow \infty, k \in K} \alpha^k = 0$. Taking into account (28), it follows that there exists $\bar{k} \in K$, $\bar{k} \geq \hat{k}$, such that

$$\alpha^k < \alpha_{\max}^k, \quad \forall k \geq \bar{k}, k \in K.$$

In other words, for $k \geq \bar{k}$, $k \in K$, the stepsize α^k cannot be set equal to the maximum feasible stepsize and, taking into account the line search procedure, we can write

$$f\left(\tilde{x}^k + \frac{\alpha^k}{\delta} d^k\right) > f(\tilde{x}^k) + \gamma \frac{\alpha^k}{\delta} \nabla f(\tilde{x}^k)^T d^k, \quad \forall k \geq \bar{k}, k \in K. \quad (31)$$

We can apply the mean value theorem and we have that $\xi_k \in (0, 1)$ exists such that

$$f\left(\tilde{x}^k + \frac{\alpha^k}{\delta} d^k\right) = f(\tilde{x}^k) + \frac{\alpha^k}{\delta} \nabla f\left(\tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k\right)^T d^k, \quad \forall k \geq \bar{k}, k \in K. \quad (32)$$

By substituting (32) within (31), we have

$$\nabla f\left(\tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k\right)^T d^k > \gamma \nabla f(\tilde{x}^k)^T d^k, \quad \forall k \geq \bar{k}, k \in K. \quad (33)$$

From (24), and exploiting the fact that $\{\xi_k\}$, $\{\alpha^k\}$ and $\{d^k\}$ are bounded, we get

$$\lim_{k \rightarrow \infty, k \in K} \tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k = \lim_{k \rightarrow \infty, k \in K} \tilde{x}^k = x^*.$$

Therefore, taking the limits in (31) and (32), and taking into account (26), we obtain that $\nabla f(x^*)^T \bar{d} \geq \gamma \nabla f(x^*)^T \bar{d}$, or equivalently,

$$(1 - \gamma) \nabla f(x^*)^T \bar{d} \geq 0.$$

Since $\gamma \in (0, 1)$, it follows that $\nabla f(x^*)^T \bar{d} \geq 0$, contradicting (27). Hence, we get $\Phi_i(x^*) = 0$, for all $i = 1, \dots, n$ and x^* is a stationary point for problem (1). \square

Remark 5. *Theorem 2 holds when using as stepsize in AS-SIMPLEX any value $\alpha^k \in (0, \alpha_{max}^k]$ such that*

$$f(\tilde{x}^k + \alpha^k d^k) \leq f(\tilde{x}^k + \alpha_A^k d^k),$$

where α_A^k is the value computed by the Armijo line search. It follows from the fact that, if the above relation is satisfied, then (29) holds, as well as all the subsequent steps in the proof.

In particular, this implies that Theorem 2 holds under the assumption that the stepsize is computed in AS-SIMPLEX by means of an exact line search, that is, α^k is computed as

$$\alpha^k \in \underset{\alpha \in (0, \alpha_{max}^k]}{\text{Argmin}} f(\tilde{x}^k + \alpha d^k).$$

4.2 Active-Set Gradient Related Directions in AS-SIMPLEX

As shown in Theorem 2, to ensure global convergence of AS-SIMPLEX to stationary points we need a sequence of directions that satisfies Definition 4. Moreover, from the instructions of the algorithm, $d_{A^k}^k$ must be zero at each iteration, so we only need to focus on the computation of $d_{N^k}^k$.

As examples of $d_{N^k}^k$ that can be used in practice, we consider Frank-Wolfe-type and projected gradient directions:

(FW) *Frank-Wolfe direction:*

$$d_{N^k}^{\text{FW}} = (e_{\hat{i}} - \tilde{x}^k)_{N^k}, \quad \hat{i} \in \underset{i \in N^k}{\text{Argmin}} \{ \nabla_i f(\tilde{x}^k) \}; \quad (34)$$

(AFW) *away-step Frank-Wolfe direction:*

$$d_{N^k}^{\text{AFW}} = \begin{cases} d_{N^k}^{\text{FW}}, & \text{if } \nabla_{N^k} f(\tilde{x}^k)^T d_{N^k}^{\text{FW}} \leq \nabla_{N^k} f(\tilde{x}^k)^T d_{N^k}^{\text{A}}, \\ d_{N^k}^{\text{A}}, & \text{otherwise,} \end{cases}$$

where

$$d_{N^k}^{\text{A}} = (\tilde{x}^k - e_{\hat{j}})_{N^k}, \quad \hat{j} \in \underset{j \in N_0^k}{\text{Argmax}} \{ \nabla_j f(\tilde{x}^k) \} \quad (35)$$

and $N_0^k = \{j \in N^k: \tilde{x}_j^k > 0\}$;

(PG) *projected gradient direction:*

$$d_{N^k}^{\text{PG}} = \left(P(\tilde{x}^k - s\nabla f(\tilde{x}^k))_{\Delta_{N^k}} - \tilde{x}^k \right)_{N^k},$$

where $s > 0$ is a fixed scalar.

In the following, we will refer to d^{FW} , d^{AFW} and d^{PG} when the subdirection $d_{N^k}^k$ is chosen according to the Frank-Wolfe (FW), the away-step Frank-Wolfe (AFW) or the projected gradient (PG) rule, respectively. We now show that the three considered directions satisfy Definition 4.

Proposition 3. *Given one rule among (FW), (AFW) and (PG) for the computation of $d_{N^k}^k$, the resulting sequence of directions $\{d^k\}$ generated by **AS-SIMPLEX** is active-set gradient related.*

Proof. Since Δ is compact, it is easy to see that all the considered directions are bounded. Considering some of the ideas reported in [2] (see chapter 2) and the properties related to the active-set estimate, we now prove that those directions also satisfy (20) and (21). Let $\{x^k\}_K$ be a subsequence such that $N^k = \hat{N}$ for all $k \in K$ and $\lim_{k \rightarrow \infty, k \in K} x_{\hat{N}}^k = x_{\hat{N}}^*$, where x^* is non-stationary in $\Delta_{\hat{N}}$. We consider the different cases:

(FW) By definition of the index \hat{i} given in (34), it is easy to see that, for all $k \geq 0$, we have $\nabla_{\hat{i}} f(\tilde{x}^k) \leq \nabla_{N^k} f(\tilde{x}^k)^T x_{N^k}$, $\forall x \in \Delta_{N^k}$. Thus,

$$\begin{aligned} \nabla_{\hat{N}} f(\tilde{x}^k)^T d_{\hat{N}}^k &= \nabla_{\hat{N}} f(\tilde{x}^k)^T (e_{\hat{i}} - \tilde{x}^k)_{\hat{N}} \\ &\leq \nabla_{\hat{N}} f(\tilde{x}^k)^T (x - \tilde{x}^k)_{\hat{N}} = \nabla f(\tilde{x}^k)^T (x - \tilde{x}^k), \quad \forall x \in \Delta_{\hat{N}}, \end{aligned}$$

where the last equality follows from the fact that $x_{A^k} = 0$ for all $x \in \Delta_{N^k}$ and $\tilde{x}_{A^k}^k = 0$. Passing to the limit, we obtain

$$\limsup_{k \rightarrow \infty, k \in K} \nabla_{\hat{N}} f(\tilde{x}^k)^T d_{\hat{N}}^k \leq \nabla f(x^*)^T (x - x^*), \quad \forall x \in \Delta_{\hat{N}}.$$

Since x^* is non-stationary in $\Delta_{\hat{N}}$, we have that

$$\min_{x \in \Delta_{\hat{N}}} \nabla f(x^*)^T (x - x^*) < 0.$$

Therefore, combining the above two inequalities, (20) holds. Since $\alpha_{\max}^k = 1$ at every iteration, also (21) is trivially satisfied.

(AFW) Since, by definition, $\nabla f(\tilde{x}^k)^T d^{\text{AFW}} \leq \nabla f(\tilde{x}^k)^T d^{\text{FW}}$, we can repeat the same reasoning given above for the (FW) case and (20) holds. To prove (21), by contradiction let us assume that an infinite subset of K (that we still denote by K for the sake of simplicity) exists such that

$$\lim_{k \rightarrow \infty, k \in K} \alpha_{\max}^k = 0. \tag{36}$$

Recalling the definition of d^{AFW} , the case we need to analyze is the one where we get an infinite subsequence of away-step directions in \hat{N} (as $\alpha_{\max}^k = 1$ for Frank-Wolfe directions). So, we assume that an infinite subset $\tilde{K} \subseteq K$ exists such that

$$d_{\hat{N}}^k = d_{\hat{N}}^A, \quad \forall k \in \tilde{K}.$$

We have that $\alpha_{\max}^k = \frac{\tilde{x}_j^k}{1 - \tilde{x}_j^k}$, for all $k \in \tilde{K}$, where \hat{j} is the index computed according to (35). Since the number of indices in \hat{N} is finite, we can consider an infinite subset of \tilde{K} (that we still denote by \tilde{K} for the sake of simplicity) where the index \hat{j} is fixed. Taking into account (36), it is easy to see that $\{\tilde{x}_j^k\}_{\tilde{K}} \rightarrow 0$. Using (23), we get

$$\lim_{k \rightarrow \infty, k \in \tilde{K}} x_{\hat{j}}^k = 0. \quad (37)$$

Moreover, from (20), (23) and the continuity of $\nabla f(x)$, we can write

$$\limsup_{k \rightarrow \infty, k \in K} \nabla_{\hat{N}} f(x^k)^T d_{\hat{N}}^k = -\eta.$$

Exploiting the fact that $\nabla f(x^k)^T d^k = \nabla_{\hat{N}} f(x^k)^T d_{\hat{N}}^k$ by definition of d^k , we obtain

$$\begin{aligned} -\eta &= \limsup_{k \rightarrow \infty, k \in K} \nabla f(x^k)^T d^k \\ &= \limsup_{k \rightarrow \infty, k \in K} [\nabla f(x^k)^T (\tilde{x}^k - x^k) + \nabla f(x^k)^T (x^k - e_j)] \\ &= \limsup_{k \rightarrow \infty, k \in K} \nabla f(x^k)^T (x^k - e_j), \end{aligned} \quad (38)$$

where the last equality follows again from (23). From (37) and (38), an index $\tilde{k} \in \tilde{K}$ exists such that, for all $k \geq \tilde{k}$, $k \in \tilde{K}$, we have

$$\begin{aligned} \nabla f(x^k)^T (x^k - e_j) &\leq -\frac{\eta}{2}, \\ x_{\hat{j}}^k &\leq \frac{\eta}{2}. \end{aligned}$$

Therefore,

$$x_{\hat{j}}^k \leq \epsilon \nabla f(x^k)^T (e_j - x^k), \quad \forall k \geq \tilde{k}, k \in \tilde{K}.$$

Recalling (9), this implies that $\hat{j} \notin \hat{N}$ and, considering the definition of \hat{j} in (35), we get a contradiction.

(PG) Let us define $\hat{x}^k = P(\tilde{x}^k - s \nabla f(\tilde{x}^k))_{\Delta_{N^k}}$, so that $d^k = \hat{x}^k - \tilde{x}^k$. By continuity of the projection operator, we get

$$\lim_{k \rightarrow \infty, k \in K} \hat{x}^k = P(x^* - s \nabla f(x^*))_{\Delta_{\hat{N}}}.$$

From the properties of the projection, we have

$$(\tilde{x}^k - s\nabla f(\tilde{x}^k) - \hat{x}^k)^T(x - \hat{x}^k) \leq 0, \quad \forall x \in \Delta_{N^k}.$$

If we choose $x = \tilde{x}^k$ in the above inequality, we can write

$$\nabla f(\tilde{x}^k)^T d^k = \nabla f(\tilde{x}^k)^T(\hat{x} - \tilde{x}^k) \leq -\frac{1}{s}\|\hat{x}^k - \tilde{x}^k\|^2 = -\frac{1}{s}\|d^k\|^2, \quad \forall k \geq 0. \quad (39)$$

Since $d_{A^k}^k = 0$ for all $k \geq 0$, taking the limit we have

$$\begin{aligned} \limsup_{k \rightarrow \infty, k \in K} \nabla_{\hat{N}} f(\tilde{x}^k)^T d_{\hat{N}}^k &= \limsup_{k \rightarrow \infty, k \in K} \nabla f(\tilde{x}^k)^T d^k \\ &\leq -\frac{1}{s}\|P(x^* - s\nabla f(x^*))_{\Delta_{\hat{N}}} - x^*\|^2. \end{aligned}$$

From the fact that x^* is non-stationary in $\Delta_{\hat{N}}$, it follows that

$$\|P(x^* - s\nabla f(x^*))_{\Delta_{\hat{N}}} - x^*\| > 0.$$

Therefore,

$$\limsup_{k \rightarrow \infty, k \in K} \nabla_{\hat{N}} f(\tilde{x}^k)^T d_{\hat{N}}^k < 0,$$

implying that (20) holds. Finally, since $\alpha_{\max}^k = 1$ at every iteration, also (21) is trivially satisfied. □

Remark 6. *Since we set $\tilde{x}_{A^k}^k = 0$ at any iteration k , it is straightforward to verify that, when d^k is computed according to (FW), (AFW) or (PG) rule, $\nabla f(\tilde{x}^k)^T d^k < 0$ if and only if \tilde{x}^k is non-stationary on Δ_{N^k} . Equivalently, $\nabla_{N^k} f(\tilde{x}^k)^T d_{N^k}^k < 0$ if and only if $\tilde{x}_{N^k}^k$ is non-stationary on the subspace variable N^k .*

5 Convergence Rate Analysis

In this section, we analyze the convergence rate of AS-SIMPLEX when one rule among (FW), (AFW) and (PG) is used for the computation of d^k . More specifically, we focus on particular classes of non-convex problems (i.e., problems satisfying some specific assumptions we will make later on), and report linear convergence results for our framework when using those three directions. The results are asymptotic since they exploit the properties of the active-set estimate given in Theorem 1, and these properties hold only in a neighborhood of stationary points. Summarizing, on the one hand, we get asymptotic linear rate, but, on the other hand, our results hold for non-convex objective functions.

We make an assumption that is pretty common when analyzing the convergence rate of algorithms (see, e.g., [30]), and quite reasonable, taking into account the results reported in the previous section.

Assumption 2. Let $\{x^k\}$ be the infinite sequence generated by *AS-SIMPLEX*. We have that

$$\lim_{k \rightarrow \infty} x^k = x^*,$$

where x^* is a stationary point of problem (1).

From now on, we denote with \bar{I} the set $\{1, \dots, n\}$. We also denote with \bar{A} and \bar{N} the index sets defined in (5) and (6), respectively, and with

$$N^+ := \bar{N} \cup \{i \in \bar{I} : x_i^* = 0, \mu_i^* = 0\} \quad \text{and} \quad A^+ := \bar{I} \setminus N^+ = \{i \in \bar{I} : x_i^* = 0, \mu_i^* > 0\}.$$

5.1 Linear Convergence of Active-Set Frank-Wolfe

Here we show that, when the Frank-Wolfe direction (FW) is embedded in our active-set framework, one can get asymptotic linear convergence without making the classic assumptions (see, e.g., [19]) needed for proving linear convergence rate of the classical Frank-Wolfe method, that is:

- optimal solution in the interior of the feasible set,
- strongly convex objective function.

As we will see, those assumptions are replaced by strict complementarity in the optimal solution and strong convexity on $\Delta_{\bar{N}}$, respectively. Again, we remark that the results are asymptotic, but they do not require convexity assumptions of the objective function on the whole Δ . Moreover, we obtain pretty tight convergence rate constants, that get much tighter than those obtained with the classical Frank-Wolfe method as the final solution is sparse. So, we can consider the result as a good trade-off in the end.

Before reporting the theoretical results related to the active-set Frank-Wolfe (i.e., *AS-SIMPLEX* with d^k computed according to (FW) rule), we need to introduce some constants, which follow from those used in [24], adapted to our purposes. Given an index subset $I \subseteq \bar{I}$, we define:

$$C_f(I) := \sup_{\substack{x, s \in \Delta_I, \\ \alpha \in (0, 1], \\ y = x + \alpha(s - x)}} \frac{2}{\alpha^2} \left[f(y) - f(x) - \nabla f(x)^T (y - x) \right],$$

$$\mu_f(I) := \inf_{\substack{x \in \Delta_I \setminus \{x^*\}, \\ \alpha \in (0, 1], \\ \bar{s} = \bar{s}(x, x^*, \Delta), \\ y = x + \alpha(\bar{s} - x)}} \frac{2}{\alpha^2} \left[f(y) - f(x) - \nabla f(x)^T (y - x) \right],$$

where $\bar{s}(x, x^*, \Delta) := \text{ray}(x, x^*) \cap \partial(\Delta)$ and $\text{ray}(x, x^*)$ is the ray from x to x^* . The curvature constant $C_f(I)$, which measures the non-linearity of the objective function in the subspace Δ_I , is needed to give a quadratic upper bound on the objective function. The strong convexity constant $\mu_f(I)$, which measures the strong convexity of the objective function on Δ_I (and can be interpreted as the lower curvature of the function), is used to give a quadratic lower bound instead (see [24] for further details).

Remark 7. *The main difference between the constants given above and those introduced in [24] is that ours are restricted to a particular subspace. Moreover, for any index subset $I \subseteq \bar{I}$, it is easy to see that*

$$\mu_f(\bar{I}) \leq \mu_f(I) \leq C_f(I) \leq C_f(\bar{I}). \quad (40)$$

Now, we are ready to state linear convergence rate of AS-SIMPLEX when (FW) rule is used to compute the search direction.

Theorem 3. *Let Assumption 1 and 2 hold, let $f(x)$ be strongly convex on $\Delta_{\bar{N}}$, and let us assume that strict complementarity holds at x^* . Let us further assume that d^k is computed by (FW) rule and that the exact line search is used.*

Then, there exists \bar{k} such that

$$f(x^{k+1}) - f(x^*) \leq (1 - \rho^{AS-FW}) [f(x^k) - f(x^*)], \quad \forall k \geq \bar{k},$$

where

$$\rho^{AS-FW} = \min \left\{ \frac{1}{2}, \frac{\mu_f(\bar{N})}{C_f(\bar{N})} \right\}.$$

Proof. From Theorem 1, exploiting the fact that strict complementarity holds at x^* , for sufficiently large k we have that

$$N(x^k) = N(\tilde{x}^k) = \bar{N} \quad \text{and} \quad A(x^k) = A(\tilde{x}^k) = \bar{A}.$$

From the instructions of AS-SIMPLEX, for sufficiently large k we have $\tilde{x}_A^k = x_A^k = 0$, implying that $\tilde{x}^k = x^k$. Then, for sufficiently large k the minimization is restricted to the variable subspace $N^k = \bar{N}$. Since the search direction d^k is computed according to (FW) rule, the rest of the proof follows by repeating the same arguments of the proof given for Theorem 3 in [24], observing that $\mu_f(\bar{N}) > 0$ and $C_f(\bar{N}) < \infty$ under the hypothesis we made. \square

Remark 8. *From (40), it follows that the smaller \bar{N} (i.e., the sparser x^*), the better the convergence rate of AS-SIMPLEX. Moreover,*

$$\rho^{AS-FW} \geq \min \left\{ \frac{1}{2}, \frac{\mu_f(\bar{I})}{C_f(\bar{I})} \right\} = \rho^{FW},$$

where ρ^{FW} is the constant given in [24] for the convergence rate of the standard Frank-Wolfe method.

5.2 Linear Convergence of Active-Set Away-Step Frank-Wolfe

In this subsection, we prove that active-set away-step Frank-Wolfe (i.e., AS-SIMPLEX with d^k computed according to (AFW) rule) asymptotically converges at linear rate. We can prove the result without making the strong convexity assumption (see, e.g., [19]) needed for proving linear convergence rate of the classical away-step Frank-Wolfe method. As we will see, that assumption is replaced by strong convexity of the objective function on Δ_{N^+} . Similarly to the (FW) direction, here we get asymptotic results, but we do not need strong convexity

assumptions of the objective function on the whole Δ and we obtain pretty tight convergence rate constants that depend on the sparsity of the final solution. Again, we can consider the result as a good trade-off in the end.

Given an index subset $I \subseteq \bar{I}$, we define the following two constants, which follow from those used in [25], adapted to our purposes:

$$C_f^\Delta(I) := \sup_{\substack{x,s,v \in \Delta_I \\ \alpha \in (0,1], \\ y=x+\alpha(s-v)}} \frac{2}{\alpha^2} \left[f(y) - f(x) - \alpha \nabla f(x)^T (s-v) \right],$$

$$\mu_f^\Delta(I) := \inf_{x \in \Delta_I} \inf_{\substack{\hat{x} \in \Delta_I \\ \nabla f(x)^T (\hat{x}-x) < 0}} \frac{2}{\alpha_I^\Delta(x, \hat{x})^2} \left[f(\hat{x}) - f(x) - \nabla f(x)^T (\hat{x}-x) \right],$$

where

$$\alpha_I^\Delta(x, \hat{x}) := \frac{\nabla f(x)^T (\hat{x}-x)}{\nabla f(x)^T (s_I(x) - v_I(x))},$$

$$s_I(x) := e_{\hat{i}}, \quad \hat{i} \in \underset{i \in I}{\text{Argmin}} \{ \nabla_i f(x) \},$$

$$v_I(x) := e_{\hat{j}}, \quad \hat{j} \in \underset{j \in I: x_j > 0}{\text{Argmax}} \{ \nabla_j f(x) \}.$$

These two new constants are motivated in the analysis by the fact that both Frank-Wolfe and away-step directions are used (see [25] for further details).

Remark 9. *Also in this case, the main difference between the constants given above and those introduced in [25] is that ours are restricted to a particular subspace. Moreover, for any index subset $I \subseteq \bar{I}$, it is easy to see that the following inequalities hold:*

$$\mu_f^\Delta(\bar{I}) \leq \mu_f^\Delta(I) \leq C_f^\Delta(I) \leq C_f^\Delta(\bar{I}). \quad (41)$$

Theorem 8 in [25] shows, for the standard away-step Frank-Wolfe method, that the quantity $[f(x^k) - f(x^*)]$ decreases linearly at each iteration k that is not a so-called *drop step*. Iteration k is a drop step when the stepsize $\alpha^k = \alpha_{\max}^k < 1$ and the number of zero components in x^{k+1} increases by one. In the convergence rate analysis, these iterations are troublesome since a geometric decrease of $[f(x^k) - f(x^*)]$ cannot be guaranteed.

In our context, these definitions apply when considering the computation of x^{k+1} from \tilde{x}^k and, as to be shown in the next theorem, we can still guarantee that the quantity $[f(x^k) - f(x^*)]$ decreases linearly at each iteration k that is a *good step* (i.e., not a drop step) with tighter constants (that depend on the sparsity of the optimal solution).

Theorem 4. *Let Assumption 1 and 2 hold, let $f(x)$ be strongly convex on Δ_{N+} , with $\nabla f(x)$ Lipschitz continuous on $\Delta_{N+} + (\Delta_{N+} - \Delta_{N+})$ (in the Minkowski sense). Let us further assume that d^k is computed by (AFW) rule and that the exact line search is used.*

Then, there exists \bar{k} such that, for every iteration $k \geq \bar{k}$ that is a good step (i.e., it is not a drop step), we have

$$f(x^{k+1}) - f(x^*) \leq (1 - \rho^{AFW}) [f(x^k) - f(x^*)],$$

where

$$\rho^{AFW} = \frac{\mu_f^\Delta(N^+)}{4C_f^\Delta(N^+)}, \quad (42)$$

Moreover, for $k \geq \bar{k}$, we have that at most $|N^+| - 1$ drop steps can be performed in between two good steps.

Proof. First, we observe that Theorem 1 implies that an iteration \tilde{k} exists such that $A^k \supseteq A^+$ and $N^k \subseteq N^+$ for $k \geq \tilde{k}$. Now, we show that there exists $\bar{k} \geq \tilde{k}$ such that

- (i) $x_{A^+}^k = \tilde{x}_{A^+}^k = 0$, for all $k \geq \bar{k}$;
- (ii) $\nabla f(\tilde{x}^k)^T d^k < 0$, for all $k \geq \bar{k}$;
- (iii) $x^* \in \underset{x \in \Delta_{N^k}}{\text{Argmin}} f(x)$, for all $k \geq \bar{k}$.

Point (i) follows from the instructions of the algorithm and the fact that $A^k \supseteq A^+$, for $k \geq \tilde{k}$. To prove point (ii), we proceed by contradiction. We assume that an infinite subsequence $\{\tilde{x}^k\}_K$ exists such that $\nabla f(\tilde{x}^k)^T d^k = 0$ for all $k \in K$. Recalling Remark 6, this means that \tilde{x}^k is stationary over Δ_{N^k} (but \tilde{x}^k is not stationary over Δ), for all $k \in K$. Since $N^k \subseteq N^+$ for $k \geq \tilde{k}$ and $f(x)$ is strongly convex on Δ_{N^+} , there exists a unique point satisfying stationarity conditions over Δ_{N^k} for $k \geq \tilde{k}$. Taking into account that A^k and N^k are subsets of a finite set of indices and $\tilde{x}_{A^k}^k = 0$, we have that, after a finite number of iterations, the algorithm should cycle. This cannot be possible as we guarantee a strict decrease in the objective function at each iteration. Point (iii) follows from the fact that $A^k \subseteq \bar{A}$ for all $k \geq \tilde{k}$ and $f(x)$ is strongly convex on Δ_{N^+} .

Consequently, recalling that $d_{A^k}^k = 0$, for $k \geq \bar{k}$ the minimization is restricted to the variable subspace $N^k \subseteq N^+$. We can thus repeat the same arguments of the proof given for Theorem 8 in [25] to provide the following bound for all $k \geq \bar{k}$:

$$f(x^{k+1}) - f(x^*) \leq (1 - \rho^{AFW})[f(\tilde{x}^k) - f(x^*)] \leq (1 - \rho^{AFW})[f(x^k) - f(x^*)],$$

where the last inequality follows from the fact that $f(\tilde{x}^k) \leq f(x^k)$. Moreover, we have that $\mu_f^\Delta(N^+) > 0$ and $C_f^\Delta(N^+) < \infty$ under the hypothesis we made.

Finally, to bound the number of iterations for which k is not a good step, we need to consider those iterations such that $\alpha^k = \alpha_{\max}^k < 1$, for $k \geq \bar{k}$. The fact that $\alpha_{\max}^k < 1$ implies that $d^k = d^A$. Consequently, when $\alpha^k = \alpha_{\max}^k$, we have that $x_{\hat{j}}^{k+1} = 0$, where \hat{j} is the index computed according to (35). In other words, the number of zero components in x^{k+1} increases by 1 (since $d_i^k = 0$ for all i such that $\tilde{x}_i^k = 0$, i.e., the away-step direction does not change zero components). From the instructions of the algorithm, we also have that the number of zero components in \tilde{x}^{k+1} cannot decrease from x^{k+1} . Combining these observations with the fact that $\tilde{x}_{A^+}^k = 0$ for all $k \geq \bar{k}$, we conclude that after at most $|N^+| - 1$ iterations with $\alpha^k = \alpha_{\max}^k < 1$, a point \tilde{x}^k with $n - 1$ zero components is produced. Of course, we cannot further increase the number of zero components. \square

Remark 10. From (41), it follows that the smaller N^+ , the better the convergence rate of AS-SIMPLEX. Moreover,

$$\rho^{AS-AFW} \geq \frac{\mu_f^\Delta(\bar{I})}{4C_f^\Delta(\bar{I})} = \rho^{AFW},$$

where ρ^{AFW} is the constant given in [25] for the convergence rate of the standard away-step Frank-Wolfe. Furthermore, also the upper bound on the number of bad steps between two good steps depends on the cardinality of N^+ (for sufficiently large k). We would like to recall that, in the standard away-step Frank-Wolfe, this value is equal to $n - 1$.

5.3 Linear Convergence of Active-Set Projected Gradient

In this subsection, we prove that the active-set Projected Gradient (i.e., AS-SIMPLEX with d^k computed according to (PG) rule) asymptotically converges at a linear rate, under the assumption that the objective function is strictly convex on Δ_{N^+} . We follow the same arguments of the proof of Theorem 3.1 in [26]. First, we need to give two additional results, stated in Lemma 1 and Lemma 2.

Lemma 1. Let $\{x^k\}$ be the sequence produced by Algorithm 1. Then, there exists \bar{k} such that

$$f(\tilde{x}^k) - f(x^*) \leq \frac{L}{2} \|\tilde{x}^k - x^*\|^2, \quad \forall k \geq \bar{k}.$$

Proof. From the Lipschitz continuity of the gradient, for all $k \geq 0$ we can write

$$f(\tilde{x}^k) - f(x^*) \leq \nabla f(x^*)^T (\tilde{x}^k - x^*) + \frac{L}{2} \|\tilde{x}^k - x^*\|^2.$$

From Theorem 1, an iteration \bar{k} exists such that $A^k \supseteq A^+$ and $\bar{N} \subseteq N^k$ for all $k \geq \bar{k}$. Hence, from the first-order necessary optimality conditions, $\nabla_i f(x^*) = \lambda^*$ for all $i \in N^k$ and for all $k \geq \bar{k}$. Since $\tilde{x}_i^k = x_i^* = 0$ for all $i \in A^k$ and for all $k \geq \bar{k}$, we get

$$\nabla f(x^*)^T (\tilde{x}^k - x^*) = \sum_{i \in N^k} \lambda^* (\tilde{x}^k - x^*)_i = 0, \quad \forall k \geq \bar{k},$$

where the last equality follows from the feasibility of \tilde{x}^k and x^* . Therefore, for all $k \geq \bar{k}$ we obtain $f(\tilde{x}^k) - f(x^*) \leq L/2 \|\tilde{x}^k - x^*\|^2$. \square

Lemma 2. Let $\{x^k\}$ be the sequence produced by Algorithm 1, where d^k is computed by (PG) rule and the Armijo line search is used. Then, at any iteration k such that $\nabla f(\tilde{x}^k)^T d^k < 0$ we have

$$\alpha^k \geq R, \tag{43}$$

$$f(\tilde{x}^k) - f(x^{k+1}) \geq \frac{\gamma R \min\{1, s\}^2}{s} \|P(\tilde{x}^k - \nabla f(\tilde{x}^k))_{\Delta_{N^k}} - \tilde{x}^k\|^2, \tag{44}$$

where $R = \min\left\{1, \frac{2\delta(1-\gamma)}{sL}\right\}$ and δ, γ are the parameters used in the Armijo line search.

Proof. Let k be an iteration such that $\nabla f(\tilde{x}^k)^T d^k < 0$. Repeating the same reasonings done in Proposition 3, we obtain (39), i.e.,

$$\nabla f(\tilde{x}^k)^T d^k \leq -\frac{1}{s} \|d^k\|^2. \quad (45)$$

First, we prove (43). From the Lipschitz continuity of $\nabla f(x)$, we can write

$$\begin{aligned} f(x^{k+1}) - f(\tilde{x}^k) &\leq \nabla f(\tilde{x}^k)^T (x^{k+1} - \tilde{x}^k) + \frac{L}{2} \|x^{k+1} - \tilde{x}^k\|^2 \\ &= \alpha^k \nabla f(\tilde{x}^k)^T d^k + \frac{L}{2} (\alpha^k)^2 \|d^k\|^2, \end{aligned}$$

Combining the above inequality with (45), we obtain

$$\begin{aligned} f(x^{k+1}) - f(\tilde{x}^k) &\leq \alpha^k \nabla f(\tilde{x}^k)^T d^k - \frac{sL}{2} (\alpha^k)^2 \nabla f(\tilde{x}^k)^T d^k \\ &= \alpha^k \left(1 - \frac{sL\alpha^k}{2}\right) \nabla f(\tilde{x}^k)^T d^k. \end{aligned}$$

Therefore, recalling that α is multiplied by δ when is decreased in the line search procedure (see step 3 of Algorithm 2), to satisfy the criterion within the Armijo line search and the feasibility we need to have $\alpha^k \geq R$, and then (43) holds.

Now, we prove (44). From the Armijo line search, we have $f(\tilde{x}^k) - f(x^{k+1}) \geq -\gamma \alpha^k \nabla f(\tilde{x}^k)^T d^k$. Recalling (45), we obtain

$$f(\tilde{x}^k) - f(x^{k+1}) \geq \frac{\gamma \alpha^k}{s} \|d^k\|^2.$$

Using the fact that

$$\|d^k\| = \|P(\tilde{x}^k - s \nabla f(\tilde{x}^k))_{\Delta_{N^k}} - \tilde{x}^k\| \geq \min\{1, s\} \|P(\tilde{x}^k - \nabla f(\tilde{x}^k))_{\Delta_{N^k}} - \tilde{x}^k\|$$

(see proof of Theorem 4.1 in [26] for the above inequality), we get

$$f(\tilde{x}^k) - f(x^{k+1}) \geq \frac{\gamma \alpha^k \min\{1, s\}^2}{s} \|P(\tilde{x}^k - \nabla f(\tilde{x}^k))_{\Delta_{N^k}} - \tilde{x}^k\|^2.$$

Combining this inequality with (43), we obtain that (44) holds. \square

Theorem 5. *Let $\{x^k\}$ be the sequence produced by Algorithm 1, where d^k is computed by (PG) rule and the Armijo line search is used. Let Assumption 1 and 2 hold, and let $f(x)$ be strictly convex on Δ_{N^+} .*

Then, there exists \bar{k} such that

$$f(x^{k+1}) - f(x^*) \leq (1 - \rho^{AS-PG}) [f(x^k) - f(x^*)], \quad \forall k \geq \bar{k},$$

with $\rho^{AS-PG} > 0$.

Proof. From Theorem 1, we have that $A^k \supseteq A^+$ and $N^k \subseteq N^+$ for sufficiently large k . Reasoning as in the first part of the proof of Theorem 4, we claim that there exists an iteration \bar{k} such that

- (i) $x_{A^+}^k = \tilde{x}_{A^+}^k = 0$, for all $k \geq \bar{k}$;
- (ii) $\nabla f(\tilde{x}^k)^T d^k < 0$, for all $k \geq \bar{k}$;
- (iii) $x^* \in \underset{x \in \Delta_{N^k}}{\text{Argmin}} f(x)$, for all $k \geq \bar{k}$.

Hence, by Theorem 2.1 in [26], for sufficiently large k we have

$$\|\tilde{x}^k - x^*\| \leq \tau \|P(\tilde{x}^k - \nabla f(\tilde{x}^k))_{\Delta_{N^k}} - \tilde{x}^k\|, \quad (46)$$

for some $\tau > 0$. Without loss of generality, we can assume that \bar{k} is sufficiently large to satisfy both the above inequality and the one of Lemma 1. Therefore, combining Lemma 1, (44) and (46), for $k \geq \hat{k}$ we can write

$$\begin{aligned} f(\tilde{x}^k) - f(x^*) &\leq \frac{L}{2} \|\tilde{x}^k - x^*\|^2 \\ &\leq \frac{L}{2} \tau^2 \|P(\tilde{x}^k - \nabla f(\tilde{x}^k))_{\Delta_{N^k}} - \tilde{x}^k\|^2 \\ &\leq \frac{sL\tau^2}{2\gamma R \min\{1, s\}^2} [f(\tilde{x}^k) - f(x^{k+1})]. \end{aligned}$$

Rearranging the terms and taking into account that $f(\tilde{x}^k) \leq f(x^k)$, we get for all $k \geq \bar{k}$

$$f(x^{k+1}) - f(x^*) \leq (1 - \rho^{\text{AS-PG}}) [f(\tilde{x}^k) - f(x^*)] \leq (1 - \rho^{\text{AS-PG}}) [f(x^k) - f(x^*)],$$

where $\rho^{\text{AS-PG}} = \frac{2\gamma R \min\{1, s\}^2}{sL\tau^2}$. □

6 Numerical Results

In this section, we report the numerical experience related to our active-set algorithmic framework. In the following, we denote by **FW**, **AFW** and **PG** the standard Frank-Wolfe, the standard away-step Frank-Wolfe and the standard Projected Gradient method, respectively. For any feasible point x , the search directions for these three algorithms are computed as follows:

- *standard Frank-Wolfe direction:*

$$d^{\text{FW}} = e_{\hat{i}} - x, \quad \hat{i} \in \underset{i=1, \dots, n}{\text{Argmin}} \{\nabla_i f(x)\};$$

- *standard away-step Frank-Wolfe direction:*

$$d^{\text{AFW}} = \begin{cases} d^{\text{FW}}, & \text{if } \nabla f(x)^T d^{\text{FW}} \leq \nabla f(x)^T d^{\text{A}}, \\ d^{\text{A}}, & \text{otherwise,} \end{cases}$$

where

$$d^{\text{A}} = x - e_{\hat{j}}, \quad \hat{j} \in \underset{j: x_j > 0}{\text{Argmax}} \{\nabla_j f(x)\};$$

- *standard projected gradient direction:*

$$d^{\text{PG}} = P(x - s\nabla f(x))_{\Delta} - x,$$

where $s > 0$ is a fixed scalar.

In the first set of experiments we considered a convex quadratic objective function and the comparisons also included the the **P2GP** algorithm [14], a gradient-based method specifically devised for solving quadratic problems with a single linear constraint and bounds on the variables.

In order to have a fair comparison between standard and active-set versions, the line search used for standard methods is the one described in Algorithm 2.

We further denote by **AS-FW**, **AS-AFW** and **AS-PG** the methods deriving from our algorithmic framework, where the search direction d^k is computed according to (FW), (AFW) and (PG) rule, respectively. The main aim of our numerical experience is showing how the use of our active-set strategy can improve the performance of existing global convergent algorithms.

In our experiments, we set $\delta = 0.5$ and $\gamma = 10^{-4}$ for the Armijo line search, and $s = 1$ for the computation of d^k when using (PG) rule and the standard projected gradient direction.

In order to calculate the active-set estimate at each iteration of the active-set methods, we need to set the ϵ parameter to a proper value, so that Assumption 1 is satisfied. In general, the value of this parameter cannot be a priori computed. Following [9, 12], we employ this simple updating rule: at every iteration k , we compute \tilde{x}^k and, if a sufficient decrease in the objective function is obtained (according to Theorem 1), we accept \tilde{x}^k and we do not change the value of ϵ . Otherwise, we do not accept \tilde{x}^k , we reduce ϵ and we estimate the active set again, continuing until we get a sufficient decrease in the objective function. The starting value for the ϵ parameter is 10^{-1} and we set $C = 10^{-6}$.

All the codes used in the tests were implemented in Matlab R2014b and the experiments were ran on an Intel Xeon(R), CPU E5-1650 v2 3.50 GHz.

6.1 Comparison on Instances from the Chebyshev Center Problem

The Chebyshev center problem consists in finding the circle of minimum radius that encloses all the points in a given finite set $C = \{c_1, \dots, c_n\} \subset \mathbb{R}^m$ (see, e.g., [32] and the references therein). The problem can be formulated as problem (1), where $f(x) = x^T A^T A x - \sum_{i=1}^n \|c_i\|^2 x_i$, with $A = (c_1 \quad \dots \quad c_n) \in \mathbb{R}^{m \times n}$. We generated instances with

- n (i.e., cardinality of C) = 2^{15} ;
- m (i.e., samples' dimension) = 10, 100, 1000.

For each combination of n and m , we considered 10 problems p_1, \dots, p_{10} where we randomly generated the vectors $c_i \in \mathbb{R}^m$, $i = 1, \dots, n$, by the Matlab's built-in function *rand*. For each problem p_h , $h = 1, \dots, 10$, we used the Matlab's command *rng(h)* to set the seed for the random number generator. For each problem we fix the starting point to e_1 (the same for all methods).

In this set of experiments we first run **AS-AFW**, stopping it at the first iteration k satisfying

$$\nabla f(x^k)^T (x - x^k) \geq -10^{-6}, \quad \forall x \in \Delta,$$

Table 1: Comparison on instances from the Chebyshev center problem. A star indicates that the time limit of 3600 seconds was reached.

m		CPU time						Obj							
		FW	AS-FW	AFW	AS-AFW	PG	AS-PG	P2GP	FW	AS-FW	AFW	AS-AFW	PG	AS-PG	P2GP
10	p_1	*	36.22	40.75	0.92	269.21	0.48	146.90	-33.49	-33.49	-33.49	-33.49	-33.49	-33.49	-33.49
	p_2	*	0.34	12.55	0.37	350.77	0.33	*	-35.78	-35.79	-35.79	-35.79	-35.79	-35.79	-35.77
	p_3	*	20.47	12.46	0.42	396.47	0.59	652.64	-36.58	-36.58	-36.58	-36.58	-36.58	-36.58	-36.58
	p_4	*	58.68	14.62	0.39	261.99	0.47	74.20	-34.92	-34.92	-34.92	-34.92	-34.92	-34.92	-34.92
	p_5	*	0.25	7.62	0.27	389.11	0.34	961.91	-37.56	-37.56	-37.56	-37.56	-37.56	-37.56	-37.56
	p_6	*	279.68	51.44	1.01	419.54	0.89	*	-35.96	-35.96	-35.96	-35.96	-35.96	-35.96	-35.96
	p_7	*	0.34	11.48	0.36	216.04	0.41	508.57	-36.01	-36.01	-36.01	-36.01	-36.01	-36.01	-36.01
	p_8	*	0.54	15.92	0.40	165.88	0.42	3127.75	-35.34	-35.34	-35.34	-35.34	-35.34	-35.34	-35.34
	p_9	*	16.87	14.63	0.38	336.34	0.35	100.14	-35.60	-35.60	-35.60	-35.60	-35.60	-35.60	-35.60
	p_{10}	*	36.90	25.33	0.47	251.71	0.46	*	-33.13	-33.13	-33.13	-33.13	-33.13	-33.13	-33.13
100	p_1	*	332.95	60.12	1.07	1089.13	1.60	22.51	-156.85	-156.86	-156.86	-156.86	-156.86	-156.86	-156.86
	p_2	*	326.44	51.96	1.64	878.55	2.04	31.94	-153.68	-153.68	-153.68	-153.68	-153.68	-153.68	-153.68
	p_3	*	74.47	43.54	0.83	137.13	2.45	18.70	-156.05	-156.05	-156.05	-156.05	-156.05	-156.05	-156.05
	p_4	*	33.49	50.20	1.16	848.55	0.96	22.97	-153.15	-153.16	-153.16	-153.16	-153.16	-153.16	-153.16
	p_5	*	6.93	43.55	0.97	920.43	2.76	26.49	-152.91	-152.91	-152.91	-152.91	-152.91	-152.91	-152.91
	p_6	*	326.46	46.81	1.06	908.36	1.39	17.83	-152.75	-152.75	-152.75	-152.75	-152.75	-152.75	-152.75
	p_7	*	6.68	45.52	1.07	983.68	0.85	*	-151.88	-151.88	-151.88	-151.88	-151.88	-151.88	-151.88
	p_8	*	317.54	48.29	1.05	1033.65	1.33	217.97	-153.43	-153.43	-153.43	-153.43	-153.43	-153.43	-153.43
	p_9	*	225.92	51.03	1.06	968.77	1.97	231.58	-152.26	-152.26	-152.26	-152.26	-152.26	-152.26	-152.26
	p_{10}	*	52.11	58.23	1.26	967.47	2.25	41.47	-153.34	-153.34	-153.34	-153.34	-153.34	-153.34	-153.34
1000	p_1	*	8.45	151.76	7.38	1770.16	3.76	29.18	-1131.67	-1131.67	-1131.67	-1131.67	-1131.67	-1131.67	-1131.67
	p_2	*	128.50	140.18	6.62	1374.36	5.30	13.65	-1134.48	-1134.49	-1134.49	-1134.49	-1134.49	-1134.49	-1134.49
	p_3	*	38.59	165.97	6.30	1114.03	3.43	13.14	-1136.53	-1136.54	-1136.54	-1136.54	-1136.54	-1136.54	-1136.54
	p_4	*	32.73	155.13	6.55	2009.08	2.98	12.77	-1137.88	-1137.88	-1137.88	-1137.88	-1137.88	-1137.88	-1137.88
	p_5	*	10.74	153.57	6.37	2284.88	4.88	11.11	-1138.01	-1138.01	-1138.01	-1138.01	-1138.01	-1138.01	-1138.01
	p_6	*	25.39	179.13	8.07	900.20	3.42	11.34	-1131.69	-1131.70	-1131.70	-1131.70	-1131.70	-1131.70	-1131.70
	p_7	*	13.82	123.64	6.83	1467.58	8.15	12.76	-1136.24	-1136.26	-1136.26	-1136.26	-1136.26	-1136.26	-1136.26
	p_8	*	65.76	136.15	5.35	2422.12	3.37	13.49	-1138.78	-1138.79	-1138.79	-1138.79	-1138.79	-1138.79	-1138.79
	p_9	*	542.44	161.34	6.63	969.59	13.65	11.90	-1136.86	-1136.87	-1136.87	-1136.87	-1136.87	-1136.87	-1136.87
	p_{10}	*	37.31	162.30	7.43	1644.57	18.22	118.83	-1135.62	-1135.62	-1135.62	-1135.62	-1135.62	-1135.62	-1135.62

and setting $f_{\min} = f(x^k)$. Then, we used f_{\min} as target value to stop the other algorithms using a certain tolerance (keep in mind that the problem is convex). Namely, all the other algorithms, including P2GP, were stopped at the first iteration k satisfying

$$f(x^k) \leq f_{\min} + 10^{-6}(1 + |f_{\min}|).$$

We also used a time limit of 3600 seconds for each method. In P2GP, all parameters were set to their default values (except those for termination, that were set in order to stop the algorithm as just described).

In Table 1, for each instance and each algorithm, we report the CPU time needed to satisfy the stopping criterion and the final objective function value found.

We first see that the active-set versions are up to two orders of magnitude faster than the standard counterparts. More in detail, in Figure 1 we report the optimization error for the comparison between AS-FW and FW, between AS-AFW and AFW, and between AS-PG and PG. For every m , in each plot we report the optimization error $E^k = f(x^k) - f_{\min}$, averaged over the 10 runs, versus the computational time. We can notice that the active-set methods clearly outperform the standard algorithms.

For what concerns the comparisons among our active-set methods and P2GP, from Table 1 we see that both AS-AFW and AS-PG are much faster than P2GP for all the considered values of m , while AS-FW is faster than P2GP for $m = 10$, slower for $m = 1000$ and is comparable with P2GP for $m = 100$ (in the sense that on average they take an amount of time of the

same order of magnitude). For these comparisons, the optimization error $E^k = f(x^k) - f_{\min}$, averaged over the 10 runs, is reported in Figure 2.

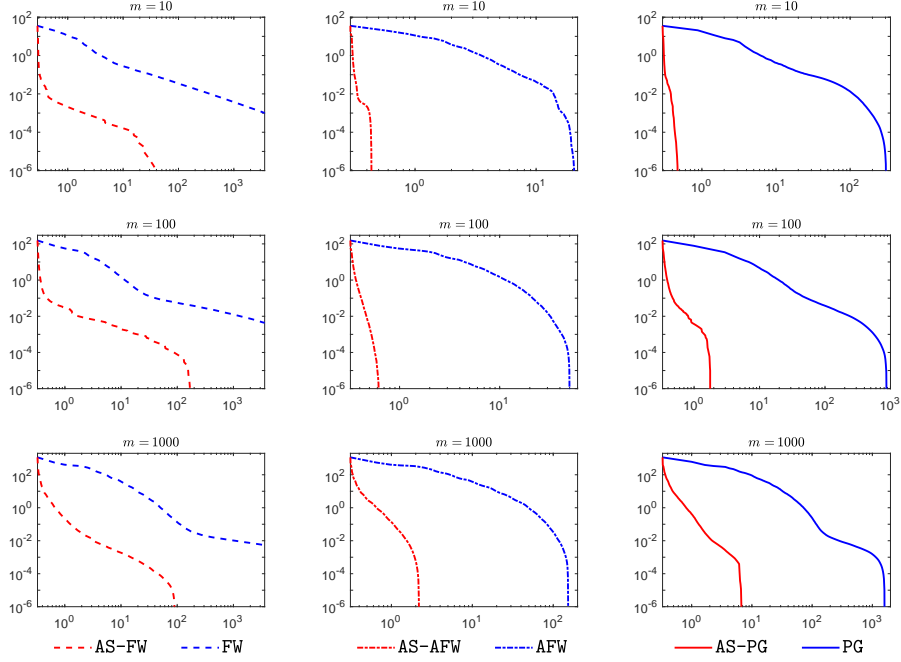


Figure 1: Objective function error (y axis) vs CPU time in seconds (x axis). Comparison between original and active-set algorithms on instances from the Chebyshev center problem. Both y axis and x axis are in logarithmic scale.

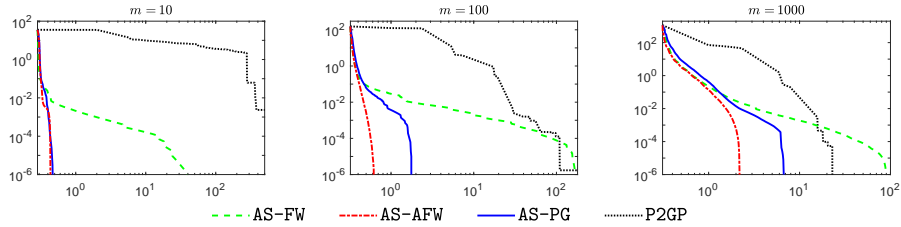


Figure 2: Objective function error (y axis) vs CPU time in seconds (x axis). Comparison among active-set algorithms and P2GP on instances from the Chebyshev center problem. Both y axis and x axis are in logarithmic scale.

6.2 Comparison on Instances from the Symmetric Eigenvalue Complementarity Problem

In our experiments, we further consider a class of non-convex non-quadratic problems, namely instances from the symmetric Eigenvalue Complementarity Problem (EiCP). Given the symmetric matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$, $B \succ 0$, the symmetric EiCP consists of finding a

scalar λ and a vector $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$\begin{aligned} w &= (\lambda B - A)x, \\ w &\geq 0, \quad x \geq 0, \\ w^T x &= 0. \end{aligned}$$

It can be proven that the symmetric EiCP reduces to computing a stationary point of problem (1) where $f(x) = -\frac{x^T Ax}{x^T Bx}$ [23].

Taking into account the numerical experience reported in [23], we considered instances with B equal to the identity matrix and $A \in \mathbb{R}^{n \times n}$ a negative definite matrix. More in detail, we considered 10 problems p_1, \dots, p_{10} where $-A$ was generated as proposed in [27], with $n = 2^{15}$. Starting from a vector $y \in \mathbb{R}^n$ with components randomly uniformly distributed in $[-1, 1]$ (obtained by the Matlab's built-in function *rand*), we set the matrix Y as

$$Y = I - 2 \frac{yy^T}{\|y\|^2},$$

where I is the identity matrix and, for each problem p_h , $h = 1, \dots, 10$, we used the Matlab's command *rng(h)* to set the seed for the random number generator. The matrix A was then generated as $-YDY$, where D is a diagonal matrix with entries $D_{ii} = e^{\frac{i-1}{n-1}}$, $i = 1, \dots, n$. For each problem, we randomly generated a feasible starting point (the same for all methods).

All algorithms were stopped at the first iteration k satisfying

$$\nabla f(x^k)^T(x - x^k) \geq -10^{-4}, \quad \forall x \in \Delta.$$

We also considered a time limit of 3600 seconds (as for the experiments of the previous subsection), but it was never reached by any considered method.

In Table 2, for each instance and each algorithm, we report the CPU time needed to satisfy the stopping criterion and the final objective function value found. In Figure 3, we report the optimization error for the comparison between AS-FW and FW, between AS-AFW and AFW, and between AS-PG and PG. In particular, in each plot we report the optimization error $E^k = f(x^k) - f_{\min}$, averaged over the 10 runs, versus the computational time, where f_{\min} is the smallest objective value found in any comparison between the two considered algorithms.

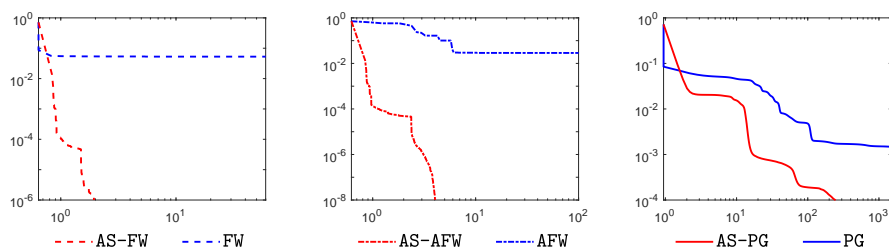
Also for these problems, we can notice that the active-set framework clearly outperforms on average the original algorithms used in the comparison. In particular, the active-set versions are on average able to find a better solution in a smaller amount of time.

7 Conclusions

In this paper, we focused on minimization problems over the simplex and described an active-set algorithmic framework. The active-set strategy we adopted here does not only focus on the zero variables and keep them fixed, but rather tries to quickly identify as many active variables as possible (including nonzero variables) at a given point. Furthermore, it suitably reduces the objective function (when setting to zero those variables estimated active), while guaranteeing feasibility. This last feature, together with the use of active-set

Table 2: Comparison on instances from the symmetric Eigenvalue Complementarity Problem.

	CPU time						Obj					
	FW	AS-FW	AFW	AS-AFW	PG	AS-PG	FW	AS-FW	AFW	AS-AFW	PG	AS-PG
p_1	56.57	1.62	35.73	1.15	3148.11	40.47	1.00	1.00	1.00	1.00	1.00	1.00
p_2	31.41	1.15	140.99	2.77	258.68	177.18	1.00	1.00	1.00	1.00	1.00	1.00
p_3	265.40	11.85	345.04	15.75	1602.88	1557.03	1.00	1.00	1.00	1.00	1.00	1.00
p_4	0.66	1.01	52.39	1.21	2056.99	149.71	1.24	1.00	1.00	1.00	1.00	1.00
p_5	0.69	0.82	5.13	0.81	240.35	254.33	1.07	1.07	1.07	1.07	1.00	1.00
p_6	0.63	1.13	0.97	1.72	1531.66	654.37	1.11	1.00	1.11	1.00	1.00	1.00
p_7	3.08	0.83	6.91	0.80	1224.86	622.67	1.00	1.00	1.00	1.00	1.00	1.00
p_8	51.59	1.35	121.98	2.17	835.16	35.57	1.00	1.00	1.00	1.00	1.00	1.00
p_9	217.15	4.86	376.64	7.86	3444.62	41.61	1.00	1.00	1.00	1.00	1.00	1.00
p_{10}	0.66	2.72	3.67	6.51	5.07	135.14	1.18	1.00	1.18	1.00	1.01	1.00

Figure 3: Objective function error (y axis) vs CPU time in seconds (x axis). Comparison between original and active-set algorithms on instances from the symmetric eigenvalue complementarity problem. Both y axis and x axis are in logarithmic scale.

gradient related directions and an Armijo line search, allowed us to prove global convergence of the framework. We further described three different types of active-set gradient related directions and proved linear convergence rate when using those directions in the algorithm. Our numerical experience on sparse optimization problems highlighted the efficiency of our new method when dealing with both non-convex and convex instances.

References

- [1] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20(2):221–246, 1982.
- [2] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 1999.
- [3] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, 23(1):101–125, 2002.
- [4] I. M. Bomze. Evolution towards the maximum clique. *Journal of Global Optimization*, 10(2):143–164, 1997.
- [5] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of combinatorial optimization*, pages 1–74. Springer, 1999.

- [6] C. P. Brás, A. Fischer, J. J. Júdice, K. Schönefeld, and S. Seifert. A block active set algorithm with spectral choice line search for the symmetric eigenvalue complementarity problem. *Applied Mathematics and Computation*, 294:36–48, 2017.
- [7] C. Buchheim, M. De Santis, S. Lucidi, F. Rinaldi, and L. Trieu. A feasible active set method with reoptimization for convex quadratic mixed-integer programming. *SIAM Journal on Optimization*, 26(3):1695–1714, 2016.
- [8] K. L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms*, 6(4):63, 2010.
- [9] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. A Two-Stage Active-Set Algorithm for Bound-Constrained Optimization. *Journal of Optimization Theory and Applications*, 172(2):369–401, 2017.
- [10] E. de Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2):111–125, 2008.
- [11] M. De Santis, G. Di Pillo, and S. Lucidi. An active set feasible method for large-scale minimization problems with bound constraints. *Computational Optimization and Applications*, 53(2):395–423, 2012.
- [12] M. De Santis, S. Lucidi, and F. Rinaldi. A Fast Active Set Block Coordinate Descent Algorithm for ℓ_1 -Regularized Least Squares. *SIAM Journal on Optimization*, 26(1):781–809, 2016.
- [13] G. Di Pillo and L. Grippo. A class of continuously differentiable exact penalty function algorithms for nonlinear programming problems. In *System Modelling and Optimization*, pages 246–256. Springer, 1984.
- [14] D. di Serafino, G. Toraldo, M. Viola, and J. Barlow. A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables. *SIAM Journal on Optimization*, 28(4):2809–2838, 2018.
- [15] F. Facchinei and S. Lucidi. Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. *Journal of Optimization Theory and Applications*, 85(2):265–289, 1995.
- [16] F. Facchinei, A. Fischer, and C. Kanzow. On the accurate identification of active constraints. *SIAM Journal on Optimization*, 9(1):14–32, 1998.
- [17] F. Facchinei, J. Júdice, and J. Soares. An active set Newton algorithm for large-scale nonlinear programs with box constraints. *SIAM Journal on Optimization*, 8(1):158–186, 1998.
- [18] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.
- [19] J. Guélat and P. Marcotte. Some comments on Wolfe’s ‘away step’. *Mathematical Programming*, 35(1):110–119, 1986.
- [20] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM Journal on Optimization*, 17(2):526–557, 2006.
- [21] W. W. Hager and H. Zhang. An active set algorithm for nonlinear optimization with polyhedral constraints. *Science China Mathematics*, 59(8):1525–1542, 2016.
- [22] W. W. Hager and H. Zhang. Projection onto a polyhedron that exploits sparsity. *SIAM Journal on Optimization*, 26(3):1773–1798, 2016.
- [23] A. N. Iusem, J. J. Júdice, V. Sessa, and P. Sarabando. Splitting methods for the

- eigenvalue complementarity problem. *Optimization Methods and Software*, 34(6):1184–1212, 2019.
- [24] S. Lacoste-Julien and M. Jaggi. An affine invariant linear convergence analysis for Frank-Wolfe algorithms. *arXiv preprint arXiv:1312.7864*, 2013.
- [25] S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *NIPS 2015 - Advances in Neural Information Processing Systems*, 2015.
- [26] Z.-Q. Luo and P. Tseng. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2):408–425, 1992.
- [27] J. J. Moré and G. Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerische Mathematik*, 55(4):377–400, 1989.
- [28] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
- [29] J. Nocedal and S. J. Wright. *Numerical optimization*, 2006.
- [30] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.
- [31] P. Wolfe. Convergence theory in nonlinear programming. *Integer and nonlinear programming*, pages 1–36, 1970.
- [32] S. Xu, R. M. Freund, and J. Sun. Solution methodologies for the smallest enclosing circle problem. *Computational Optimization and Applications*, 25(1-3):283–292, 2003.