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, Università di Padova, Via Trieste, 63, 35121 Padova, 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 Roma, 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 function over the simplex. The method, at each iteration, makes use of a rule for identifying active variables (i.e., variables that are zero at a stationary point) and a specific class of directions (so-called active-set gradient related directions) satisfying a new “nonorthogonality” type of condition that well suits to our needs. We prove convergence when using an Armijo line search in the given framework. We further describe three different active-set gradient related directions guaranteeing linear convergence of our framework (under suitable assumptions). Finally, we report numerical experiments showing the effectiveness of the approach.

Keywords. Active-set methods, Unit simplex

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 ℓp, mixture density estimation, finding maximum stable sets (maximum cliques) in graphs, portfolio optimization and population dynamics problems (see, e.g., [3, 6, 8] and references therein).

The problem we address can be stated as follows:

\[ \min_{x \in \Delta} f(x) \]  

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, \( f : \mathbb{R}^n \to \mathbb{R} \) 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 \( V = [v_1 \ldots v_m] \in \mathbb{R}^{n \times m} \), with \( v_1, \ldots, v_m \) vertices of \( P \), 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., [22] 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 (see, e.g., [1, 4, 7, 12, 16, 18, 17] and references therein) are based on the idea of combining an “identification” step (i.e., a step used to identify the variables that are active at the solution), with a minimization step in a reduced space (i.e., a minimization step performed in the space obtained by keeping the estimated active variables fixed).

Here, we propose an active-set algorithmic framework for solving problem (1), where \( f(x) \) is a possibly non-convex objective function. In the first part of the paper, we describe an active-set estimate to identify the set of variables that are zero at a stationary point of problem (1) by adapting some specific strategies proposed in the contexts of box-constrained problems (see [3, 7, 9, 10] and references therein) 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).

The second property, which is somehow related to the fact that estimated active variables satisfy an approximate optimality condition, enables us to easily use this strategy into a globally convergent algorithm.

In the second part of the paper, inspired by the “nonorthogonality” type of condition described in [2], we define a new class of directions, the so-called active-set gradient related directions, 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 [14], away-step Frank-Wolfe [24] 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 $I \subseteq \{1, \ldots, n\}$, 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 $S$. The open ball with center $x$ and radius $\rho > 0$ is denoted by $B(x, \rho)$. Finally, given a subset $I \subseteq \{1, \ldots, n\}$, we denote $\Delta_I$ the subset of points of $\Delta$ with $x_i = 0$ for all $i \in I$, i.e.,

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

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

\[
\nabla f(x^*) - \lambda^* e - \mu^* = 0, \\
(\mu^*)^T x^* = 0, \\
\mu^* \geq 0.
\]

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:

\[
\nabla_i f(x^*) \begin{cases} 
\geq \lambda^*, & x_i^* = 0, \\
= \lambda^*, & x_i^* > 0, 
\end{cases} \quad i = 1, \ldots, n.
\]
Remark 1. Note that at a stationary point \( x^* \), we have \( \nabla f(x^*)^T(e_i - x^*) \geq 0 \) for all \( i = 1, \ldots, n \), so that
\[
\max\{0, -\nabla f(x^*)^T(e_i - x^*)\} = 0, \quad i = 1, \ldots, n.
\]

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 \( x^* \). 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:
\[
\bar{A}(x^*) = \{i \in \{1, \ldots, n\} : x^*_i = 0\}.
\]

We further define the nonactive set \( \bar{N}(x^*) \) as the complement of \( \bar{A}(x^*) \):
\[
\bar{N}(x^*) = \{1, \ldots, n\} \setminus \bar{A}(x^*) = \{i \in \{1, \ldots, n\} : x_i^* > 0\}.
\]

Now, we describe how, at any feasible point \( x \), we estimate the active set. Following the approach proposed in [11, 13], 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 \((\lambda^*, \mu^*)\) be the KKT multipliers associated to a given stationary point \( x^* \). By (2), we have
\[
\mu^* = \nabla f(x^*)^T x^* - \lambda^* e,
\]
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:
\[
\lambda(x) = \nabla f(x)^T x, \quad (7)
\]
\[
\mu_i(x) = \nabla_i f(x) - \lambda(x), \quad i = 1, \ldots, n. \quad (8)
\]

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
\[
A(x) = \{i : x_i \leq \epsilon \mu_i(x)\} = \{i : x_i \leq \epsilon \nabla f(x)^T (e_i - x)\},
\]
\[
N(x) = \{i : x_i > \epsilon \mu_i(x)\} = \{i : x_i > \epsilon \nabla f(x)^T (e_i - x)\},
\]
where \( \epsilon \) is a positive scalar.
By adapting Theorem 2.1 in [13], 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 2.** If \((x^*, \lambda^*, \mu^*)\) satisfies KKT conditions for problem (1), then there exists a neighborhood \( B(x^*, \rho) \) such that, for each \( x \) in this neighborhood, 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 \( x \in B(x^*, \rho) \).

### 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 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 \( J(x) \) be the set:

\[
J(x) = \left\{ j: \ j \in \text{Argmin}_{i=1, \ldots, n}\{\nabla_i f(x)\} \right\}.
\]

Let \( x \in \mathbb{R}^n \) be a feasible non-stationary point of problem (1). 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 definition of multipliers (7)–(8), we obtain

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

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 3.** 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 \( \epsilon \) appearing in Definition 3.
**Assumption 4.** Assume that the parameter $\epsilon$ 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 4 hold. Given a feasible non-stationary point $x$ of problem (1), let $j \in N(x) \cap J(x)$ and $I = \{1, \ldots, n\} \setminus \{j\}$. Let $\hat{A}(x)$ be a set of indices such that $\hat{A}(x) \subseteq A(x)$. Let $\hat{x}$ 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 4.

**Proof.** Define

$$\hat{A}^+ = \hat{A}(x) \cap \{i: x_i > 0\}.$$  \hspace{1cm} (12)

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

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

and, by adding and removing $CL\|\tilde{x} - x\|^2$, we get

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

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

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

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

$$\|\tilde{x} - x\|^2 = \sum_{i \in \hat{A}^+} (x_i)^2 + \left(\sum_{i \in \hat{A}^+} x_i\right)^2 \leq \sum_{i \in \hat{A}^+} (x_i)^2 + |\hat{A}^+| \sum_{i \in \hat{A}^+} (x_i)^2 = (|\hat{A}^+| + 1)x_{\hat{A}^+}^Tx_{\hat{A}^+}$$ \hspace{1cm} (15)

and

$$\nabla f(x)^T(\tilde{x} - x) = -\nabla_{\hat{A}^+}f(x)^Tx_{\hat{A}^+} + \nabla_jf(x)\sum_{i \in \hat{A}^+} x_i = x_{\hat{A}^+}^T \left(\nabla_jf(x)e_{\hat{A}^+} - \nabla_{\hat{A}^+}f(x)\right).$$ \hspace{1cm} (16)

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

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

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

(18)

From (16) and (18), we get

\[ \nabla f(x)^T(\tilde{x} - x) + \frac{L(2C + 1)}{2} \|\tilde{x} - x\|^2 \leq x_{\hat{A}^+}^T \left[ \nabla_j f(x)e_{\hat{A}^+} - \nabla f(x) \right] + \]

\[ + \frac{L(2C + 1)}{2} (|\hat{A}^+| + 1) \epsilon x_{\hat{A}^+}^T \left( \nabla_{\hat{A}^+} f(x) - \nabla_j f(x)e_{\hat{A}^+} \right) \]

\[ = \left( \frac{L(2C + 1)}{2} (|\hat{A}^+| + 1) \epsilon - 1 \right) x_{\hat{A}^+}^T \left( \nabla_{\hat{A}^+} f(x) - \nabla_j f(x)e_{\hat{A}^+} \right) \]

\[ \leq \left( \frac{L(2C + 1)}{2} n \epsilon - 1 \right) x_{\hat{A}^+}^T \left( \nabla_{\hat{A}^+} f(x) - \nabla_j f(x)e_{\hat{A}^+} \right), \]

where the last inequality follows from the non-negativity of \( x_{\hat{A}^+}^T \left( \nabla_{\hat{A}^+} f(x) - \nabla_j f(x)e_{\hat{A}^+} \right) \)

(impied by (18)) and from the fact that \(|\hat{A}^+| + 1 \leq n\) (impied by Proposition 1). Then, (14) follows from the assumption we made on \( \epsilon \).

Remark 5. In Assumption 4 the upper bound of \( \epsilon \) depends on \( n \). Actually, Proposition 2 still holds by replacing the constant \( n \) by \(|\hat{A}^+| + 1 \) in the upper bound of \( \epsilon \), 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. Not that, in general, \( \hat{A}^+ \) might be considerably smaller than \( n \), but it depends on both the specific point \( x \) and \( \epsilon \) itself. So, for the sake of simplicity, in Assumption 4 we use the constant \( n \), even if all the theoretical results of the paper would hold by using \(|\hat{A}^+| + 1 \) instead.

Remark 6. From Assumption 4 and Theorem 2 we see that there is a trade-off, depending on the constant \( C \), between the magnitude of the upper bound of \( \epsilon \) and the decrease in the objective function guaranteed by Theorem 2. Namely, for small values of \( C \), large values of \( \epsilon \) 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 [11, 13], 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 described in the previous section. The framework performs two different steps at each iteration: the first one for updating the estimated active variables, and the second one for updating 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 2; second, the ability to get a decrease in the objective function when moving the variables as indicated in 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 \( \tilde{x}^k \) by setting \( \tilde{x}_{A(x^k)} \) to zero and updating \( \tilde{x}_j \), with \( j \in J(x^k) \) (all the other variables stay the same). Then, in the second step, we compute a search direction \( d^k_{N(x^k)} \) in the subspace of the estimated nonactive variables, and we eventually perform a line search to get a new iterate \( x^{k+1} \).

From now on, given any feasible point \( x^k \) generated by the algorithm and a feasible direction \( d^k \), we call \( \alpha^k_{\text{max}} \) the maximum stepsize that can be taken 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:

**Definition 4.** The sequence of directions \( \{d^k\} \) is active-set gradient related if, for any subsequence \( \{x^k\}_K \) such that \( N(x^k) = \hat{N} \) for all \( k \in K \) and \( \lim_{k \to \infty, k \in K} x^k = x^* \), where \( x^* \) is non-stationary in \( \Delta_{\hat{N}} \), we have that

\[
\{d^k_{N}\}_K \text{ is bounded,}
\]

\[
\limsup_{k \to \infty, k \in K} \nabla_{N} f(\tilde{x}^k)^T d^k_{N} < 0,
\]  

(19)

\[
\liminf_{k \to \infty, k \in K} \alpha^k_{\text{max}} \geq M > 0.
\]  

(20)

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

A possibility for the computation of the stepsize, at Step 9 of Algorithm 1, 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.

4.1 Global Convergence Analysis.

In this section, we show the global convergence of \texttt{AS-SIMPLEX} to stationary points. First, we need an intermediate result.
Algorithm 1 Active-Set algorithmic framework for minimization over the simplex (AS-SIMPLEX)

1. Choose a feasible point $x^0$
2. For $k = 0, 1,\ldots$
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 $j \in N^k \cap J^k$ and define $\hat{N}^k = N^k \setminus \{j\}$
6. Set $\bar{a}^k_{i} = 0$, $\bar{x}^k_{\hat{N}^k} = x^k_{\hat{N}^k}$ and $\tilde{x}^k_j = x^k_j + \sum_{h \in A^k} x^k_h$
7. Compute a feasible direction $d^k$ satisfying Definition 4, such that $d^k_{A^k} = 0$
8. If $\nabla f(\tilde{x}^k) \cdot d^k < 0$ then
9. Compute a stepsize $\alpha^k \in (0, \alpha_{\text{max}}]$ 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

Algorithm 2 Armijo line search

0. Choose $\delta \in (0, 1)$, $\gamma \in (0, 1)$
1. Set initial stepsize $\alpha = \alpha_{\text{max}}$
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

Lemma 1. Let Assumption 4 hold. Let $\{x^k\}$ be the sequence of points produced by AS-SIMPLEX, where the stepsize $\alpha^k$ is computed using the Armijo line search. Then,

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

$$\lim_{k \to \infty} \|\tilde{x}^k - x^k\| = 0.$$  \hspace{1cm} (23)

Proof. Proof. From the instructions of the algorithm, we can write

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

From the continuity of the objective function and the compactness of the feasible set, it follows that (22) holds. Then, using again the above relation, we have that also (23) holds.

Theorem 7. Let Assumption 2 hold. Let $\{x^k\}$ be the sequence of points produced by AS-SIMPLEX, where the stepsize $\alpha^k$ is computed using the Armijo line search. Then, either an integer $\bar{k} \geq 0$ exists such that $x^k$ 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. 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. Since the feasible set is
compact, then the sequence \( \{x^k\} \) attains a limit point \( x^* \) and, recalling (23) of Lemma 1, there exists \( K \subseteq \mathbb{N} \) such that
\[
\lim_{k \to \infty, k \in K} x^k = \lim_{k \to \infty, k \in K} \tilde{x}^k = x^*.
\] (24)

Taking into account the structure of the feasible set, we can characterize a stationary point \( x \) using the following condition (see Remark 1):
\[
\nabla f(x)^T (e_i - x) \geq 0, \quad \forall \ i \in \{1, \ldots, n\}.
\]

Let \( \Phi_i(x) \) be the continuous function defined as
\[
\Phi_i(x) = \max\{0, -\nabla f(x)^T (e_i - x)\}, \quad i = 1, \ldots, n,
\]
that measures the violation of the stationarity conditions for a variable \( x_i, i = 1, \ldots, n \).

By contradiction, we assume that \( x^* \) is non-stationary, so that an index \( \nu \in \{1, \ldots, n\} \) exists such that
\[
\Phi_\nu(x^*) > 0.
\] (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
\[
0 \leq x^k_\nu \leq \epsilon \nabla f(x^k)^T (e_\nu - x^k),
\]
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, \( \nu \) 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 \( d^k_A = 0 \), we also have that \( \{d^k\}_K \) is bounded. Then, there exists a further subsequence (that we rename \( K \) again without loss of generality) such that
\[
\lim_{k \to \infty, k \in K} d^k = \bar{d}.
\] (26)

Moreover, exploiting again Definition 4 we have that \( \eta > 0 \) and \( M > 0 \) exist such that
\[
\limsup_{k \to \infty, k \in K} \nabla f(\tilde{x}^k)^T d^k = -\eta,
\] (27)
\[
\alpha^k_{\text{max}} \geq M > 0, \quad k \text{ sufficiently large, } k \in K.
\] (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 Step 9 of Algorithm 1, the Armijo line search computes a value \( \alpha^k \in (0, \alpha^k_{\text{max}}] \) in a finite number of iterations for \( k \geq \hat{k} \), such that
\[
f(x^{k+1}) \leq f(x^k) + \gamma \alpha^k \nabla f(\tilde{x}^k)^T d^k, \quad \forall k \geq \hat{k}, k \in K,
\] (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) of Lemma 1, we get that the left-hand side of the above inequality converges to zero for $k \to \infty$, hence
\[
\lim_{k \to \infty, k \in K} \alpha^k |\nabla f(\tilde{x}^k)^T d^k| = 0.
\] (30)

Using (27), we obtain that $\lim_{k \to \infty, k \in K} \alpha^k = 0$. Taking into account (28), it follows that there exists $\bar{k} \in K$, $\bar{k} \geq \hat{k}$, such that
\[
\alpha^k < \alpha^k_{max}, \quad \forall k \geq \bar{k}, k \in K.
\]

In other words, for $k \geq \bar{k}$, $k \in K$, the stepsize $\alpha^k$ cannot be set equal to the maximum stepsize and, taking into account the line search procedure, we can write
\[
f(\tilde{x}^k + \alpha^k \frac{\delta}{\delta} d^k) > f(\tilde{x}^k) + \gamma \frac{\alpha^k}{\delta} \nabla f(\tilde{x}^k)^T d^k, \quad \forall k \geq \bar{k}, k \in K.
\] (31)

We can apply the mean value theorem and we have that $\xi_k \in (0,1)$ exists such that
\[
f(\tilde{x}^k + \frac{\alpha^k}{\delta} d^k) = f(\tilde{x}^k) + \frac{\alpha^k}{\delta} \nabla f(\tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k)^T d^k, \quad \forall k \geq \bar{k}, k \in K.
\] (32)

By substituting (32) within (31), we have
\[
\nabla f(\tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k)^T d^k > \gamma \nabla f(\tilde{x}^k)^T d^k, \quad \forall k \geq \bar{k}, k \in K.
\] (33)

From (24), and exploiting the fact that $\{\xi_k\}$, $\{\alpha^k\}$ and $\{d^k\}$ are bounded, we get
\[
\lim_{k \to \infty, k \in K} \tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k = \lim_{k \to \infty, k \in K} \tilde{x}^k = x^*.
\]

Therefore, taking the limits in (31) and (32), and taking into account (26), we obtain that $\nabla f(x^*)^T d \geq \gamma \nabla f(x^*)^T d$, or equivalently,
\[
(1 - \gamma) \nabla f(x^*)^T d \geq 0.
\]

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

**Remark 8.** Theorem 7 holds when using as stepsize in AS-SIMPLEX any value $\alpha^k \in (0, \alpha^k_{max}]$ such that
\[
f(\tilde{x}^k + \alpha^k d^k) \leq f(\tilde{x}^k + \alpha^k_A d^k),
\]
where $\alpha^k_A$ 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 7 holds under the assumption that the stepsize is computed in AS-SIMPLEX by means of an exact line search, that is, $\alpha^k$ is computed as
\[
\alpha^k \in \text{Argmin}_{\alpha \in (0, \alpha^k_{max}]} f(\tilde{x}^k + \alpha d^k).
\]
4.2 Active-Set Gradient Related Directions in AS-SIMPLEX.

At every iteration \( k \) of Algorithm 1 we need to compute an active-set gradient related \( d^k \) in \( \tilde{x}^k \), according to Definition 4 such that \( d^k_{A_k} = 0 \) and \( d^k_N \) satisfies (19)–(21).

As examples of \( d^k_N \), we consider Frank-Wolfe-type and projected gradient directions:

(FW) Frank-Wolfe direction:
\[
d^k_N = (e_i - \tilde{x}_i)^N, \quad i \in \text{Argmin}_{i \in N} \{ \nabla_i f(\tilde{x}^k) \} ;
\]  

(AFW) away-step Frank-Wolfe direction:
\[
d^k_N = d^k_{AFW} = \begin{cases} 
\hat{d}^k_{AFW}, & \text{if } \nabla_N f(\tilde{x}^k)^T \hat{d}^k_{AFW} \leq \nabla_N f(\tilde{x}^k)^T d^k_N, \\
\hat{d}^k_A, & \text{otherwise},
\end{cases}
\]

where
\[
d^k_A = (\tilde{x}^k - e_j)^N, \quad j \in \text{Argmax}_{j \in N^0} \{ \nabla_j f(\tilde{x}^k) \}
\]  

and \( N^0 = \{ j \in N^k : \tilde{x}_j^k > 0 \} \).

(PG) projected gradient direction:
\[
d^k_N = d^k_{PG} = \left( P(\tilde{x}^k - s\nabla f(\tilde{x}^k))\Delta_N - \tilde{x}^k \right)^N,
\]

where \( s > 0 \) is a fixed scalar.

In the following, we will refer to \( d^{FW}, d^{AFW} \) and \( d^{PG} \) when the subdirection \( d^k_N \) 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 \( d^k_N \), the resulting sequence of directions \( \{d^k\} \) generated by AS-SIMPLEX is active-set gradient related, i.e., it satisfies Definition 4.

**Proof.** Since \( \Delta \) 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 \( \lim_{k \to \infty, k \in K} x^k = x^*_N \) and \( N(x^k) = \hat{N} \) for all \( k \in K \), where \( x^* \) is non-stationary in \( \Delta_N \). We consider the different cases:

(FW) By definition of the index \( i \) given in (34), it is easy to see that, for all \( k \geq 0 \), we have \( \nabla_i f(\tilde{x}^k) \leq \nabla_N f(\tilde{x}^k)^T x_N, \forall x \in \Delta_N \). Thus,
\[
\nabla_N f(\tilde{x}^k)^T d^k_N = \nabla_N f(\tilde{x}^k)^T (e_i - \tilde{x}_i)^N \leq \nabla_N f(\tilde{x}^k)^T (x - \tilde{x}^k)^N = \nabla f(\tilde{x}^k)^T (x - \tilde{x}^k), \quad \forall x \in \Delta_N,
\]
where the last equality follows from the fact that \( x_{A^k} = 0 \) for all \( x \in \Delta_N \) and \( \tilde{x}_{A^k} = 0 \). Passing to the limit, we obtain

\[
\limsup_{k \to \infty, k \in K} \nabla_{\hat{N}} f(\tilde{x}^k)^T d^k_{\hat{N}} \leq \nabla f(x^*)^T (x - x^*), \quad \forall x \in \Delta_{\hat{N}}.
\]

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

(AFW) Taking into account that, 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 denote with \( \tilde{K} \) for simplicity) exists such that

\[
\lim_{k \to \infty, k \in \tilde{K}} \alpha_{\text{max}}^k = 0.
\]

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

\[
d^k_{\hat{N}} = A_{\hat{N}}^k, \quad \forall k \in \tilde{K}.
\]

We have that \( \alpha_{\text{max}}^k = \frac{x^k_{\tilde{j}}}{1 - x^k_{\tilde{j}}} \), for all \( k \in \tilde{K} \), where \( \tilde{j} \) is the index computed according to (35). Since the number of indices in \( \hat{N} \) is finite, we can consider a further subsequence (that we denote with \( \tilde{K} \) for simplicity), where the index \( \tilde{j} \) is fixed. Taking into account (35), it is easy to see that \( \{\tilde{x}^k_{\tilde{j}}\}_{\tilde{K}} \to 0 \). Using (23), we get

\[
\lim_{k \to \infty, k \in \tilde{K}} x^k_{\tilde{j}} = 0.
\]

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

\[
\limsup_{k \to \infty, k \in K} \nabla f(x^k)^T d^k_{\hat{N}} = -\eta.
\]

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

\[
-\eta = \limsup_{k \to \infty, k \in K} \nabla f(x^k)^T d^k
= \limsup_{k \to \infty, k \in K} [\nabla f(x^k)^T (\tilde{x}^k - x^k) + \nabla f(x^k)^T (x^k - e_{\tilde{j}})]
\]

\[
= \limsup_{k \to \infty, k \in K} \nabla f(x^k)^T (x^k - e_{\tilde{j}}),
\]

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

\[
\nabla f(x^k)^T (x^k - e_{\tilde{j}}) \leq -\frac{\eta}{2},
\]

\[
x^k_{\tilde{j}} \leq \frac{\eta}{2}.
\]
Therefore,
\[ x_j^k \leq \epsilon \nabla f(x^k)^T(e_j - x^k), \quad \forall k \geq \tilde{k}, \ k \in \hat{K}. \]
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 \( \hat{x}^k = P(\tilde{x}^k - s \nabla f(\tilde{x}^k))\Delta_{\hat{N}_k} \), so that \( d^k = \hat{x}^k - \tilde{x}^k \). By continuity of the projection operator, we get
\[
\lim_{k \to \infty, k \in K} \hat{x}^k = P(x^* - s \nabla f(x^*))\Delta_{\hat{N}}.
\]
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
\[
\nabla f(\tilde{x}^k)^T d^k = \nabla f(\tilde{x}^k)^T (\tilde{x} - \tilde{x}^k) \leq -\frac{1}{s} \| \tilde{x}^k - \tilde{x} \|^2 = -\frac{1}{s} \| d^k \|^2, \quad \forall k \geq 0. \tag{39}
\]
Since \( d^k \Delta_{Ak} = 0 \) for all \( k \geq 0 \), taking the limit we have
\[
\limsup_{k \to \infty, k \in K} \nabla_{\hat{N}} f(\tilde{x}^k)^T d^k = \limsup_{k \to \infty, k \in K} \nabla f(\tilde{x}^k)^T d^k \leq -\frac{1}{s} \| P(x^* - s \nabla f(x^*))\Delta_{\hat{N}} - x^* \|^2.
\]
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,
\[
\limsup_{k \to \infty, k \in K} \nabla_{\hat{N}} f(\tilde{x}^k)^T d^k < 0,
\]
implying that (20) holds. Finally, since \( \alpha_{\max}^k = 1 \) at every iteration, also (21) is trivially satisfied.

Remark 9. Since we set \( \tilde{x}^k \Delta_{Ak} = 0 \) 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 \( \Delta_{N_k} \). Equivalently, \( \nabla_{N_k} f(\tilde{x}^k)^T d^k_{N_k} < 0 \) if and only if \( \tilde{x}^k_{N_k} \) is non-stationary on the subspace variable \( N_k \).

5 Convergence Rate Analysis.

In this section, we analyze the convergence rate of \textit{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 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 2 and these properties hold only in a neighborhood of stationary point. 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., [23]), and quite reasonable, taking into account the results reported in the previous section.

**Assumption 10.** Let \( \{x^k\} \) be the infinite sequence generated by **AS-SIMPLEX**. We have that
\[
\lim_{k \to \infty} x^k = x^*,
\]
where \( x^* \) is a stationary point of problem (1).

From now on, we denote with \( \bar{I} \) the set \( \{1, \ldots, n\} \). 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., [15]) 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 \( \Delta \). 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 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 [13], adapted to our purposes. Given an index subset \( \bar{I} \subseteq \bar{I} \), we define:

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

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

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 \( \Delta_I \), 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 $\Delta_I$ (and can be interpreted as the lower curvature of the function), is used to give a quadratic lower bound instead (see [19] for further details).

**Remark 11.** The main difference between the constants given above and those introduced in [19] is that ours are restricted to a particular subspace. Moreover, for any index subset $I \subseteq \bar{I}$, it is easy to see that

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

(40)

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

**Theorem 12.** Let Assumption 4 and 10 hold, let $f(x)$ be strongly convex on $\Delta_N$, 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 $k$ such that

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

where

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

**Proof.** Proof. From Theorem 2 exploiting the fact that strict complementarity holds at $x^*$, for sufficiently large $k$ we have that $N(x^k) = N(\tilde{x}^k) = \bar{N}$ and $A(x^k) = A(\tilde{x}^k) = \bar{A}$. From the instructions of **AS-SIMPLEX**, for sufficiently large $k$ we have $\tilde{x}^k_A = x^k_A = 0$, 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 [19], observing that $\mu_f(\bar{N}) > 0$ and $C_f(\bar{N}) < \infty$ under the hypothesis we made. 

**Remark 13.** 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 $\rho_{FW}$ is the constant given in [19] 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., \texttt{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., [15]) 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 \(\Delta_{\mathcal{N}^+}\). 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 \(\Delta\) 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 [20], adapted to our purposes:

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

\[
\mu_{f}^{\Delta}(I) := \inf_{x \in \Delta_I} \inf_{\tilde{x} \in \Delta_I \atop \nabla f(x)^T (\tilde{x} - x) < 0} \frac{2}{\alpha_{f}^{\Delta}(x, \tilde{x})^2} \left[ f(\tilde{x}) - f(x) - \nabla f(x)^T (\tilde{x} - x) \right],
\]

where

\[
\alpha_{f}^{\Delta}(x, \tilde{x}) := \frac{\nabla f(x)^T (\tilde{x} - x)}{\nabla f(x)^T (s_I(x) - v_I(x))},
\]

\[
s_I(x) := e_i, \quad i \in \text{Argmin}_{i \in \bar{I}} \{\nabla_i f(x)\},
\]

\[
v_I(x) := e_j, \quad j \in \text{Argmax}_{j \in I: x_j > 0} \{\nabla_j f(x)\}.
\]

These two new constants are motivated in the analysis by the fact that both Frank-Wolfe and away-step directions are used (see [20] for further details).

**Remark 14.** Also in this case, the main difference between the constants given above and those introduced in [20] 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:

\[
\mu_{f}^{\Delta}(I) \leq C_{f}^{\Delta}(I) \leq C_{f}^{\Delta}(\bar{I}). \tag{41}
\]

Theorem 8 in [20] 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 \(\alpha^k = \alpha^k_{\max} < 1\) 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 15. Let Assumption 4 and 10 hold, let \( f(x) \) be strongly convex on \( \Delta_{N^+} \), 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 \( \tilde{k} \) such that, for every iteration \( k \geq \tilde{k} \) 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

\[
    \rho^{AFW} = \frac{\mu^N_f(N^+)}{4C_f^N(N^+)}, \tag{42}
\]

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

Proof. First, we observe that Theorem 2 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 \( k \geq \tilde{k} \) such that

\[
    [\text{i}]
    \begin{align*}
    1. & \quad x^k_{A^+} = \tilde{x}^k_{A^+} = 0, \text{ for all } k \geq \tilde{k}; \\
    2. & \quad \nabla f(\tilde{x}^k)^T d^k < 0, \text{ for all } k \geq \tilde{k}; \\
    3. & \quad x^* \in \text{Argmin}_{x \in \Delta_{N^k}} f(x), \text{ for all } k \geq \tilde{k}.
    \end{align*}
\]

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 11, this means that \( \tilde{x}^k \) is stationary over \( \Delta_{N^k} \) (but \( \tilde{x}^k \) is not stationary over \( \Delta \)), for all \( k \in K \). Since \( N^k \subseteq N^+ \) for \( k \geq \tilde{k} \) and \( f(x) \) is strongly convex on \( \Delta_{N^+} \), there exists a unique point satisfying stationarity conditions over \( \Delta_{N^k} \) for \( k \geq \tilde{k} \). Taking into account that \( A^k \) and \( N^k \) are subsets of a finite set of indices and \( \tilde{x}^k_{A^k} = 0 \), 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 \( A^k \subseteq \hat{A} \) for all \( k \geq \tilde{k} \) and \( f(x) \) is strongly convex on \( \Delta_{N^+} \).

Consequently, recalling that \( d^k_{A^k} = 0 \), for \( k \geq \tilde{k} \) 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 [20] to provide the following bound:

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

where the last inequality follows from the fact that \( f(\tilde{x}^k) \leq f(x^k) \). Moreover, we have that \( \mu^N_f(N^+) > 0 \) and \( C^N_f(N^+) < \infty \) 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 \( \alpha^k = \alpha^k_{\text{max}} < 1 \), for \( k \geq \tilde{k} \). The fact that \( \alpha^k_{\text{max}} < 1 \) implies that \( d^k = d^A \). Consequently, when \( \alpha^k = \alpha^k_{\text{max}} \), we have that \( x_j^{k+1} = 0 \), where \( j \) 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 \( \tilde{x}_i^{k+1} = 0 \) for all \( k \geq \bar{k} \), we conclude that after at most \(|N^+| - 1 \) iterations with \( \alpha^k = \alpha^\max < 1 \), 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 16.** From [11], it follows that the smaller \( N^+ \), the better the convergence rate of \( \text{AS-SIMPLEX} \). Moreover,

\[
\rho_{\text{AS-AFW}} \geq \frac{\mu_{\max}^2(\bar{I})}{4C_f^2(\bar{I})} = \rho_{\text{FW}},
\]

where \( \rho_{\text{FW}} \) is the constant given in [20] 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., \( \text{AS-SIMPLEX} \) with \( d_i^k \) computed according to (PG) rule) asymptotically converges at a linear rate. We follow the same arguments of the proof of Theorem 3.1 in [21]. First, we need to give two additional results, stated in Lemma 2 and Lemma 3.

**Lemma 2.** 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 2, 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

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

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

**Lemma 3.** Let \( \{x^k\} \) be the sequence produced by Algorithm 1, where \( d_i^k \) 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

\[
\alpha^k > \frac{2\delta(1 - \gamma)}{sL},
\]

\[
f(\tilde{x}^k) - f(x^{k+1}) \geq \frac{2\delta\gamma(1 - \gamma)\min\{1,s\}^2}{s^2L} \|P(\tilde{x}^k - \nabla f(\tilde{x}^k))\|_{N^k} - \tilde{x}^k\|^2,
\]

19
where $\delta$ and $\gamma$ are the parameters used in the Armijo line search.

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

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

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

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

Combining the above inequality with (45), we obtain

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

Using the fact that $\|d^k\| = \|P(\bar{x}^k - s \nabla f(\bar{x}^k))_{N^k} - \bar{x}^k\| \geq \min\{1, s\} \|P(\bar{x}^k - \nabla f(\bar{x}^k))_{N^k} - \bar{x}^k\| \geq \min\{1, s\} \|P(\bar{x}^k - \nabla f(\bar{x}^k))_{N^k} - \bar{x}^k\|$, (see proof of Theorem 4.1 in [21] for the above inequality), we get

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

Combining this inequality with (45), we obtain that (44) holds.

Theorem 17. 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 4 and 10 hold, and let $f(x)$ be strongly convex on $\Delta_{N^+}$.

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 2 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 15 we claim that there exists an iteration $\bar{k}$ such that

1. $x^k_{A^+} = \bar{x}^k_{A^+} = 0$, for all $k \geq \bar{k}$;
2. $\nabla f(\bar{x}^k)^T d^k < 0$, for all $k \geq \bar{k}$;

3. $x^* \in \text{Argmin}_{x \in \Delta_{N_k}} f(x)$, for all $k \geq \bar{k}$.

Hence, by Theorem 2.1 in [21], for sufficiently large $k$ we have

$$
\|\bar{x}^k - x^*\| \leq \tau \|P(\bar{x}^k - \nabla f(\bar{x}^k))_{\Delta_{N_k}} - \bar{x}^k\|,
$$

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 2. Therefore, combining Lemma 2, (44) and (46), for $k \geq \hat{k}$ we can write

$$
f(\bar{x}^k) - f(x^*) \leq \frac{L}{2}\|\bar{x}^k - x^*\|^2
\leq \frac{L}{2}\tau^2\|P(\bar{x}^k - \nabla f(\bar{x}^k))_{\Delta_{N_k}} - \bar{x}^k\|^2
\leq \frac{s^2L^2\tau^2}{4\delta\gamma(1 - \gamma)\min\{1, s\}^2}[f(\bar{x}^k) - f(x^{k+1})].
$$

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

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

where $\rho^{\text{AS-PG}} = \frac{4\delta\gamma(1 - \gamma)\min\{1, s\}^2}{s^2L^2\tau^2}$.

\section{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 Frank-Wolfe, the away-step Frank-Wolfe and the Projected Gradient method, respectively. We further denote by AS-FW, AS-AFW and AS-PG the methods we have from our algorithmic framework, where the search direction $d^k$ is computed according to (FW), (AFW) and (PG) rule, respectively.

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.

We compare the performance of AS-FW, AS-AFW and AS-PG against FW, AFW and PG, respectively, on non-convex quadratic instances that satisfy strict complementarity at a stationary point and on instances from the Chebyshev center problem. All algorithms were stopped at the first iteration $k$ satisfying

$$
\nabla f(x^k)^T(x - x^k) \geq -10^{-6}, \ \forall x \in \Delta,
$$
or in case the maximum number of iterations, denoted with maxit, was reached.

In order to calculate the active-set estimate at each iteration, we need to set the $\epsilon$ parameter to a proper value, so that Assumption 4 is satisfied. In general, the value of this parameter cannot be a priori computed. Following [21,40], 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 2), we accept \( \tilde{x}^k \) and we do not change the value of \( \epsilon \). Otherwise, we do not accept \( \tilde{x}^k \), we reduce \( \epsilon \) and we estimate the active set again, continuing until we get a sufficient decrease in the objective function. The starting value for the \( \epsilon \) 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 Non-Convex Quadratic Instances.

We built instances of problem (1) where \( f(x) = \frac{1}{2}x^TQx - c^Tx \), with \( Q \in \mathbb{R}^{n \times n} \) symmetric and indefinite, and \( c \in \mathbb{R}^n \). More specifically, we generated artificial problems of dimension \( n = 2^{13} \), where the matrix \( Q \in \mathbb{R}^{n \times n} \) is built as the convex combination of a positive definite matrix and a randomly generated symmetric matrix. We further generated a random feasible solution \( x^* \in \mathbb{R}^n \) with \( T = \text{round}(\rho n) \) nonzero variables, where \( \rho \in \{0.1, 0.15, 0.2\} \). Then, the vector \( c \in \mathbb{R}^n \) is defined as \( c = Qx^* - r \), where \( r \in \mathbb{R}^n \) is such that \( r_i = 1 \) if \( x^*_i > 0 \) and \( r_i > 1 \) if \( x^*_i = 0 \). In this way, we ensured that \( x^* \) is a stationary point that satisfies strict complementarity.

For any \( \rho \), we randomly generated 10 different instances and, for each of them, we considered 10 randomly generated starting points, for a total of 100 runs for each \( \rho \). The maximum number of iterations \( \text{maxit} \) was set equal to \( n \).

In Figure 1, we report the optimization error for the comparison between AS-FW and FW, AS-AFW and AFW, and AS-PG and PG, aggregating the results with respect to the sparsity level \( \rho \). More specifically, for an instance \( \text{ins} \) and a given starting point \( \text{sp} \), we ran the compared algorithms (e.g., AS-FW and FW) and evaluated, for each of them, the optimization error \( E^k = f(x^k) - f_{\text{min}} \) versus the computational time, where \( f_{\text{min}} \) is the smallest objective function value obtained by the compared algorithms over the pair \((\text{ins}, \text{sp})\). Then, for every \( \rho \), we averaged the results over the 100 runs.

We can easily see that our active-set framework gets much better performance for every considered sparsity level \( \rho \). In particular, we notice a pretty fast reduction that enables our method to stop much earlier. We further notice that the active-set variants stop at points with better objective function value in the vast majority of the cases.

In Table 1, for each instance considered and each algorithm, we report the average CPU time needed to satisfy the stopping criterion and the average objective function value found (the results are hence averaged over the 10 runs related to the same instance). We can notice that the active-set algorithms have much faster running time (up to two orders of magnitude). With respect to the objective function, we can observe that, in general, the active-set algorithms are able to stop at better points, with differences up to \( 10^{-1} \).

6.2 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 \( C = \{c_1, \ldots, c_n\} \subset \mathbb{R}^m \). 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 \( A = (c_1 \ldots c_n) \in \mathbb{R}^{m \times n} \). We generated instances with
Table 1: Comparison on non-convex quadratic instances.

| $\rho$ | $P$  | CPU time | Obj | CPU time | Obj | CPU time | Obj |
|--------|------|----------|-----|----------|-----|----------|-----|
|       |      | FW AS-FW |     | FW AS-FW |     | FW AS-FW |     |
| 0.1    | $p_1$| 137.56 3.58 | 0.96 0.96 | 138.32 0.54 | 0.97 0.96 | 6.52 0.71 | 0.96 0.96 |
|        | $p_2$| 139.43 3.93 | 0.96 0.96 | 141.45 0.66 | 0.96 0.96 | 5.66 0.75 | 0.96 0.96 |
|        | $p_3$| 139.07 3.79 | 0.96 0.96 | 138.76 0.49 | 0.97 0.96 | 6.19 0.85 | 0.96 0.95 |
|        | $p_4$| 140.82 3.77 | 0.96 0.96 | 120.90 0.48 | 1.01 0.96 | 4.07 0.67 | 0.95 0.96 |
|        | $p_5$| 138.39 3.99 | 0.96 0.96 | 119.94 0.53 | 1.02 0.96 | 5.36 0.86 | 0.96 0.96 |
|        | $p_6$| 136.38 3.74 | 0.96 0.96 | 138.99 0.62 | 0.96 0.96 | 3.90 0.52 | 0.96 0.96 |
|        | $p_7$| 136.55 3.81 | 0.96 0.96 | 140.95 0.55 | 0.97 0.96 | 5.66 0.41 | 0.96 0.95 |
|        | $p_8$| 140.35 3.86 | 0.96 0.96 | 90.12 0.62 | 1.09 0.96 | 5.50 0.52 | 0.96 0.96 |
|        | $p_9$| 138.97 3.20 | 0.96 0.95 | 142.55 0.46 | 0.96 0.95 | 2.58 0.54 | 0.96 0.96 |
|        | $p_{10}$| 142.71 3.72 | 0.96 0.96 | 131.17 0.47 | 0.99 0.96 | 8.14 0.52 | 0.96 0.96 |
| 0.15   | $p_1$| 141.82 3.79 | 0.96 0.96 | 140.99 0.50 | 0.96 0.96 | 4.86 0.64 | 0.96 0.96 |
|        | $p_2$| 139.06 3.42 | 0.96 0.95 | 110.11 0.52 | 1.03 0.96 | 6.80 0.48 | 0.95 0.95 |
|        | $p_3$| 138.80 3.88 | 0.95 0.95 | 130.04 0.49 | 0.98 0.95 | 5.42 0.70 | 0.94 0.95 |
|        | $p_4$| 136.75 4.10 | 0.96 0.96 | 140.98 0.51 | 0.96 0.96 | 5.17 0.47 | 0.95 0.96 |
|        | $p_5$| 131.38 3.69 | 0.95 0.95 | 138.28 0.56 | 0.96 0.95 | 5.46 0.77 | 0.96 0.95 |
|        | $p_6$| 136.49 3.56 | 0.95 0.95 | 121.29 0.47 | 1.01 0.96 | 5.71 0.64 | 0.95 0.95 |
|        | $p_7$| 137.68 3.71 | 0.96 0.95 | 140.66 0.53 | 0.96 0.95 | 7.93 0.62 | 0.96 0.95 |
|        | $p_8$| 138.69 3.58 | 0.96 0.95 | 140.23 0.62 | 0.96 0.95 | 7.52 0.57 | 0.95 0.95 |
|        | $p_9$| 139.71 3.78 | 0.95 0.95 | 132.93 0.58 | 0.98 0.95 | 3.83 0.54 | 0.95 0.95 |
|        | $p_{10}$ | 138.61 4.06 | 0.96 0.96 | 115.25 0.52 | 1.02 0.96 | 5.81 0.73 | 0.96 0.95 |
| 0.2    | $p_1$| 137.04 3.74 | 0.96 0.95 | 132.44 0.53 | 0.97 0.95 | 4.69 0.68 | 0.95 0.95 |
|        | $p_2$| 139.23 3.99 | 0.95 0.96 | 139.39 0.96 | 0.96 0.95 | 6.53 0.70 | 0.95 0.95 |
|        | $p_3$| 137.66 3.93 | 0.96 0.95 | 116.48 0.61 | 1.01 0.95 | 6.09 0.72 | 0.95 0.95 |
|        | $p_4$| 137.40 3.59 | 0.95 0.95 | 115.30 0.55 | 1.00 0.95 | 4.28 0.56 | 0.95 0.95 |
|        | $p_5$| 141.07 3.54 | 0.96 0.95 | 141.59 0.65 | 0.96 0.95 | 4.75 0.61 | 0.95 0.95 |
|        | $p_6$| 138.03 3.71 | 0.95 0.95 | 131.81 0.55 | 0.97 0.95 | 5.16 0.51 | 0.95 0.95 |
|        | $p_7$| 133.11 3.38 | 0.95 0.96 | 108.85 0.48 | 1.02 0.95 | 7.92 0.64 | 0.95 0.95 |
|        | $p_8$| 140.28 3.62 | 0.95 0.95 | 124.84 0.58 | 0.99 0.95 | 7.51 0.96 | 0.95 0.95 |
|        | $p_9$| 135.36 3.43 | 0.95 0.95 | 135.44 0.50 | 0.97 0.95 | 6.77 0.70 | 0.95 0.95 |
|        | $p_{10}$ | 135.33 3.08 | 0.95 0.95 | 138.89 0.40 | 0.97 0.95 | 4.80 0.48 | 0.95 0.94 |
Figure 1: Objective function error vs CPU time (in seconds). Comparison between original and active-set algorithms. Strict complementarity holds at a stationary point. The $y$ axis is in logarithmic scale.

- $n$ (i.e., cardinality of $C$) = $2^{13}$;
- $m$ (i.e., samples’ dimension) = 10, 100, 1000.

For each combination of $n$ and $m$, we randomly generated 10 different instances with $c_i \in \mathbb{R}^m$, $i = 1, \ldots, n$. We further set the maximum number of iterations $\maxit = n$. For all algorithms we fix the starting point to $e_1$ (keep in mind that we deal with convex problems now).

In Figure 2, we report the optimization error for the comparison between AS-FW and FW, AS-AFW and AFW, and AS-PG and PG, aggregating the results with respect to the cardinality of $C$. For every $m$, the results have been averaged over the 10 runs. In each plot, we report, same way as before, the optimization error $E_k = f(x^k) - f_{\min}$ versus the computational time. Again, we can notice that the active-set framework clearly outperforms the original algorithms used in the comparison.

In Table 1, for each instance and each algorithm, we report the average CPU time needed to satisfy the stopping criterion and the average objective function value found.
We can notice a significant difference with respect to the running times, while there are no remarkable differences with respect to the objective function value found by the algorithms. Also in this case, we can notice that the active-set algorithms are up to two orders of magnitude faster than the original ones.

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 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
Table 2: Comparison on instances from the Chebyshev center problem.

| $m$ | $P$ | CPU time | Obj | CPU time | Obj | CPU time | Obj |
|-----|-----|----------|-----|----------|-----|----------|-----|
|     |     | FW AS-FW |     | FW AS-FW |     | FW AS-FW |     |
| 10  | $p_1$ | 147.58 2.24 | -30.40 -30.40 | 3.21 0.09 | -30.40 -30.40 | 36.02 0.21 | -30.40 -30.40 |
|     | $p_2$ | 142.43 2.28 | -30.22 -30.22 | 10.75 0.20 | -30.22 -30.22 | 45.12 0.44 | -30.22 -30.22 |
|     | $p_3$ | 137.45 2.26 | -34.47 -34.47 | 1.13 0.05 | -34.47 -34.47 | 58.92 0.29 | -34.47 -34.47 |
|     | $p_4$ | 144.09 2.30 | -30.09 -30.09 | 43.65 0.46 | -30.09 -30.09 | 39.76 0.62 | -30.09 -30.09 |
|     | $p_5$ | 140.83 0.59 | -29.72 -29.72 | 1.24 0.04 | -29.72 -29.72 | 49.32 0.34 | -29.72 -29.72 |
|     | $p_6$ | 140.82 2.18 | -32.64 -32.64 | 3.71 0.08 | -32.64 -32.64 | 43.28 0.23 | -32.64 -32.64 |
|     | $p_7$ | 135.55 0.15 | -32.86 -32.86 | 1.18 0.05 | -32.86 -32.86 | 21.46 0.13 | -32.86 -32.86 |
|     | $p_8$ | 146.25 0.82 | -32.28 -32.28 | 1.97 0.04 | -32.28 -32.28 | 21.95 0.15 | -32.28 -32.28 |
|     | $p_9$ | 136.53 2.15 | -34.58 -34.59 | 1.20 0.04 | -34.59 -34.59 | 41.12 0.13 | -34.59 -34.59 |
|     | $p_{10}$ | 142.71 2.14 | -30.72 -30.72 | 3.33 0.17 | -30.72 -30.72 | 64.59 0.39 | -30.72 -30.72 |
| 10^2 | $p_1$ | 144.90 2.96 | -147.75 -147.76 | 7.21 0.17 | -147.76 -147.76 | 169.81 0.46 | -147.76 -147.76 |
|     | $p_2$ | 137.93 2.87 | -148.24 -148.25 | 6.39 0.18 | -148.25 -148.25 | 151.57 0.92 | -148.25 -148.25 |
|     | $p_3$ | 136.11 2.53 | -151.99 -151.99 | 5.12 0.10 | -151.99 -151.99 | 106.40 0.39 | -151.99 -151.99 |
|     | $p_4$ | 137.41 3.10 | -145.03 -145.03 | 8.28 0.23 | -145.03 -145.03 | 194.54 0.78 | -145.03 -145.03 |
|     | $p_5$ | 138.20 2.94 | -145.85 -145.85 | 7.31 0.17 | -145.86 -145.86 | 168.24 0.38 | -145.86 -145.86 |
|     | $p_6$ | 134.05 2.99 | -146.22 -146.23 | 7.85 0.21 | -146.23 -146.23 | 172.04 0.49 | -146.23 -146.23 |
|     | $p_7$ | 143.87 3.10 | -146.01 -146.02 | 7.04 0.17 | -146.02 -146.02 | 174.16 0.79 | -146.02 -146.02 |
|     | $p_8$ | 139.09 3.17 | -149.66 -149.67 | 6.71 0.15 | -149.67 -149.67 | 138.02 0.45 | -149.67 -149.67 |
|     | $p_9$ | 137.54 2.91 | -145.90 -145.90 | 6.63 0.15 | -145.91 -145.91 | 144.23 0.50 | -145.91 -145.91 |
|     | $p_{10}$ | 145.02 3.35 | -146.56 -146.56 | 8.75 0.18 | -146.56 -146.56 | 171.81 1.13 | -146.56 -146.56 |
| 10^3 | $p_1$ | 137.07 8.39 | -1114.73 -1114.73 | 22.08 1.42 | -1114.73 -1114.73 | 275.70 10.79 | -1114.73 -1114.73 |
|     | $p_2$ | 136.03 7.21 | -1119.57 -1119.57 | 18.98 1.00 | -1119.57 -1119.57 | 272.22 2.02 | -1119.57 -1119.57 |
|     | $p_3$ | 133.79 7.55 | -1117.10 -1117.10 | 19.25 1.12 | -1117.10 -1117.10 | 275.45 7.65 | -1117.10 -1117.10 |
|     | $p_4$ | 134.84 7.62 | -1117.15 -1117.15 | 20.39 1.16 | -1117.15 -1117.15 | 275.45 1.47 | -1117.15 -1117.15 |
|     | $p_5$ | 139.91 6.20 | -1118.81 -1118.81 | 21.80 0.96 | -1118.81 -1118.81 | 278.43 2.30 | -1118.81 -1118.81 |
|     | $p_6$ | 139.73 6.11 | -1112.68 -1112.68 | 21.72 0.96 | -1112.68 -1112.68 | 267.95 3.23 | -1112.68 -1112.68 |
|     | $p_7$ | 137.25 5.88 | -1119.29 -1119.29 | 20.51 0.91 | -1119.29 -1119.29 | 273.95 1.61 | -1119.29 -1119.29 |
|     | $p_8$ | 132.80 6.20 | -1112.23 -1112.23 | 21.00 1.00 | -1112.23 -1112.23 | 267.31 3.14 | -1112.23 -1112.23 |
|     | $p_9$ | 133.38 6.20 | -1116.77 -1116.77 | 21.45 0.96 | -1116.77 -1116.77 | 275.30 5.39 | -1116.77 -1116.77 |
|     | $p_{10}$ | 132.95 6.81 | -1114.10 -1114.11 | 23.55 1.13 | -1114.11 -1114.11 | 271.47 1.70 | -1114.11 -1114.11 |
linear converge 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 J. Control Optim.*, 20(2):221–246, 1982.
[2] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 1999.
[3] 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.
[4] 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.
[5] 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 J. Optim.*, 26(3):1695–1714, 2016.
[6] K. L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Trans. Algorithms*, 6(4):63, 2010.
[7] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. A Two-Stage Active-Set Algorithm for Bound-Constrained Optimization. *J. Optim. Theory Appl.*, 172(2):369–401, 2017.
[8] E. de Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *CEJOR Cent. Eur. J. Oper. Res.*, 16(2):111–125, 2008.
[9] M. De Santis, G. Di Pillo, and S. Lucidi. An active set feasible method for large-scale minimization problems with bound constraints. *Comput. Opt. Appl.*, 53(2):395–423, 2012.
[10] M. De Santis, S. Lucidi, and F. Rinaldi. A Fast Active Set Block Coordinate Descent Algorithm for $\ell_1$-Regularized Least Squares. *SIAM J. Optim.*, 26(1):781–809, 2016.
[11] 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.
[12] 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. *arXiv preprint arXiv:1705.01797*, 2017.
[13] F. Facchinei and S. Lucidi. Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. *J. Optim. Theory Appl.*, 85(2):265–289, 1995.
[14] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.
[15] J. Guélat and P. Marcotte. Some comments on Wolfe’s ‘away step’. *Math. Program.*, 35 (1):110–119, 1986.
[16] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM J. Optim.*, 17(2):526–557, 2006.
[17] W. W. Hager and H. Zhang. An active set algorithm for nonlinear optimization with
polyhedral constraints. *Science China Mathematics*, 59(8):1525–1542, 2016.

[18] W. W. Hager and H. Zhang. Projection onto a polyhedron that exploits sparsity. *SIAM J. Optim.*, 26(3):1773–1798, 2016.

[19] S. Lacoste-Julien and M. Jaggi. An affine invariant linear convergence analysis for Frank-Wolfe algorithms. *arXiv preprint arXiv:1312.7864*, 2013.

[20] 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.

[21] Z.-Q. Luo and P. Tseng. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM J. Control Optim.*, 30(2):408–425, 1992.

[22] J. Nocedal and S. J. Wright. Numerical optimization, 2006.

[23] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[24] P. Wolfe. Convergence theory in nonlinear programming. *Integer and nonlinear programming*, pages 1–36, 1970.