Math. Program., Ser. A (2017) 164:1–27
DOI 10.1007/s10107-016-1069-4

FULL LENGTH PAPER

Linearly convergent away-step conditional gradient for non-strongly convex functions

Amir Beck[1] · **Shimrit Shtern**[1]

Received: 19 April 2015 / Accepted: 2 September 2016 / Published online: 21 September 2016

Abstract We consider the problem of minimizing the sum of a linear function and a composition of a strongly convex function with a linear transformation over a compact polyhedral set. Jaggi and Lacoste-Julien (An affine invariant linear convergence analysis for Frank-Wolfe algorithms. NIPS 2013 Workshop on Greedy Algorithms, Frank-Wolfe and Friends, 2014) show that the conditional gradient method with away steps — employed on the aforementioned problem without the additional linear term — has a linear rate of convergence, depending on the so-called pyramidal width of the feasible set. We revisit this result and provide a variant of the algorithm and an analysis based on simple linear programming duality arguments, as well as corresponding error bounds. This new analysis (a) enables the incorporation of the additional linear term, and (b) depends on a new constant, that is explicitly expressed in terms of the problem's parameters and the geometry of the feasible set. This constant replaces the pyramidal width, which is difficult to evaluate.

Mathematics Subject Classification 90-08 · 90C25 · 65K10

✉ Amir Beck
becka@ie.technion.ac.il

Shimrit Shtern
shimrits@tx.technion.ac.il

[1] Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Haifa, Israel

1 Introduction

Consider the minimization problem

$$\min_{\mathbf{x}\in X} \{ f(\mathbf{x}) \equiv g(\mathbf{E}\mathbf{x}) + \langle \mathbf{b}, \mathbf{x} \rangle \}, \tag{P}$$

where $X \subseteq \mathbb{R}^n$ is a compact polyhedral set, $\mathbf{E} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^n$ and $g : \mathbb{R}^m \to \mathbb{R}$ is strongly convex and continuously differentiable over $\mathbb{R}^m$. Note that for a general matrix $\mathbf{E}$, the function f is not necessarily strongly convex.

When the problem at hand is large-scale, first-order methods, which have relatively low computational cost per iteration, are usually utilized. These methods include, for example, the class of projected (proximal) gradient methods. A drawback of these methods is that under general convexity assumptions, they posses only a sublinear rate of convergence [2,19], while a linear rate of convergence can be established only under additional conditions such as strong convexity of the objective function [19]. Luo and Tseng [21] show that the strong convexity assumption can be relaxed and replaced by an assumption on the existence of a local error bound, and under this assumption, certain classes of algorithms, which they refer to as "feasible descent methods", converge in an asymptotic linear time. The model (P) with assumptions on strong convexity of g, compactness and polyhedrality of X was shown in [21] to satisfy the error bound. Wang and Lin [23] extend [21] and show that there exists a *global* error bound for problem (P) with the additional assumption of compactness of X, and derive the exact linear rate for this case. Note that the family of "feasible descent methods" includes the block alternating minimization algorithm (under the assumption of block strong convexity), as well as gradient projection methods, and therefore are usually at least as complex as evaluating at each iteration the orthogonal projection operator onto the feasible set X.

An alternative to algorithms based on projection (or proximal) operators are *linear-oracle*-based algorithms such as the conditional gradient (CG) method. The CG algorithm was presented by Frank and Wolfe [8] in 1956 for minimizing a convex function over a compact polyhedral set. At each iteration, the algorithm requires a solution to the problem of minimizing a linear objective function over the feasible set. It is assumed that this solution is obtained by a call to a linear-oracle, i.e., a black box which, given a linear function, returns an optimal solution of this linear function over the feasible set (see an exact definition in Sect. 2.3). In some instances, and specifically for certain types of polyhedral sets, obtaining such a linear-oracle can be done more efficiently than computing the orthogonal projection onto the feasible set (see examples in [10]), and therefore the CG algorithm has an advantage over projection-based algorithms. The original paper of Frank and Wolfe also contains a proof of an $O(1/k)$ rate of convergence of the function values to the optimal value. Levitin and Polyak [18] show that this $O(1/k)$ rate can also be extended to the case where the feasible set is a general compact convex set. Cannon and Culum [5] prove that this rate is in fact *tight*. However, if in addition to strong convexity of the objective function, the optimal solution is in the interior of the feasible set, then linear rate of convergence

of the CG method can be established [12].[1] Epelman and Freund [7], as well as Beck and Teboulle [1], show a linear rate of convergence of the conditional gradient with a special stepsize choice in the context of finding a point in the intersection of an affine space and a closed and convex set under a Slater-type assumption. Another setting in which linear rate of convergence can be derived is when the feasible set is uniformly (strongly) convex and the norm of the gradient of the objective function is bounded away from zero [18].

Another approach for deriving a linear rate of convergence is to modify the algorithm. For example, Hazan and Garber [10] use *local* linear-oracles in order to show linear rate of convergence of a "localized" version of the conditional gradient method. A different modification is to use a variation of the conditional gradient method that incorporates away steps. This version of the conditional gradient method, which we refer to as *away steps conditional gradient* (ASCG), was initially suggested by Wolfe [24] and then studied by Guelat and Marcotte [12], where a linear rate of convergence was established under the assumption that the objective function is strongly convex and the set is a compact polytope, as well as an assumption on the location of the optimal solution. A modified version of the away step method, which calculates the away step on a simplex type representation of the problem, was studied by Jaggi and Lacoste-Julien [16], which were able to show linear convergence for the more general model (P) for the case where $\mathbf{b} = \mathbf{0}$, without restrictions on the location of the solution. Jaggi proves [15] that the rate of convergence of any method which only adds or removes one atom at a time must be at least sublinear in the first n iterations with n being the dimension of the space. Although the original ASCG suggested by Wolfe and the modified ASCG discussed in [16] are closely related, they have some significant differences. Specifically, as stated in recent work of Freund, Grigas and Mazumder [9], in order to analyze the convergence of this new modification of the ASCG algorithm, it is required that the linear-oracle produces outputs which reside in a certain finite set of points (e.g., the set of extreme points in the case of a polytope), while the analysis of the original ASCG in [12] does not require such an assumption. In this paper we will refer to the variation presented in [16] as "the ASCG method", and call its required oracle a *vertex linear-oracle* (see the discussion in Sect. 3.1).

Contribution. In this work, our starting point and main motivation are the results of Jaggi and Lacoste-Julien [16]. Our contribution is twofold:

(a) We extend the results given in [16] and show that the ASCG algorithm converges linearly for the general case of problem (P), that is, for any value of $\mathbf{E}$ and $\mathbf{b}$. The additional linear term $\langle \mathbf{b}, \mathbf{x} \rangle$ enables us to consider much more general models. For example, consider the l_1-regularized least squares problem $\min_{\mathbf{x} \in S} \{\|\mathbf{B}\mathbf{x} - \mathbf{c}\|_2^2 + \lambda \|\mathbf{x}\|_1\}$, where $S \subseteq \mathbb{R}^n$ is a compact polyhedral set, $\mathbf{B} \in \mathbb{R}^{k \times n}$, $\mathbf{c} \in \mathbb{R}^k$ and $\lambda > 0$. Since S is compact, there exists a constant $M > 0$ for which $\|\mathbf{x}\|_1 \leq M$ for any optimal solution $\mathbf{x}$.We can now rewrite the model as

[1] The paper [12] assumes that the feasible set is a bounded polyhedral, but the proof is actually correct for general compact convex sets.

$$\min_{\mathbf{x}\in S, \|\mathbf{x}\|_1 \le y, y\in[0,M]} \|\mathbf{B}\mathbf{x}-\mathbf{c}\|^2 + \lambda y,$$

which obviously fits the general model (P).

(b) We present an analysis based on simple linear programming duality arguments, which are completely different than the geometric arguments in [16]. Consequently, we obtain a computable constant for the rate of convergence, which is explicitly expressed as a function of the problem's parameters and the geometry of the feasible set. This constant, which we call "the vertex-facet distance constant", replaces the so-called *pyramidal width* constant from [16], which reflects the geometry of the feasible set and is obtained as the optimal value of a very complex mixed integer saddle point optimization problem whose exact value is unknown even for simple polyhedral sets.

Paper outline. The paper is organized as follows. Section 2 presents some preliminary results and definitions needed for the analysis. In particular, it provides a brief introduction to the classical CG algorithm and linear oracles. Section 3 presents the ASCG algorithm and the convergence analysis, and is divided into four subsections. In Sect. 3.1 the concept of vertex linear-oracle, needed for the implementation of ASCG, is presented, and the difficulties of obtaining a vertex linear-oracle on a linear transformation of the feasible set are discussed. In Sect. 3.2 we present the ASCG method with different possible stepsize choices. In Sect. 3.3, we provide the rate of convergence analysis of the ASCG for problem (P), and present the new *vertex-facet distance* constant used in the analysis. In Sect. 3.4, we demonstrate how to compute this new constant for a few examples of simple polyhedral sets.

Notations. We denote the cardinality of set I by $|I|$. The difference, union and intersection of two given sets I and J are denoted by $I\backslash J = \{a \in I : a \notin J\}$, $I \cup J$ and $I \cap J$ respectively. Subscript indices represent elements of a vector, while superscript indices represent iterates of the vector, i.e., x_i is the ith element of vector $\mathbf{x}$, $\mathbf{x}^k$ is a vector at iteration k, and x_i^k is the ith element of $\mathbf{x}^k$. The vector $\mathbf{e}_i \in \mathbb{R}^n$ is the ith vector of the standard basis of $\mathbb{R}^n$, $\mathbf{0} \in \mathbb{R}^n$ is the all-zeros vector, and $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones. Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, their dot product is denoted by $\langle \mathbf{x}, \mathbf{y}\rangle$. Given a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{A}\|$ denotes the spectral norm of $\mathbf{A}$, and $\|\mathbf{x}\|$ denotes the ℓ_2-norm of $\mathbf{x}$, unless stated otherwise. $\mathbf{A}^T$, $\mathrm{rank}(\mathbf{A})$ and $\mathrm{Im}(\mathbf{A})$ represent the transpose, rank and image of $\mathbf{A}$ respectively. We denote the ith row of a given matrix $\mathbf{A}$ by $\mathbf{A}_i$, and given a set $I \subseteq \{1,\ldots,m\}$, $\mathbf{A}_I \in \mathbb{R}^{|I|\times n}$ is the submatrix of $\mathbf{A}$ such that $(\mathbf{A}_I)_j = \mathbf{A}_{I_j}$ for any $j = 1,\ldots,|I|$. If $\mathbf{A}$ is a symmetric matrix, then $\lambda_{\min}(\mathbf{A})$ is its minimal eigenvalue. If a matrix $\mathbf{A}$ is also invertible, we denote its inverse by $\mathbf{A}^{-1}$. Given matrices $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times k}$, the matrix $[\mathbf{A}, \mathbf{B}] \in \mathbb{R}^{n\times(m+k)}$ is their horizontal concatenation. Given a point $\mathbf{x}$ and a closed convex set X, the distance between $\mathbf{x}$ and X is denoted by $d(\mathbf{x}, X) = \min_{\mathbf{y}\in X} \|\mathbf{x}-\mathbf{y}\|$. The standard unit simplex in $\mathbb{R}^n$ is denoted by $\Delta_n = \left\{\mathbf{x} \in \mathbb{R}^n_+ : \langle \mathbf{1}, \mathbf{x}\rangle = 1\right\}$ and its relative interior by $\Delta_n^+ = \left\{\mathbf{x} \in \mathbb{R}^n_{++} : \langle \mathbf{1}, \mathbf{x}\rangle = 1\right\}$. Given a set $X \subseteq \mathbb{R}^n$, its convex hull is denoted by $\mathrm{conv}(X)$. Given a convex set C, the set of all its extreme points is denoted by $\mathrm{ext}(C)$.

2 Preliminaries

2.1 Mathematical preliminaries

We start by presenting two technical lemmas. The first lemma is the well known *descent lemma* which is fundamental in convergence rate analysis of first-order methods. The second lemma is *Hoffman's lemma* which is used in various error bound analyses over polyhedral sets.

Lemma 2.1 (The Descent Lemma [3, Proposition A.24]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function whose gradient is Lipschitz continuous with constant ρ. Then for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ we have

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\rho}{2} \|\mathbf{x} - \mathbf{y}\|^2 .$$

Lemma 2.2 (Hoffman's Lemma [14]) Let X be a polyhedron defined by $X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{a}\}$, for some $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{a} \in \mathbb{R}^m$, and let $S = \left\{\mathbf{x} \in \mathbb{R}^n : \tilde{\mathbf{E}}\mathbf{x} = \tilde{\mathbf{e}}\right\}$ where $\tilde{\mathbf{E}} \in \mathbb{R}^{r \times n}$ and $\tilde{\mathbf{e}} \in \mathbb{R}^r$. Assume that $X \cap S \neq \emptyset$. Then, there exists a constant θ, depending only on $\mathbf{A}$ and $\tilde{\mathbf{E}}$, such that any $\mathbf{x} \in X$ satisfies

$$d(\mathbf{x}, X \cap S) \leq \theta \left\| \tilde{\mathbf{E}}\mathbf{x} - \tilde{\mathbf{e}} \right\| .$$

A complete and simple proof of this lemma is given in [13, pg. 299–301]. Defining $\mathcal{B}$ as the set of all matrices constructed by taking linearly independent rows from the matrix $\left[\tilde{\mathbf{E}}^T, \mathbf{A}^T\right]^T$, we can write θ as

$$\theta = \max_{\mathbf{B} \in \mathcal{B}} \frac{1}{\lambda_{\min}\left(\mathbf{B}\mathbf{B}^T\right)}.$$

We will refer to θ as the *Hoffman constant* associated with matrix $\left[\tilde{\mathbf{E}}^T, \mathbf{A}^T\right]^T$.

2.2 Problem properties

Throughout this paper we make the following assumption regarding problem (P).

Assumption 1 (a) f is continuously differentiable and has a Lipschitz continuous gradient with constant ρ.
(b) g is strongly convex with parameter σ_g.
(c) X is a nonempty compact polyhedral set given by $X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{a}\}$ for some $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{a} \in \mathbb{R}^m$.

We assume without loss of generality that $\mathbf{A}$ does not contain a row of zeros. We denote the optimal solution set of problem (P) by X^*. The diameter of the compact

set X is denoted by D, and the diameter of the set $\mathbf{E}X$ (the diameter of the image of X under the linear mapping associated with matrix $\mathbf{E}$) by $D_{\mathbf{E}}$. The two diameters satisfy the following relation:

$$D_{\mathbf{E}} = \max_{\mathbf{x},\mathbf{y}\in X} \|\mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{y}\| \leq \|\mathbf{E}\| \max_{\mathbf{x},\mathbf{y}\in X} \|\mathbf{x} - \mathbf{y}\| = \|\mathbf{E}\| D,$$

We define $G \equiv \max_{\mathbf{x}\in X} \|\nabla g(\mathbf{E}\mathbf{x})\|$ to be the maximal norm of the gradient of g over $\mathbf{E}X$.

Problem (P) possesses some properties, which we present in the following lemmas.

Lemma 2.3 (Lemma 14, [23]) Let X^* be the optimal set of problem (P). Then, there exists a constant vector $\mathbf{t}^*$ and a scalar s^* such that any optimal solution $\mathbf{x}^* \in X^*$ satisfies $\mathbf{E}\mathbf{x}^* = \mathbf{t}^*$ and $\langle \mathbf{b}, \mathbf{x}^* \rangle = s^*$.

Although the proof of the lemma in the given reference is for polyhedral sets, the extension for any convex set is trivial.

Lemma 2.4 *Let f^* be the optimal value of problem (P). Then, for any $\mathbf{x} \in X$*

$$f(\mathbf{x}) - f^* \leq C$$

where $C = G D_{\mathbf{E}} + \|\mathbf{b}\| D$.

Proof Let $\mathbf{x}^*$ be some optimal solution of problem (P), so that $f(\mathbf{x}^*) = f^*$. Then for any $\mathbf{x} \in X$, it follows from the convexity of f that

$$\begin{aligned} f(\mathbf{x}) - f(\mathbf{x}^*) &\leq \left\langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}^* \right\rangle \\ &= \left\langle \nabla g(\mathbf{E}\mathbf{x}), \mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^* \right\rangle + \left\langle \mathbf{b}, \mathbf{x} - \mathbf{x}^* \right\rangle \\ &\leq \|\nabla g(\mathbf{E}\mathbf{x})\| \left\|\mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^*\right\| + \|\mathbf{b}\| \left\|\mathbf{x} - \mathbf{x}^*\right\| \\ &\leq G D_{\mathbf{E}} + \|\mathbf{b}\| D = C \end{aligned}$$

where the last two inequalities are due to the Cauchy-Schwarz inequality and the definition of G,D and $D_{\mathbf{E}}$. □

The following lemma provides an *error bound*, i.e., a bound on the distance of any feasible solution to the optimal set. This error bound will later be used as an alternative to a strong convexity assumption on f, which is usually needed in order to prove a linear rate of convergence. This is a different bound than the one given in [23], since it relies heavily on the compactness of the set X, thus enabling to circumvent the use of the so-called gradient mapping.

Lemma 2.5 *For any $\mathbf{x} \in X$,*

$$d(\mathbf{x}, X^*)^2 \leq \kappa (f(\mathbf{x}) - f^*),$$

where $\kappa = \theta^2 \left(\|\mathbf{b}\| D + 3G D_{\mathbf{E}} + \frac{2(G^2+1)}{\sigma_g} \right)$, and θ is the Hoffman constant associated with the matrix $\left[\mathbf{A}^T, \mathbf{E}^T, \mathbf{b}\right]^T$.

Proof Lemma 2.3 implies that the optimal solution set X^* can be defined as $X^* = X \cap S$ where $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{E}\mathbf{x} = \mathbf{t}^*,\ \langle \mathbf{b}, \mathbf{x}\rangle = s^*\}$ for some $\mathbf{t}^* \in \mathbb{R}^m$ and $s^* \in \mathbb{R}$. For any $\mathbf{x} \in X$, applying Lemma 2.2 with $\tilde{\mathbf{E}} = \left[\mathbf{E}^T, \mathbf{b}\right]^T$, we have that

$$d(\mathbf{x}, X^*)^2 \leq \theta^2\left(\left(\langle \mathbf{b}, \mathbf{x}\rangle - s^*\right)^2 + \left\|\mathbf{E}\mathbf{x} - \mathbf{t}^*\right\|^2\right), \tag{2.1}$$

where θ is the Hoffman constant associated with matrix $\left[\mathbf{A}^T, \mathbf{E}^T, \mathbf{b}\right]^T$. Now, let $\mathbf{x} \in X$ and $\mathbf{x}^* \in X^*$. Utilizing the σ_g-strong convexity of g, it follows that

$$\left\langle \nabla g(\mathbf{E}\mathbf{x}^*), \mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^*\right\rangle + \frac{\sigma_g}{2}\left\|\mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^*\right\|^2 \leq g(\mathbf{E}\mathbf{x}) - g(\mathbf{E}\mathbf{x}^*). \tag{2.2}$$

By the first-order optimality conditions for problem (P), we have (recalling that $\mathbf{x} \in X$ and $\mathbf{x}^* \in X^*$)

$$\left\langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\right\rangle \geq 0. \tag{2.3}$$

Therefore,

$$\begin{aligned}\frac{\sigma_g}{2}\left\|\mathbf{E}\mathbf{x} - \mathbf{t}^*\right\|^2 &\leq \left\langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\right\rangle + \frac{\sigma_g}{2}\left\|\mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^*\right\|^2 \\ &= \left\langle \nabla g(\mathbf{E}\mathbf{x}^*), \mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^*\right\rangle + \left\langle \mathbf{b}, \mathbf{x} - \mathbf{x}^*\right\rangle + \frac{\sigma_g}{2}\left\|\mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^*\right\|^2.\end{aligned} \tag{2.4}$$

Now, using (2.2) we can continue (2.4) to obtain

$$\frac{\sigma_g}{2}\left\|\mathbf{E}\mathbf{x} - \mathbf{t}^*\right\|^2 \leq g(\mathbf{E}\mathbf{x}) - g(\mathbf{E}\mathbf{x}^*) + \langle \mathbf{b}, \mathbf{x}\rangle - \left\langle \mathbf{b}, \mathbf{x}^*\right\rangle = f(\mathbf{x}) - f(\mathbf{x}^*). \tag{2.5}$$

We are left with the task of upper bounding $(\langle \mathbf{b}, \mathbf{x}\rangle - s^*)^2$. By the definitions of s^* and f we have that

$$\begin{aligned}\langle \mathbf{b}, \mathbf{x}\rangle - s^* &= \left\langle \mathbf{b}, \mathbf{x} - \mathbf{x}^*\right\rangle \\ &= \left\langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\right\rangle - \left\langle \nabla g(\mathbf{E}\mathbf{x}^*), \mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^*\right\rangle \\ &= \left\langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\right\rangle - \left\langle \nabla g(\mathbf{t}^*), \mathbf{E}\mathbf{x} - \mathbf{t}^*\right\rangle.\end{aligned} \tag{2.6}$$

Therefore, using (2.3), (2.6) as well as the Cauchy-Schwarz inequality, we can conclude the following:

$$s^* - \langle \mathbf{b}, \mathbf{x}\rangle \leq \left\langle \nabla g(\mathbf{t}^*), \mathbf{E}\mathbf{x} - \mathbf{t}^*\right\rangle \leq \left\|\nabla g(\mathbf{t}^*)\right\|\left\|\mathbf{E}\mathbf{x} - \mathbf{t}^*\right\|. \tag{2.7}$$

On the other hand, exploiting (2.6), the convexity of f and the Cauchy-Schwarz inequality, we also have that

$$\begin{aligned}\langle \mathbf{b}, \mathbf{x}\rangle - s^* &= \left\langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\right\rangle - \left\langle \nabla g(\mathbf{t}^*), \mathbf{E}\mathbf{x} - \mathbf{t}^*\right\rangle \\ &\leq f(\mathbf{x}) - f^* - \left\langle \nabla g(\mathbf{t}^*), \mathbf{E}\mathbf{x} - \mathbf{t}^*\right\rangle \\ &\leq f(\mathbf{x}) - f^* + \left\|\nabla g(\mathbf{t}^*)\right\|\left\|\mathbf{E}\mathbf{x} - \mathbf{t}^*\right\|.\end{aligned} \tag{2.8}$$

Combining (2.7), (2.8), and the fact that $f(\mathbf{x}) - f^* \geq 0$, we obtain that

$$(\langle \mathbf{b}, \mathbf{x} \rangle - s^*)^2 \leq \left(f(\mathbf{x}) - f^* + \left\| \nabla g(\mathbf{t}^*) \right\| \left\| \mathbf{E}\mathbf{x} - \mathbf{t}^* \right\| \right)^2 . \tag{2.9}$$

Moreover, the definitions of G and $D_{\mathbf{E}}$ imply $\|\nabla g(\mathbf{t}^*)\| \leq G$, $\|\mathbf{E}\mathbf{x} - \mathbf{t}^*\| \leq D_{\mathbf{E}}$, and since $\mathbf{x} \in X$, it follows from Lemma 2.4 that $f(\mathbf{x}) - f^* \leq C = GD_{\mathbf{E}} + \|\mathbf{b}\| D$. Utilizing these bounds, as well as (2.5) to bound (2.9) results in

$$\begin{aligned}
(\langle \mathbf{b}, \mathbf{x} \rangle - s^*)^2 &\leq \left(f(\mathbf{x}) - f^* + G \left\| \mathbf{E}\mathbf{x} - \mathbf{t}^* \right\| \right)^2 \\
&= (f(\mathbf{x}) - f^*)^2 + 2G \left\| \mathbf{E}\mathbf{x} - \mathbf{t}^* \right\| (f(\mathbf{x}) - f^*) + G^2 \left\| \mathbf{E}\mathbf{x} - \mathbf{t}^* \right\|^2 \\
&\leq (f(\mathbf{x}) - f^*)C + 2GD_{\mathbf{E}}(f(\mathbf{x}) - f^*) + G^2 \frac{2}{\sigma_g}(f(\mathbf{x}) - f^*) \\
&= (f(\mathbf{x}) - f^*) \left(C + 2GD_{\mathbf{E}} + \frac{2G^2}{\sigma_g} \right) \\
&= (f(\mathbf{x}) - f^*) \left(\|\mathbf{b}\| D + 3GD_{\mathbf{E}} + \frac{2G^2}{\sigma_g} \right) .
\end{aligned} \tag{2.10}$$

Plugging (2.5) and (2.10) back into (2.1), we obtain the desired result:

$$d(\mathbf{x}, X^*)^2 \leq \theta^2 \left(\|\mathbf{b}\| D + 3GD_{\mathbf{E}} + \frac{2(G^2 + 1)}{\sigma_g} \right) (f(\mathbf{x}) - f^*).$$

□

2.3 Conditional gradient and linear oracles

In order to present the CG algorithm, we first define the concept of linear oracles.

Definition 2.1 (*Linear Oracle*) Given a set X, an operator $\mathcal{O}_X : \mathbb{R}^n \to X$ is called a **linear oracle** for X, if for each $\mathbf{c} \in \mathbb{R}^n$ it returns a vector $\mathbf{p} \in X$ such that $\langle \mathbf{c}, \mathbf{p} \rangle \leq \langle \mathbf{c}, \mathbf{x} \rangle$ for any $\mathbf{x} \in X$, i.e., $\mathbf{p}$ is a minimizer of the linear function $\langle \mathbf{c}, \mathbf{x} \rangle$ over X.

Linear oracles are black-box type functions, where the actual algorithm used in order to obtain the minimizer is unknown. For many feasible sets, such as ℓ_p balls and specific polyhedral sets, the oracle can be represented by a closed form solution or can be computed by an efficient method.

The CG algorithm and its variants are linear-oracle based algorithms. The original CG algorithm, presented in [8]—also known as the Frank-Wolfe algorithm — is as follows.

Conditional Gradient Algorithm (CG)

Input: A linear oracle $\mathcal{O}_X$

Initialize: $\mathbf{x}^1 \in X$

For $k = 1, 2, \ldots$

1. Compute $\mathbf{p}^k := \mathcal{O}_X(\nabla f(\mathbf{x}^k))$.
2. Choose a stepsize γ^k.
3. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \gamma^k(\mathbf{p}^k - \mathbf{x}^k)$.

The algorithm is guaranteed to have an $O(\frac{1}{k})$ rate of convergence for stepsize determined according to exact line search [8], adaptive stepsize [18], which has a closed form when the smoothness constant of the objective function is known, and predetermined stepsize [6]. This upper bound on the rate of convergence is tight [5] and therefore variants, such as the ASCG were developed.

3 Away steps conditional gradient

The ASCG algorithm was proposed by Wolfe in [24]. A linear convergence rate was proven for problems consisting of minimizing strongly convex objective functions over polyhedral feasible sets in [12] under some restrictions on the location of the optimal solution. Jaggi and Lacoste-Julien [16] suggested a variation on the original ASCG algorithm, which assumes that a finite representation of the current point is maintained throughout the algorithm, and this representation is used to calculate the away step. The authors then showed that under this modification, it is possible to prove a linear convergence result without a restriction on the optimal solution's location; this result is also applicable for the specific case of problem (P) where $\mathbf{b} = \mathbf{0}$ [or more generally $\mathbf{b} \in \text{Im}(\mathbf{E})$], provided that an appropriate linear-oracle is available for the set $\mathbf{E}X$. From this point on we refer to the latter variant of the ASCG, where a thorough explanation of the difference between the algorithms can be found in [9]. In this section, we extend this result for the general case of problem (P), i.e., for any $\mathbf{E}$ and $\mathbf{b}$. Furthermore, we explore the potential issues with obtaining a linear-oracle for the set $\mathbf{E}X$, and suggest an alternative analysis, which only assumes existence of an appropriate linear-oracle on the original set X. Moreover, our analysis differs from the one presented in [16] by the fact that it is based on duality rather than geometric arguments. This approach enables to derive a computable constant for the rate of convergence, which is explicitly expressed as a function of the problem's parameters and the geometry of the feasible set.

We separate the discussion on the ASCG method into four sections. In Sect. 3.1 we define the concept of *vertex linear oracles*, and the issues of obtaining such an oracle for linear transformations of simple sets. Section 3.2 contains a full description of the ASCG method itself, including the concept of vertex representation, and representation reduction. In Sect. 3.3 we present the rate of convergence analysis of the ASCG method for problem (P), as well as introduce the new computable convergence constant Ω_X. Finally, in Sect. 3.4 we demonstrate how to compute Ω_X for three types of simple sets.

3.1 Vertex linear oracles

In order to prove convergence, the ASCG algorithm requires a linear oracle which output is within a finite set of points in X; more specifically, in the case of a polytope, it is natural to assume a *vertex linear oracle*,[2] a concept that we now define explicitly.

Definition 3.1 (*Vertex Linear Oracle*) Given a polyhedral set X with vertex set V, a linear oracle $\tilde{\mathcal{O}}_X : \mathbb{R}^n \to V$ is called a **vertex linear oracle** for X, if for each $\mathbf{c} \in \mathbb{R}^n$ it returns a vertex $\mathbf{p} \in V$ such that $\langle \mathbf{c}, \mathbf{p} \rangle \leq \langle \mathbf{c}, \mathbf{x} \rangle$ for any $\mathbf{x} \in X$.

Notice that, according to the fundamental theorem of linear programming [4, Theorem 2.7], the problem of optimizing any linear objective function over the compact set X always has an optimal solution which is a vertex. Therefore, the vertex linear oracle $\tilde{\mathcal{O}}_X$ is well defined. We also note that in this paper the term "vertex" is synonymous with the term "extreme point".

In [16], Jaggi and Lacoste-Julien proved that both the CG and the ASCG algorithms are affine invariant. This means that given the problem

$$\min_{\mathbf{x} \in X} g(\mathbf{E}\mathbf{x}), \tag{3.1}$$

where g is a strongly convex function and $\mathbf{E}$ is some matrix, applying the ASCG algorithm on the equivalent problem

$$\min_{\mathbf{y} \in Y} g(\mathbf{y}), \tag{3.2}$$

where $Y = \mathbf{E}X$, yields a linear rate of convergence, which depends only on the strong convexity parameter of g and the geometry of the set Y (regardless of what $\mathbf{E}$ generated it). However, assuming that $\mathbf{E}$ is not of a full column rank, i.e., f is not strongly convex, retrieving an optimal solution $\mathbf{x}^* \in X$ from the optimal solution $\mathbf{y}^* \in Y$ requires solving a linear feasibility problem, where we aim to find $\mathbf{x}^* \in X$ such that $\mathbf{E}\mathbf{x}^* = \mathbf{y}^*$. This feasibility problem is equivalent to solving the following constrained least squares problem:

$$\min_{\mathbf{x} \in X} \left\| \mathbf{E}\mathbf{x} - \mathbf{y}^* \right\|^2$$

in the sense that the optimal set of the above problem comprises all vectors satisfying $\mathbf{x}^* \in X, \mathbf{E}\mathbf{x}^* = \mathbf{y}^*$. For a general $\mathbf{E}$, solving the least squares problem might be more computationally expensive than simply applying the linear oracle on set X. Moreover, the entire algorithm is then applied on the transformed set $Y = \mathbf{E}X$, which

[2] This is how the algorithm is described in [16] although in [17] the authors extend this result to include atom linear oracles, which are oracles whose output is within a predetermined finite set of points, called atoms. This set of atoms includes but is not limited to the set of vertices.

might be geometrically more complicated than the set X, or may lie in a higher dimension.

It is interesting to note that the vertex oracle property is not preserved when changing the representation. As an example, let us take a naive approach to construct a general linear oracle $\mathcal{O}_{\mathbf{E}X}$, given $\tilde{\mathcal{O}}_X$, by the formula

$$\mathcal{O}_{\mathbf{E}X}(\mathbf{c}) = \mathbf{E}\tilde{\mathcal{O}}_X(\mathbf{E}^T\mathbf{c}). \tag{3.3}$$

However, the output $\tilde{\mathbf{p}} = \mathcal{O}_{\mathbf{E}X}(\mathbf{c})$ of this linear oracle is not guaranteed to be a vertex of $\mathbf{E}X$ and incurs an additional computational cost associated with the matrix vector multiplications. As an example, take X to be the unit box in three dimensions, $X = [-1, 1]^3 \subseteq \mathbb{R}^3$, and let $\mathbf{E}$ be given by

$$\mathbf{E} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix}.$$

We denote the vertex set V of the set X by the letters A–H as follows:

$$A = (1, 1, 1)^T, \quad B = (1, 1, -1)^T, \quad C = (1, -1, -1)^T, \quad D = (1, -1, 1)^T,$$
$$E = (-1, 1, 1)^T, \quad F = (-1, -1, 1)^T, \quad G = (-1, 1, -1)^T, \quad H = (-1, -1, -1)^T,$$

and the linear mappings of these vertices by the matrix $\mathbf{E}$ by A'–H':

$$A' = (3, 1, 2)^T, \quad B' = (1, 3, -2)^T, \quad C' = G' = (-1, 1, -2)^T,$$
$$F' = (-1, -3, 2)^T, \quad H' = (-3, -1, -2)^T, \quad D' = E' = (1, -1, 2)^T.$$

The vertex set of $\mathbf{E}X$ is $\text{ext}(\mathbf{E}X) = \{A', B', F', H'\}$.

The sets X and $\mathbf{E}X$ are presented in Fig. 1. Notice that finding a vertex linear oracle for X is trivial, while finding one for $\mathbf{E}X$ is not. In particular, a vertex linear oracle for X may be given by any operator $\tilde{\mathcal{O}}_X(\cdot)$ satisfying

$$\tilde{\mathcal{O}}_X(\mathbf{c}) \in \underset{\mathbf{x}\in V}{\text{argmin}} \{\langle \mathbf{c}, \mathbf{x}\rangle\} = \left\{\mathbf{x} \in \{-1, 1\}^3 : x_i c_i = -|c_i|,\ \forall i = 1, \ldots, n\right\}, \quad \forall\, \mathbf{c} \in \mathbb{R}^3. \tag{3.4}$$

Given the vector $\mathbf{c} = (-1,\ 1,\ 3)^T$, we want to find

$$\mathbf{p} \in \underset{\mathbf{y}\in \text{ext}(\mathbf{E}X)}{\text{argmin}} \langle \mathbf{c}, \mathbf{y}\rangle .$$

Using the naive approach, described in (3.3), we obtain a vertex of X by applying the vertex linear oracle $\tilde{\mathcal{O}}_X$ described in (3.4) with parameter $\mathbf{E}^T\mathbf{c} = (0,\ 0,\ 1)$, which may return either one of the vertices B, C, G or H. If vertex C is returned, then its mapping C' does not yield a vertex in $\mathbf{E}X$.

We aim to show that given a vertex linear oracle for X, the ASCG algorithm converges in a linear rate for the more general problem (P). Since in our analysis we do

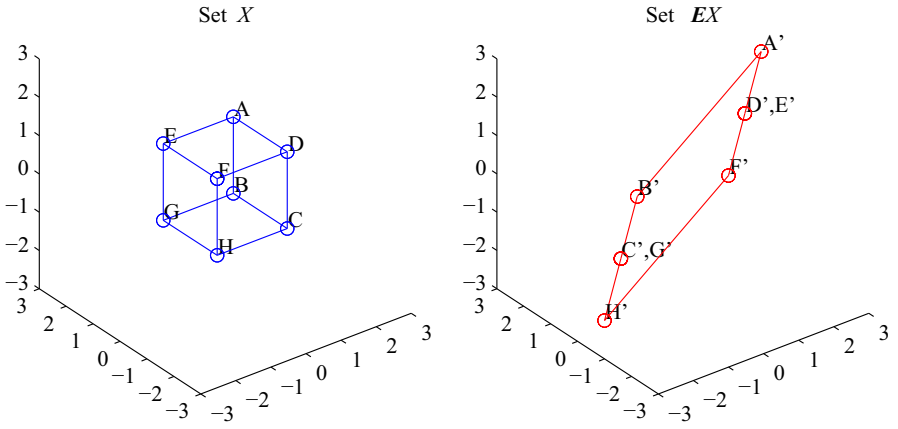

Fig. 1 The sets X and $\mathbf{E}X$

not assume the existence of a linear oracle for $\mathbf{E}X$, but rather a vertex linear oracle for X, the computational cost per iteration related to the linear oracle is independent of the matrix $\mathbf{E}$, and depends only on the geometry of X.

3.2 The ASCG method

We will now present the ASCG algorithm. In the following we denote the vertex set of X as $V = \text{ext}(X)$. Moreover, as part of the ASCG algorithm, at each iteration k the iterate $\mathbf{x}^k$ is represented as a convex combination of points in V. Specifically, $\mathbf{x}^k$ is assumed to have the representation

$$\mathbf{x}^k = \sum_{\mathbf{v}\in V} \mu_{\mathbf{v}}^k \mathbf{v},$$

where $\boldsymbol{\mu}^k \in \Delta_{|V|}$. Let $U^k = \left\{\mathbf{v} \in V : \mu_{\mathbf{v}}^k > 0\right\}$; then U^k and $\left\{\mu_{\mathbf{v}}^k\right\}_{\mathbf{v}\in U^k}$ provide a compact representation of $\mathbf{x}^k$, and $\mathbf{x}^k$ lies in the relative interior of the set $\text{conv}(U^k)$. Throughout the algorithm we update U^k and $\boldsymbol{\mu}^k$ via the vertex representation updating (VRU) scheme. The ASCG method has two types of updates: a *forward step*, used in the classical CG algorithm, where a vertex is added to the representation, and an *away step*, unique to this algorithm, in which the coefficient of one of the vertices used in the representation is reduced or even nullified. Specifically, the away step uses the direction $(\mathbf{x}^k - \mathbf{u}^k)$ where $\mathbf{u}^k \in U^k$ and stepsize $\gamma^k > 0$ so that

$$\begin{aligned}
\mathbf{x}^{k+1} &= \mathbf{x}^k + \gamma^k(\mathbf{x}^k - \mathbf{u}^k) \\
&= \left(\mathbf{x}^k - \mu_{\mathbf{u}^k}^k \mathbf{u}^k\right)(1+\gamma^k) + \left(\mu_{\mathbf{u}^k}^k - \gamma^k\left(1 - \mu_{\mathbf{u}^k}^k\right)\right)\mathbf{u}^k \\
&= \sum_{\mathbf{v}\in U^k/\{\mathbf{u}^k\}} (1+\gamma^k)\mu_{\mathbf{v}}\mathbf{v} + \left(\mu_{\mathbf{u}^k}^k(1+\gamma^k) - \gamma^k\right)\mathbf{u}^k,
\end{aligned}$$

and so $\mu^{k+1}_{\mathbf{u}^k} = \mu^k_{\mathbf{u}^k} - \gamma^k\left(1-\mu^k_{\mathbf{u}^k}\right) < \mu^k_{\mathbf{u}^k}$. Moreover, if $\gamma^k = \frac{\mu^k_{\mathbf{u}^k}}{1-\mu^k_{\mathbf{u}^k}}$, then $\mu^{k+1}_{\mathbf{u}^k}$ is nullified, and consequently, the vertex $\mathbf{u}^k$ is removed from the representation. This vertex removal is referred to as a *drop step*.

The full description of the ASCG algorithm and the VRU scheme is given as follows.

Away Step Conditional Gradient algorithm (ASCG)

Input: A vertex linear oracle $\tilde{\mathcal{O}}_X$

Initialize: $\mathbf{x}^1 \in V$ where $\mu^1_{\mathbf{x}^1} = 1$, $\mu^1_{\mathbf{v}} = 0$ for any $\mathbf{v} \in V/\left\{\mathbf{x}^1\right\}$ and $U^1 = \{\mathbf{x}^1\}$

For $k = 1, 2, \ldots$

1. Compute $\mathbf{p}^k := \tilde{\mathcal{O}}_X(\nabla f(\mathbf{x}^k))$.
2. Compute $\mathbf{u}^k \in \operatorname{argmax}_{\mathbf{v}\in U^k}\langle \nabla f(\mathbf{x}^k), \mathbf{v}\rangle$.
3. If $\langle \nabla f(\mathbf{x}^k), \mathbf{p}^k - \mathbf{x}^k\rangle \le \langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{u}^k\rangle$, then set $\mathbf{d}^k := \mathbf{p}^k - \mathbf{x}^k$ and $\overline{\gamma}^k := 1$. Otherwise, set $\mathbf{d}^k := \mathbf{x}^k - \mathbf{u}^k$ and $\overline{\gamma}^k := \frac{\mu^k_{\mathbf{u}^k}}{1-\mu^k_{\mathbf{u}^k}}$
4. Choose a stepsize $\gamma^k \in [0, \bar{\gamma}^k]$.
5. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \gamma^k \mathbf{d}^k$.
6. Employ the VRU procedure with input $(\mathbf{x}^k, U^k, \boldsymbol{\mu}^k, \mathbf{d}^k, \gamma^k, \mathbf{p}^k, \mathbf{v}^k)$ and obtain an updated representation $(U^{k+1}, \boldsymbol{\mu}^{k+1})$.

The stepsize in the ASCG algorithm can be chosen according to one of the following stepsize selection rules, where $\mathbf{d}^k$ and $\overline{\gamma}^k$ are as defined in the algorithm.

$$\gamma^k \begin{cases} \in \operatorname*{argmin}_{0\le\gamma\le\overline{\gamma}^k} f(\mathbf{x}^k + \gamma \mathbf{d}^k) & \text{Exact line search} \\ = \min\left\{-\frac{\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle}{\rho\|\mathbf{d}^k\|^2}, \overline{\gamma}^k\right\} \in \operatorname*{argmin}_{0\le\gamma\le\overline{\gamma}^k}\left\{\gamma\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle + \gamma^2\frac{\rho}{2}\|\mathbf{d}^k\|^2\right\} & \text{Adaptive [18].} \end{cases} \tag{3.5}$$

Remark 3.1 It is simple to show that under the above two choices of stepsize strategies, the sequence of function values $\{f(\mathbf{x}^k)\}_{k\ge1}$ is nonincreasing.

Since the convergence rate analyses for both of these stepsize options is similar, we chose to conduct a unified analysis for both cases. Following is exact definition of the VRU procedure.

Vertex Representation Updating (VRU) Procedure

Input: $\mathbf{x}^k$ - current point.

$(U^k, \boldsymbol{\mu}^k)$ - vertex representation of $\mathbf{x}^k$,

$\mathbf{d}^k$, γ^k - current direction and stepsize,

$\mathbf{p}^k$, $\mathbf{v}^k$ - candidate vertices.

Output: Updated vertex representation $(U^{k+1}, \boldsymbol{\mu}^{k+1})$ of $\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma^k \mathbf{d}^k$.

If $\mathbf{d}^k = \mathbf{x}^k - \mathbf{u}^k$ (away step) then

1. Update $\mu_{\mathbf{v}}^{k+1} := \mu_{\mathbf{v}}^k (1 + \gamma^k)$ for any $\mathbf{v} \in U^k / \left\{\mathbf{u}^k\right\}$.
2. Update $\mu_{\mathbf{u}^k}^{k+1} := \mu_{\mathbf{u}^k}^k (1 + \gamma^k) - \gamma^k$.
3. If $\mu_{\mathbf{u}^k}^{k+1}$=0 (drop step), then update U^{k+1}:=$U^k / \left\{\mathbf{u}^k\right\}$; otherwise U^{k+1}:=U^k.

Else ($\mathbf{d}^k = \mathbf{p}^k - \mathbf{x}^k$ - forward step)

1. Update $\mu_{\mathbf{v}}^{k+1} := \mu_{\mathbf{v}}^k (1 - \gamma^k)$ for any $\mathbf{v} \in U^k / \left\{\mathbf{p}^k\right\}$.
2. Update $\mu_{\mathbf{p}^k}^{k+1} := \mu_{\mathbf{p}^k}^k (1 - \gamma^k) + \gamma^k$.
3. If $\mu_{\mathbf{p}^k}^{k+1} = 1$, then update $U^{k+1} = \left\{\mathbf{p}^k\right\}$; otherwise update $U^{k+1} := U^k \cup \left\{\mathbf{p}^k\right\}$.

Update $(U^{k+1}, \boldsymbol{\mu}^{k+1}) := \mathcal{R}(U^{k+1}, \boldsymbol{\mu}^{k+1})$ with $\mathcal{R}$ being a representation reduction procedure with constant N.

The VRU scheme uses a representation reduction procedure $\mathcal{R}$ with constant N, which is a procedure that takes a representation $(U, \boldsymbol{\mu})$ of a point $\mathbf{x}$ and replaces it by a representation $(\tilde{U}, \tilde{\boldsymbol{\mu}})$ of $\mathbf{x}$ such that $\tilde{U} \subseteq U$ and $|\tilde{U}| \leq N$. We consider two possible options for the representation reduction procedure:

1. $\mathcal{R}$ is the trivial procedure, meaning it does not change the representation, in which case its constant is $N = |V|$.
2. The procedure $\mathcal{R}$ is some implementation of the Carathéodory theorem [22, Sect. 17], in which case its constant is $N = n+1$. Although this representation does not influence the convergence results, dealing with more compact representations will reduce the storage space needed for the algorithm, as well as the cost of stage 2 of the ASCG algorithm. This is especially significant when the number of vertices is not polynomial in the problem's dimension. A full description of the incremental representation reduction (IRR) scheme, which applies the Carathéodory theorem efficiently in this context, is presented in the appendix.

3.3 Rate of convergence analysis

We will now prove the linear rate of convergence for the ASCG algorithm for problem (P). In the following we use $I(\mathbf{x})$ to denote the *index set of the active constraints at* $\mathbf{x}$,

$$I(\mathbf{x}) = \{i \in \{1, \ldots, n\} : \mathbf{A}_i \mathbf{x} = a_i\}.$$

Similarly, for a given set U, the set of active constraints for all the points in U is defined as

$$I(U) = \{i \in \{1, \ldots, n\} : \mathbf{A}_i \mathbf{v} = a_i, \ \forall \mathbf{v} \in U\} = \bigcap_{\mathbf{v} \in U} I(\mathbf{v}).$$

We present the following technical lemma, which is similar to a result presented by Jaggi and Lacoste-Julien [16].[3] In [16] the proof is based on geometrical considerations, and utilizes the so-called "pyramidal width constant", which is the optimal value of a complicated optimization problem. In contrast, the proof below relies on simple linear programming duality arguments, and in addition, the derived constant Ω_X, which replaces the pyramidal width constant, is computable for many choices of sets X.

Lemma 3.1 *Let $U \subseteq V$ and $\mathbf{c} \in \mathbb{R}^n$. If there exists $\mathbf{z} \in \mathbb{R}^n$ such that $\mathbf{A}_{I(U)}\mathbf{z} \leq 0$ and $\langle \mathbf{c}, \mathbf{z} \rangle > 0$, then*

$$\max_{\mathbf{p} \in V, \mathbf{u} \in U} \langle \mathbf{c}, \mathbf{p} - \mathbf{u} \rangle \geq \frac{\Omega_X}{(n+1)} \frac{\langle \mathbf{c}, \mathbf{z} \rangle}{\|\mathbf{z}\|},$$

where

$$\Omega_X = \min_{\mathbf{v} \in V, i \in \{1, \ldots, m\} : a_i > \mathbf{A}_i \mathbf{v}} \frac{a_i - \mathbf{A}_i \mathbf{v}}{\|\mathbf{A}_i\|}. \tag{3.6}$$

Proof By the fundamental theorem of linear programming [11], we can maximize the function $\langle \mathbf{c}, \mathbf{x} \rangle$ on X instead of on V and get the same optimal value. Similarly, we can minimize the function $\langle \mathbf{c}, \mathbf{y} \rangle$ on $\mathrm{conv}(U)$ instead of on U, and obtain the same optimal value. Therefore,

$$\begin{aligned} \max_{\mathbf{p} \in V, \mathbf{u} \in U} \langle \mathbf{c}, \mathbf{p} - \mathbf{u} \rangle &= \max_{\mathbf{p} \in V} \langle \mathbf{c}, \mathbf{p} \rangle - \min_{\mathbf{u} \in U} \langle \mathbf{c}, \mathbf{u} \rangle \\ &= \max_{\mathbf{x} \in X} \langle \mathbf{c}, \mathbf{x} \rangle - \min_{\mathbf{y} \in \mathrm{conv}(U)} \langle \mathbf{c}, \mathbf{y} \rangle \\ &= \max_{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{a}} \langle \mathbf{c}, \mathbf{x} \rangle + \max_{\mathbf{y} \in \mathrm{conv}(U)} \{-\langle \mathbf{c}, \mathbf{y} \rangle\}. \end{aligned} \tag{3.7}$$

Since X is nonempty and bounded, the problem in $\mathbf{x}$ is feasible and bounded above. Therefore, by strong duality for linear programming,

$$\max_{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{a}} \langle \mathbf{c}, \mathbf{x} \rangle = \min_{\boldsymbol{\eta} \in \mathbb{R}^m_+ : \mathbf{A}^T \boldsymbol{\eta} = \mathbf{c}} \langle \mathbf{a}, \boldsymbol{\eta} \rangle. \tag{3.8}$$

[3] This was done as part of the proof of [16, Lemma 6], and does not appear as a separate lemma.

Plugging (3.8) back into (3.7) we obtain:

$$\begin{aligned}\max_{\mathbf{p}\in V,\mathbf{u}\in U}\langle \mathbf{c},\mathbf{p}-\mathbf{u}\rangle &= \min_{\boldsymbol{\eta}\in\mathbb{R}^m_+:\mathbf{A}^T\boldsymbol{\eta}=\mathbf{c}}\langle \mathbf{a},\boldsymbol{\eta}\rangle + \max_{\mathbf{y}\in\mathrm{conv}(U)}\{-\langle \mathbf{c},\mathbf{y}\rangle\}\\ &= \min_{\boldsymbol{\eta}\in\mathbb{R}^m_+:\mathbf{A}^T\boldsymbol{\eta}=\mathbf{c}}\max_{\mathbf{y}\in\mathrm{conv}(U)}\langle \mathbf{a}-\mathbf{A}\mathbf{y},\boldsymbol{\eta}\rangle\,. \end{aligned} \tag{3.9}$$

Let $\tilde{\mathbf{y}}=\frac{1}{|U|}\sum_{\mathbf{v}\in U}\mathbf{v}$. Then by Carathéodory theorem there exists a set $\tilde{U}\subseteq U$ with cardinality $|\tilde{U}|\le(n+1)$ such that $\tilde{\mathbf{y}}\in\mathrm{conv}(\tilde{U})$. We will prove that $I(\tilde{U})=I(U)$. Since $\tilde{U}\subseteq U$ we have $I(U)\subseteq I(\tilde{U})$. To prove the reverse inclusion, let $i\notin I(U)$, then it follows from the definition of $\tilde{\mathbf{y}}$ that

$$a_i-\mathbf{A}_i\tilde{\mathbf{y}}=\frac{1}{|U|}\sum_{\mathbf{u}\in U}(a_i-\mathbf{A}_i\mathbf{u})\ge\frac{1}{|U|}\max_{\mathbf{u}\in U}(a_i-\mathbf{A}_i\mathbf{u})>0,$$

and so, since $\tilde{\mathbf{y}}\in\mathrm{conv}(\tilde{U})$, there exists $\boldsymbol{\lambda}\ge\mathbf{0}$ where $\mathbf{1}^T\boldsymbol{\lambda}=1$ such that $\tilde{\mathbf{y}}=\sum_{\mathbf{u}\in\tilde{U}}\lambda_{\mathbf{u}}\mathbf{u}$; consequently,

$$\max_{\mathbf{u}\in\tilde{U}}(a_i-\mathbf{A}_i\mathbf{u})\ge\sum_{\mathbf{u}\in\tilde{U}}\lambda_{\mathbf{u}}(a_i-\mathbf{A}_i\mathbf{u})=a_i-\mathbf{A}_i\tilde{\mathbf{y}}>0$$

which implies that $i\notin I(\tilde{U})$. Thus, $I(\tilde{U})\subseteq I(U)$, establishing the fact that $I(\tilde{U})=I(U)$. We define $\bar{\mathbf{y}}=\frac{1}{|\tilde{U}|}\sum_{\mathbf{v}\in\tilde{U}}\mathbf{v}$. Since $\bar{\mathbf{y}}\in\mathrm{conv}(\tilde{U})\subseteq\mathrm{conv}(U)$, we have that

$$\max_{\mathbf{y}\in\mathrm{conv}(U)}\langle \mathbf{a}-\mathbf{A}\mathbf{y},\boldsymbol{\eta}\rangle\ge\langle \mathbf{a}-\mathbf{A}\bar{\mathbf{y}},\boldsymbol{\eta}\rangle$$

for any value of $\boldsymbol{\eta}$, and therefore,

$$\min_{\boldsymbol{\eta}\in\mathbb{R}^m_+:\mathbf{A}^T\boldsymbol{\eta}=\mathbf{c}}\max_{\mathbf{y}\in\mathrm{conv}(U)}\langle \mathbf{a}-\mathbf{A}\mathbf{y},\boldsymbol{\eta}\rangle\ge\min_{\boldsymbol{\eta}\in\mathbb{R}^m_+:\mathbf{A}^T\boldsymbol{\eta}=\mathbf{c}}\langle \mathbf{a}-\mathbf{A}\bar{\mathbf{y}},\boldsymbol{\eta}\rangle\,. \tag{3.10}$$

Using strong duality on the RHS of (3.10), we obtain that

$$\min_{\boldsymbol{\eta}\in\mathbb{R}^m_+:\mathbf{A}^T\boldsymbol{\eta}=\mathbf{c}}\langle \mathbf{a}-\mathbf{A}\bar{\mathbf{y}},\boldsymbol{\eta}\rangle=\max_{\mathbf{x}}\{\langle \mathbf{c},\mathbf{x}\rangle:\mathbf{A}\mathbf{x}\le(\mathbf{a}-\mathbf{A}\bar{\mathbf{y}})\}\,. \tag{3.11}$$

Denote $J=I(\tilde{U})$ and $\overline{J}=\{1,\ldots,m\}/J$. From the definition of $I(\tilde{U})$, it follows that

$$\mathbf{a}_J-\mathbf{A}_J\mathbf{v}=\mathbf{0} \tag{3.12}$$

for all $\mathbf{v}\in\tilde{U}$, and that for any $i\in\overline{J}$ there exists at least one vertex $\mathbf{v}\in\tilde{U}$ such that $a_i-\mathbf{A}_i\mathbf{v}>0$, and hence,

$$a_i - \mathbf{A}_i\mathbf{v} \geq \min_{\mathbf{u}\in V:\mathbf{a}_i>\mathbf{A}_i\mathbf{u}} (a_i - \mathbf{A}_i\mathbf{u}),$$

which in particular implies that

$$\frac{1}{\|\mathbf{A}_i\|}\sum_{\mathbf{v}\in\tilde{U}}(a_i - \mathbf{A}_i\mathbf{v}) \geq \frac{1}{\|\mathbf{A}_i\|}\min_{\mathbf{u}\in V:\mathbf{a}_i>\mathbf{A}_i\mathbf{u}} (a_i - \mathbf{A}_i\mathbf{u}) \geq \Omega_X. \tag{3.13}$$

We denote $\overline{\mathbf{A}_i} = \mathbf{A}_i/\|\mathbf{A}_i\|$ and $\overline{a}_i = a_i/\|\mathbf{A}_i\|$, and similarly, $\bar{\mathbf{a}}_J$ denotes the vector comprising the components $\overline{a}_i, i \in J$ and $\overline{\mathbf{A}}_J$ stands for the matrix whose rows are $\overline{\mathbf{A}}_i, i \in J$. From the choice of $\bar{\mathbf{y}}$, we can conclude from (3.12) and (3.13) that

$$\begin{aligned} &\bar{\mathbf{a}}_J - \overline{\mathbf{A}}_J\bar{\mathbf{y}} = \mathbf{0}, \\ &\bar{\mathbf{a}}_{\overline{J}} - \overline{\mathbf{A}}_{\overline{J}}\bar{\mathbf{y}} = \frac{1}{|\tilde{U}|}\sum_{\mathbf{v}\in\tilde{U}}(\bar{\mathbf{a}}_{\overline{J}} - \overline{\mathbf{A}}_{\overline{J}}\mathbf{v}) \geq \mathbf{1}\frac{\Omega_X}{|\tilde{U}|}. \end{aligned} \tag{3.14}$$

Therefore, replacing the RHS of the set of inequalities $\mathbf{A}\mathbf{x} \leq (\mathbf{a} - \mathbf{A}\bar{\mathbf{y}})$ in (3.11) by the bounds given in (3.14), we obtain that

$$\max_{\mathbf{x}}\{\langle \mathbf{c}, \mathbf{x}\rangle : \mathbf{A}\mathbf{x} \leq (\mathbf{a} - \mathbf{A}\bar{\mathbf{y}})\} \geq \max_{\mathbf{x}}\left\{\langle \mathbf{c}, \mathbf{x}\rangle : \overline{\mathbf{A}}_J\mathbf{x} \leq \mathbf{0},\ \overline{\mathbf{A}}_{\overline{J}}\mathbf{x} \leq \mathbf{1}\frac{\Omega_X}{|\tilde{U}|}\right\}. \tag{3.15}$$

Combining (3.9),(3.10), (3.11) and (3.15) it follows that

$$\max_{\mathbf{p}\in V,\mathbf{u}\in U}\langle \mathbf{c}, \mathbf{p} - \mathbf{u}\rangle \geq Z^*, \tag{3.16}$$

where

$$Z^* = \max_{\mathbf{x}}\left\{\langle \mathbf{c}, \mathbf{x}\rangle : \overline{\mathbf{A}}_J\mathbf{x} \leq \mathbf{0},\ \overline{\mathbf{A}}_{\overline{J}}\mathbf{x} \leq \mathbf{1}\frac{\Omega_X}{|\tilde{U}|}\right\}. \tag{3.17}$$

We will now show that it is not possible for $\mathbf{z}$ to satisfy $\mathbf{A}_{\overline{J}}\mathbf{z} \leq 0$ [(recall that $J = I(U) = I(\tilde{U})$]. Suppose by contradiction that $\mathbf{z}$ satisfies does satisfy $\mathbf{A}_{\overline{J}}\mathbf{z} \leq 0$. Then $\mathbf{x}_\alpha = \alpha\mathbf{z}$ is a feasible solution of problem (3.17) for any $\alpha > 0$, and since $\langle \mathbf{c}, \mathbf{z}\rangle > 0$ we obtain that $\langle \mathbf{c}, \mathbf{x}_\alpha\rangle \to \infty$ as $\alpha \to \infty$, and thus $Z^* = \infty$. However, since V contains a finite number of points, the LHS of (3.16) is bounded from above, and so $Z^* < \infty$ in contradiction. Therefore, there exists $i \in \overline{J}$ such that $\mathbf{A}_i\mathbf{z} > 0$. Since $\mathbf{z} \neq \mathbf{0}$, the vector $\bar{\mathbf{x}} = \frac{\mathbf{z}}{\|\mathbf{z}\|}\frac{\Omega_X}{|\tilde{U}|}$ is well defined. Moreover, $\bar{\mathbf{x}}$ satisfies

$$\overline{\mathbf{A}}_J\bar{\mathbf{x}} = \frac{\Omega_X}{\|\mathbf{z}\|\,|\tilde{U}|}\overline{\mathbf{A}}_J\mathbf{z} \leq 0, \tag{3.18}$$

and (recalling that $\|\overline{\mathbf{A}}_i\| = 1$)

$$\overline{\mathbf{A}}_i\bar{\mathbf{x}} = \overline{\mathbf{A}}_i\mathbf{z}\frac{\Omega_X}{|\tilde{U}|\,\|\mathbf{z}\|} \leq \left\|\overline{\mathbf{A}}_i\right\|\|\mathbf{z}\|\frac{\Omega_X}{|\tilde{U}|\,\|\mathbf{z}\|} = \frac{\Omega_X}{|\tilde{U}|}, \quad \forall i \in \overline{J}, \tag{3.19}$$

where the inequality follows from the Cauchy-Schwarz inequality. Consequently, (3.18) and (3.19) imply that $\overline{\mathbf{x}}$ is a feasible solution for problem (3.17). Therefore, $Z^* \geq \langle \mathbf{c}, \overline{\mathbf{x}} \rangle$, which by (3.16) yields

$$\max_{\mathbf{p}\in V, \mathbf{u}\in U} \langle \mathbf{c}, \mathbf{p}-\mathbf{u} \rangle \geq \langle \mathbf{c}, \overline{\mathbf{x}} \rangle = \frac{\Omega_X}{|\tilde{U}|} \frac{\langle \mathbf{c}, \mathbf{z} \rangle}{\|\mathbf{z}\|} \geq \frac{\Omega_X}{n+1} \frac{\langle \mathbf{c}, \mathbf{z} \rangle}{\|\mathbf{z}\|},$$

where the last inequality utilizes the fact that $|\tilde{U}| \leq n+1$. □

The constant Ω_X represents a normalized minimal distance between the hyperplanes that contain facets of X and the vertices of X which do not lie on those hyperplanes. We will refer to Ω_X as *the vertex-facet distance of* X. Examples for the derivation of Ω_X for some simple polyhedral sets can be found in Sect. 3.4.

The following lemma is a technical result stating that the active constraints at a given point are the same as the active constraints of the set of vertices in its compact representation.

Lemma 3.2 *Let* $\mathbf{x} \in X$ *and the set* $U \subseteq V$ *satisfy* $\mathbf{x} = \sum_{\mathbf{v}\in U} \mu_{\mathbf{v}} \mathbf{v}$, *where* $\boldsymbol{\mu} \in \Delta^+_{|U|}$. *Then* $I(\mathbf{x}) = I(U)$.

Proof It is trivially true that $I(U) \subseteq I(\mathbf{x})$ since $\mathbf{x}$ is a convex combination of points in the affine space defined by $\left\{\mathbf{y} : \mathbf{A}_{I(U)}\mathbf{y} = \mathbf{a}_{I(U)}\right\}$. We will prove that $I(\mathbf{x}) \subseteq I(U)$. Any $\mathbf{v} \in U \subseteq X$ satisfies $\mathbf{A}_{I(\mathbf{x})}\mathbf{v} \leq \mathbf{a}_{I(\mathbf{x})}$. Assume to the contrary, that there exists $i \in I(\mathbf{x})$ such that some $\mathbf{u} \in U$ satisfies $\mathbf{A}_i\mathbf{u} < a_i$. Since $\mu_{\mathbf{u}} > 0$ and $\sum_{\mathbf{v}\in U} \mu_{\mathbf{v}} = 1$, it follows that

$$\mathbf{A}_i\mathbf{x} = \sum_{\mathbf{v}\in U} \mu_{\mathbf{v}} \mathbf{A}_i\mathbf{v} < \sum_{\mathbf{v}\in U} \mu_{\mathbf{v}} a_i = a_i,$$

in contradiction to the assumption that $i \in I(\mathbf{x})$. □

Corollary 3.1 *For any* $\mathbf{x} \in X/X^*$ *which can be represented as* $\mathbf{x} = \sum_{\mathbf{v}\in U} \mu_{\mathbf{v}} \mathbf{v}$ *for some* $\boldsymbol{\mu} \in \Delta^+_{|U|}$ *and* $U \subseteq V$, *it holds that,*

$$\max_{\mathbf{u}\in U, \mathbf{p}\in V} \langle \nabla f(\mathbf{x}), \mathbf{u}-\mathbf{p} \rangle \geq \frac{\Omega_X}{(n+1)} \max_{\mathbf{x}^*\in X^*} \frac{\langle \nabla f(\mathbf{x}), \mathbf{x}-\mathbf{x}^* \rangle}{\|\mathbf{x}-\mathbf{x}^*\|}.$$

Proof For any $\mathbf{x} \in X/X^*$ define $\mathbf{c} = -\nabla f(\mathbf{x})$. It follows from Lemma 3.2 that $I(U) = I(\mathbf{x})$. For any $\mathbf{x}^* \in X^*$, the vector $\mathbf{z} = \mathbf{x}^* - \mathbf{x}$ satisfies

$$\mathbf{A}_{I(U)}\mathbf{z} = \mathbf{A}_{I(\mathbf{x})}\mathbf{z} = \mathbf{A}_{I(\mathbf{x})}\mathbf{x}^* - \mathbf{A}_{I(\mathbf{x})}\mathbf{x} \leq \mathbf{a}_{I(\mathbf{x})} - \mathbf{a}_{I(\mathbf{x})} = \mathbf{0},$$

and, from the convexity of f, as well as the optimality of $\mathbf{x}^*$, $\langle \mathbf{c}, \mathbf{z} \rangle = -\langle \nabla f(\mathbf{x}) \mathbf{x}^* - \mathbf{x} \rangle \geq f(\mathbf{x}) - f(\mathbf{x}^*) > 0$. Therefore, invoking Lemma 3.1 achieves the desired result. □

We now present the main theorem of this section, which establishes the linear rate of convergence of the ASCG method for solving problem (P). This theorem is an extension of [16, Thorem 7], and the proof follows the same general arguments, while incorporating the use of the error bound from Lemma 2.5 and the new constant Ω_X.

Theorem 3.1 *Let $\{\mathbf{x}^k\}_{k\geq 1}$ be the sequence generated by the ASCG algorithm for solving problem (P), and let f^* be the optimal value of problem (P). Then for any $k \geq 1$*

$$f(\mathbf{x}^k) - f^* \leq C(1-\alpha^\dagger)^{(k-1)/2}, \tag{3.20}$$

where

$$\alpha^\dagger = \min\left\{\frac{(\Omega_X)^2}{8\rho\kappa D^2(n+1)^2}, \frac{1}{2}\right\}, \tag{3.21}$$

$\kappa = \theta^2\left(\|\mathbf{b}\| D + 3GD_{\mathbf{E}} + \frac{2(G^2+1)}{\sigma_g}\right)$ *with θ being the Hoffman constant associated with the matrix $\left[\mathbf{A}^T, \mathbf{E}^T, \mathbf{b}\right]^T$, $C = GD_{\mathbf{E}} + \|\mathbf{b}\| D$, and Ω_X is the vertex-facet distance of X given in (3.6).*

Proof For each k we will denote the stepsize generated by exact line search as γ_e^k and the adaptive stepsize as γ_a^k. Then

$$f\left(\mathbf{x}^k + \gamma_e^k \mathbf{d}^k\right) \leq f(\mathbf{x}^{k+1}) \leq f\left(\mathbf{x}^k + \gamma_a^k \mathbf{d}^k\right). \tag{3.22}$$

From Lemma 2.1 (the descent lemma), we have that

$$f\left(\mathbf{x}^k + \gamma_a^k \mathbf{d}^k\right) \leq f(\mathbf{x}^k) + \gamma_a^k \langle \nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle + \frac{\left(\gamma_a^k\right)^2 \rho}{2}\|\mathbf{d}^k\|^2. \tag{3.23}$$

Assuming that $\mathbf{x}^k \notin X^*$, then for any $\mathbf{x}^* \in X^*$ we have that

$$\begin{aligned}
\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle &= \min\left\{\langle \nabla f(\mathbf{x}^k), \mathbf{p}^k - \mathbf{x}^k\rangle, \langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{u}^k\rangle\right\} \\
&\leq \langle \nabla f(\mathbf{x}^k), \mathbf{p}^k - \mathbf{x}^k\rangle \\
&\leq \langle \nabla f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^k\rangle \\
&\leq f^* - f(\mathbf{x}^k),
\end{aligned} \tag{3.24}$$

where the first equality is derived from the algorithm's specific choice of $\mathbf{d}^k$, the third line follows from the fact that $\mathbf{p}^k = \tilde{\mathcal{O}}_X(\nabla f(\mathbf{x}^k))$, and the fourth line follows from the convexity of f. In particular, $\mathbf{d}^k \neq \mathbf{0}$, and by (3.5), γ_a^k is equal to

$$\gamma_a^k = \min\left\{-\frac{\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle}{\rho\|\mathbf{d}^k\|^2}, \overline{\gamma}^k\right\}. \tag{3.25}$$

We now separate the analysis to three cases: (a) $\mathbf{d}^k = \mathbf{p}^k - \mathbf{x}^k$ and $\gamma_a^k = \overline{\gamma}^k$, (b) $\mathbf{d}^k = \mathbf{x}^k - \mathbf{u}^k$ and $\gamma_a^k = \overline{\gamma}^k$, and (c) $\gamma_a^k < \overline{\gamma}^k$.

In cases (a) and (b), it follows from (3.25) that

$$\overline{\gamma}^k \rho \|\mathbf{d}^k\|^2 \leq -\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle. \tag{3.26}$$

Using inequalities (3.22), (3.23) and (3.26), we obtain

$$\begin{aligned} f(\mathbf{x}^{k+1}) &\leq f(\mathbf{x}^k) + \gamma_a^k \langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle + \frac{(\gamma_a^k)^2 \rho}{2} \|\mathbf{d}^k\|^2 \\ &\leq f(\mathbf{x}^k) + \frac{\overline{\gamma}^k}{2} \langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle. \end{aligned}$$

Subtracting f^* from both sides of the inequality and using (3.24), we have that

$$\begin{aligned} f(\mathbf{x}^{k+1}) - f^* &\leq f(\mathbf{x}^k) - f^* + \frac{\overline{\gamma}^k}{2} \langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle \\ &\leq (f(\mathbf{x}^k) - f^*) \left(1 - \frac{\overline{\gamma}^k}{2} \right). \end{aligned} \tag{3.27}$$

In case (a), $\overline{\gamma}^k = 1$, and hence

$$f(\mathbf{x}^{k+1}) - f^* \leq \frac{f(\mathbf{x}^k) - f^*}{2}. \tag{3.28}$$

In case (b), we have no positive lower bound on $\overline{\gamma}^k$, and therefore we can only conclude, by the nonnegativity of $\overline{\gamma}^k$, that

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

However, case (b) is a drop step, meaning in particular that $|U^{k+1}| \leq |U^k| - 1$, since before applying the representation reduction procedure $\mathcal{R}$, we eliminate one of the vertices in the representation of $\mathbf{x}^k$. Denoting the number of drop steps until iteration k as s^k, and the number of forward steps until iteration k as l^k, it follows from the algorithm's definition that $l^k + s^k \leq k - 1$ (at each iteration we add a vertex, remove a vertex, or neither) and $s^k \leq l^k$ (the number of removed vertices can not exceed the number of added vertices), and therefore $s^k \leq (k-1)/2$.

We arrive to case (c). In this case, (3.25) implies

$$\gamma_a^k = -\frac{\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle}{\rho \|\mathbf{d}^k\|^2},$$

which combined with (3.22) and (3.23) results in

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) + \gamma_a^k \langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle + \frac{\left(\gamma_a^k\right)^2 \rho}{2} \|\mathbf{d}^k\|^2 = f(\mathbf{x}^k) - \frac{\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle^2}{2\rho \|\mathbf{d}^k\|^2}. \tag{3.29}$$

From the algorithm's specific choice of $\mathbf{d}^k$, we obtain that

$$\begin{aligned} 0 \geq \langle \nabla f(\mathbf{x}^k), \mathbf{p}^k - \mathbf{u}^k \rangle &= \langle \nabla f(\mathbf{x}^k), \mathbf{p}^k - \mathbf{x}^k \rangle + \langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{u}^k \rangle \\ &\geq 2\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle. \end{aligned} \tag{3.30}$$

Applying the bound in (3.30) and the inequality $\left\|\mathbf{d}^k\right\| \leq D$ to (3.29), it follows that

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) - \frac{\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle^2}{2\rho \|\mathbf{d}^k\|^2} \leq f(\mathbf{x}^k) - \frac{\langle \nabla f(\mathbf{x}^k), \mathbf{p}^k - \mathbf{u}^k \rangle^2}{8\rho D^2}. \tag{3.31}$$

By the definitions of $\mathbf{u}^k$ and $\mathbf{p}^k$, Corollary 3.1 implies that for any $\mathbf{x}^* \in X^*$,

$$\langle \nabla f(\mathbf{x}^k), \mathbf{u}^k - \mathbf{p}^k \rangle = \max_{\mathbf{p} \in V, \mathbf{u} \in U^k} \langle \nabla f(\mathbf{x}^k), \mathbf{u} - \mathbf{p} \rangle \geq \frac{\Omega_X}{n+1} \frac{\langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^* \rangle}{\|\mathbf{x}^k - \mathbf{x}^*\|}. \tag{3.32}$$

Lemma 2.5 implies that there exists $\mathbf{x}^* \in X^*$ such that $\|\mathbf{x}^k - \mathbf{x}^*\|^2 \leq \kappa(f(\mathbf{x}^k) - f^*)$, which combined with convexity of f, bounds (3.32) from below as follows:

$$\begin{aligned} \langle \nabla f(\mathbf{x}^k), \mathbf{u}^k - \mathbf{p}^k \rangle^2 &\geq \left(\frac{\Omega_X}{n+1}\right)^2 \frac{\left\langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^* \right\rangle^2}{\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2} \\ &\geq \left(\frac{\Omega_X}{n+1}\right)^2 \frac{(f(\mathbf{x}^k) - f(\mathbf{x}^*))^2}{\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2} \\ &\geq \left(\frac{\Omega_X}{n+1}\right)^2 \frac{(f(\mathbf{x}^k) - f^*)^2}{\kappa(f(\mathbf{x}^k) - f^*)} \\ &= \frac{(\Omega_X)^2}{(n+1)^2 \kappa} (f(\mathbf{x}^k) - f^*), \end{aligned}$$

which along with (3.31) yields

$$\begin{aligned} f(\mathbf{x}^{k+1}) - f^* &\leq f(\mathbf{x}^k) - f^* - \frac{\langle \nabla f(\mathbf{x}^k), \mathbf{u}^k - \mathbf{p}^k \rangle^2}{8\rho D^2} \\ &\leq (f(\mathbf{x}^k) - f^*) \left(1 - \frac{(\Omega_X)^2}{8\rho\kappa D^2 (n+1)^2}\right) \end{aligned} \tag{3.33}$$

Therefore, if either of the cases (a) or (c) occurs, then by (3.28) and (3.33), it follows that

$$f(\mathbf{x}^{k+1}) - f^* \leq (1 - \alpha^\dagger)(f(\mathbf{x}^k) - f^*), \tag{3.34}$$

where $\alpha^\dagger$ is defined in (3.21). We can therefore conclude from cases (a)–(c) that until iteration k we have at least $\frac{k-1}{2}$ iterations for which (3.34) holds, and therefore

$$f(\mathbf{x}^k) - f^* \leq (f(\mathbf{x}^1) - f^*)(1 - \alpha^\dagger)^{(k-1)/2}. \tag{3.35}$$

Applying Lemma 2.4 for $\mathbf{x} = \mathbf{x}^1$ we obtain $f(\mathbf{x}^1) - f^* \leq C$, and the desired result (3.20) follows. □

Remark 3.2 Notice that while the Hoffman constant enables us to conduct the analysis despite the existence of the linear term in the objective function, the convergence result that follows is no longer affine invariant. This is a direct result from Lemma 2.5, which is needed in order to deal with the linear part of the objective.

3.4 Examples of computing the vertex-facet distance Ω_X

In this section, we demonstrate how to compute the vertex-facet distance constant Ω_X for a few simple polyhedral sets. We consider three sets: the unit simplex, the ℓ_1 ball and the ℓ_∞ ball. We first describe each of the sets as a system of linear inequalities of the form $X = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{a}\}$. Then, given the parameters $\mathbf{A}$ and $\mathbf{a}$, as well as the vertex set V, Ω_X can be computed by its definition, given by (3.6). For the unit simplex and the ℓ_∞ ball we also compare the constant that arises from our analysis and the pyramidal width constant presented in [16], with values presented more recently in [17], and discuss both tightness and ease of computation of each.

The ℓ_∞ ball. The ℓ_∞ ball is represented by

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \in \mathbb{R}^{2n \times n}, \ \mathbf{a} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \in \mathbb{R}^{2n}. \tag{3.36}$$

The set of extreme points is given by $V = \{-1, 1\}^n$, which in particular implies that $|V| = 2^n$. Therefore, for large-scale problems, using the representation reduction procedure, which is based on Carathéodory theorem, is crucial in order to obtain a practical implementation.

From the definition of $\mathbf{A}$ and V, it follows that

$$\|\mathbf{A}_i\| = \|-\mathbf{A}_{n+i}\| = \|\mathbf{e}_i\| = 1, \quad i = 1, \ldots, n$$

and so

$$\Omega_X = \min_{i \in \{1,\ldots,n\},\ \mathbf{v} \in \{-1,1\}^n : \langle \mathbf{e}_i, \mathbf{v} \rangle < 1} (1 - \langle \mathbf{e}_i, \mathbf{v} \rangle) = 2.$$

Comparing this result with the pyramidal width of the cube, which by [17] is equal to $2/\sqrt{n}$ for the given two-unit cube, we must recall that the pyramidal width is compared to the $\Omega_X/n = 2/n$ since the latter is actually a lower bound on the former. Thus we lose a factor of $\sqrt{n}$ using the looser bound Ω_X. However, looking at the proof in [17], the calculation of the pyramidal width relies strongly on geometrical considerations (e.g., symmetry of the cube), and the properties of the algorithm (the forward step being the maximizer over the chosen direction), while the computation of Ω_X is straightforward.

The unit simplex. The unit simplex Δ_n can be represented by

$$\mathbf{A} = \begin{bmatrix} -\mathbf{I}_{n\times n} \\ \mathbf{1}_n^T \\ -\mathbf{1}_n^T \end{bmatrix} \in \mathbb{R}^{(n+2)\times n}, \ \mathbf{a} = \begin{bmatrix} \mathbf{0}_n \\ 1 \\ 1 \end{bmatrix} \in \mathbb{R}^{(n+2)}. \tag{3.37}$$

The set of extreme points is given by $V = \{\mathbf{e}_i\}_{i=1}^n$. Notice that since there are only n extreme points which are all affinely independent, using a rank reduction procedure which implements the Carathéodory theorem is the same as applying the trivial procedure that does not change the representation. In order to calculate Ω_X, we first note that $I(V) = \{n+1, n+2\}$, and therefore

$$\|\mathbf{A}_i\| = \|\mathbf{e}_i\| = 1, \quad i = 1, \dots n$$

and

$$\Omega_X = \min_{\mathbf{v}\in\{\mathbf{e}_j\}_{j=1}^n, i\in\{1,\dots,n\}:-\langle\mathbf{e}_i,\mathbf{v}\rangle<0} \langle \mathbf{e}_i, \mathbf{v}\rangle = \min_{i\in\{1,\dots,n\}} \|\mathbf{e}_i\|^2 = 1.$$

Comparing this result with the pyramidal width of the unit simplex, which by [17] is given by $2/\sqrt{n}$, while we show $\Omega_X/n = 1/n$ and again we lose a factor of $\sqrt{n}$ using the looser bound Ω_X. The proof in [17] relies on a geometrical result discussing the width of the simplex, which is not easily derived, while the computation of Ω_X is a direct result of V being the standard basis in $\mathbb{R}^n$.

The ℓ_1 ball. The ℓ_1 ball is given by the set

$$X = \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n |x_i| \le 1 \right\} = \left\{ \mathbf{x} \in \mathbb{R}^n : \langle \mathbf{w}, \mathbf{x}\rangle \le 1, \forall \mathbf{w} \in \{-1, 1\}^n \right\}.$$

Therefore $\mathbf{a} = \mathbf{1} \in \mathbb{R}^{2^n}$ and each row of the matrix $\mathbf{A} \in \mathbb{R}^{2^n\times n}$ is a vector in $\{-1, 1\}^n$. The set of extreme points is given by $V = \{\mathbf{e}_i\}_{i=1}^n \bigcup \{-\mathbf{e}_i\}_{i=1}^n$, and therefore has cardinality of $|V| = 2n$.

Finally, we have that

$$\|\mathbf{A}_i\| = \sqrt{n}, \quad i \in \left\{1, \dots, 2^n\right\},$$

and so

$$\begin{aligned} \Omega_X &= \frac{1}{\sqrt{n}} \min_{\mathbf{v}\in V, \mathbf{w}\in\{-1,1\}^n:\langle\mathbf{v},\mathbf{w}\rangle<1} (1 - \langle \mathbf{v}, \mathbf{w}\rangle) \\ &= \frac{1}{\sqrt{n}} \min_{i\in\{1,\dots,n\},\ \mathbf{w}\in\{-1,1\}^n:\langle\mathbf{e}_i,\mathbf{w}\rangle<1} (1 - \langle \mathbf{e}_i, \mathbf{w}\rangle) \\ &= \frac{1}{\sqrt{n}} \min_{\mathbf{w}\in\{-1,1\}^n} (1 + |w_i|) = \frac{2}{\sqrt{n}}. \end{aligned}$$

The pyramidal width of the ℓ_1 ball is obviously smaller than that of the simplex, and it is difficult to calculate via the approach suggested in [17].

A recent work of Peña and Rodríguez [20], which appeared after the submition of this paper, showed that the pyramidal width is equal to a quantity referred to as the *facial distance* of the polytope. This quantity is actually a refinement of the vertex-facet distance, since it calculates the distance between any face of the polytope and the polytope generated by the convex-hull of the vertices not contained in that face. The vertex-facet distance is the distance from a vertex to the hyperplane containing a single polytope facet, disregarding the additional information about the polytope. Therefore, it follows that the bound generated by the vertex-facet distance can be arbitrarily smaller than the facial distance. However, the combinatorial cost of computing the vertex-facet distance is generally linearly dependent on the the number of vertices and facets, while the facial distance generally requires a computation which is exponentially dependent on the number of vertices (equivalent to the number of polytope faces).

Acknowledgements The research of Amir Beck was partially supported by the Israel Science Foundation grant 1821/16

Appendix: Incremental representation reduction using the Carathéodory theorem

In this section we will show a way to efficiently and incrementally implement the constructive proof of Carathéodory theorem, as part of the VRU scheme, at each iteration of the ASCG algorithm. We note that this reduction procedure does not have to be employed, and instead the trivial procedure, which does not change the representation can be used. In that case, the upper bound on the number of extreme points in the representation is just the number of extreme points of the feasible set X.

The implementation described in this section will allow maintaining a vertex representation set U^k, with cardinality of at most $n+1$, at a computational cost of $O(n^2)$ operations per iteration. For this purpose, we assume that at the beginning of iteration k, $\mathbf{x}^k$ has a representation with vertex set $U^k = \left\{\mathbf{v}^1, \ldots, \mathbf{v}^L\right\} \subseteq V$, such that the vectors in the set are affinely independent. Moreover, we assume that at the beginning of iteration k, we have at our disposal two matrices $\mathbf{T}^k \in \mathbb{R}^{n\times n}$ and $\mathbf{W}^k \in \mathbb{R}^{n\times(L-1)}$. We define $\mathbf{V}^k \in \mathbb{R}^{n\times(L-1)}$ to be the matrix whose ith column is the vector $\mathbf{w}^i = \mathbf{v}^{i+1} - \mathbf{v}^1$ for $i = 1, \ldots, L-1$, where $\mathbf{v}^1$ is called the reference vertex. The matrix $\mathbf{T}^k$ is a product of elementary matrices, which ensures that the matrix $\mathbf{W}^k = \mathbf{T}^k\mathbf{V}^k$ is in row echelon form. The implementation does not require to save the matrix $\mathbf{V}^k$, and so at each iteration, only the matrices $\mathbf{T}^k$ and $\mathbf{W}^k$ are updated.

Let U^{k+1} be the vertex set and let $\boldsymbol{\mu}^{k+1}$ be the coefficients vector at the end of iteration k, before applying the rank reduction procedure. Updating the matrices $\mathbf{W}^{k+1}$ and $\mathbf{T}^{k+1}$, as well as U^{k+1} and $\boldsymbol{\mu}^{k+1}$, is done according to the following *Incremental Representation Reduction* scheme, which is partially based on the proof of Carathéodory theorem presented in [22, Sect. 17].

Incremental Representation Reduction (IRR)

Input: Representation $(U^{k+1}, \boldsymbol{\mu}^{k+1})$ of point $\mathbf{x}^{k+1}$, set $U^k = \left\{\mathbf{v}^1, \ldots, \mathbf{v}^L\right\}$ of affinely independent vectors, and matrices $\mathbf{T}^k \in \mathbb{R}^{n\times n}$ and $\mathbf{W}^k \in \mathbb{R}^{n\times(L-1)}$.

Output: Updated representation $(U^{k+1}, \boldsymbol{\mu}^{k+1})$ of $\mathbf{x}^{k+1}$, and matrices $\mathbf{T}^{k+1} \in \mathbb{R}^{n\times n}$ and $\mathbf{W}^{k+1} \in \mathbb{R}^{n\times(|U_{k+1}|-1)}$.

1. Set $L := |U^k|$.
2. Update $\mathbf{T}^{k+1} := \mathbf{T}^k$.
3. If $|U^{k+1}| = 1$, then set the matrix $\mathbf{W}^{k+1}$ to be empty and $\mathbf{T}^{k+1} := \mathbf{I}$.
4. Else, if $|U^{k+1}| = L$, then set $\mathbf{W}^{k+1} := \mathbf{W}^k$.
5. Else, if $|U^{k+1}| = L - 1 > 1$ (drop step), then
   (a) Find $i^* \in \{1, \ldots, L\}$ such that $\mathbf{v}^{i^*} \in U^k / U^{k+1}$.
   (b) If $i^* = 1$ (the reference vertex was removed), then remove the first column of $\mathbf{W}^k$ and change reference vertex to $\mathbf{v}^2$, using the update formula

$$\mathbf{W}^{k+1} := \mathbf{W}^k \left[\mathbf{0}\ \mathbf{I}_{(L-2)\times(L-2)}\right]^T + \mathbf{T}^k(\mathbf{v}^1 - \mathbf{v}^2)\mathbf{1}^T,$$

   where $\mathbf{1}, \mathbf{0} \in \mathbb{R}^{L-2}$.
   (c) Else (a non-reference vertex was removed), remove column $i^* - 1$ from $\mathbf{W}^{k+1}$.
6. Else, if $|U^{k+1}| = L + 1$ (forward step), then
   (a) Find $\mathbf{v}^{L+1} \in U^{k+1} / U^k$.
   (b) Compute $\mathbf{w}^L := \mathbf{v}^{L+1} - \mathbf{v}^1$.
   (c) Update the matrix $\mathbf{W}^{k+1} := [\mathbf{W}^k, \mathbf{T}^k\mathbf{w}^L]$.
   (d) Compute M - the row rank of $\mathbf{W}^{k+1}$.
   (e) If $L > M$, then
      i. Find a solution $\boldsymbol{\lambda}$ of the following system

$$\mathbf{W}^{k+1}\boldsymbol{\lambda} = \mathbf{0},\ \lambda_L = -1.$$

      ii. Set the vector $\tilde{\boldsymbol{\lambda}} \in \mathbb{R}^{L+1}$ to be

$$\tilde{\lambda}_i := \begin{cases} -\sum_{i=2}^{L+1} \lambda_{i-1} & i = 1 \\ \lambda_{i-1} & i = 2, \ldots, L+1 \end{cases}.$$

iii. Compute $\overline{\alpha} := \min_{i:\tilde{\lambda}_i<0} -\frac{\mu_i^k}{\tilde{\lambda}_i}$ and $\underline{\alpha} := \max_{i:\tilde{\lambda}_i>0} -\frac{\mu_i^k}{\tilde{\lambda}_i}$ and set

$$\alpha = \begin{cases} \overline{\alpha} & \tilde{\lambda}_1 \geq 0 \\ \underline{\alpha} & \tilde{\lambda}_1 < 0. \end{cases}.$$

iv. Update $\mu_{\mathbf{v}^i}^{k+1} := \mu_{\mathbf{v}^i}^{k+1} + \alpha\tilde{\lambda}_i$ for all $i = 1, \ldots, L+1$.

v. Compute $I = \left\{ i \in \{1, \ldots, L+1\} : \mu_{\mathbf{v}^i}^{k+1} = 0 \right\}$.

vi. For each $i \in I$ remove column $i-1$ matrix $\mathbf{W}^{k+1}$.

vii. Update $U^{k+1} = U^{k+1} / \{\mathbf{v}_i\}_{i \in I}$.

7. If $\mathbf{W}^{k+1}$ is not in row echelon form, then construct a matrix $\tilde{\mathbf{T}}$, as a composition of elementary matrices, such that $\tilde{\mathbf{T}}\mathbf{W}^{k+1}$ is row echelon form, and update $\mathbf{W}^{k+1} := \tilde{\mathbf{T}}\mathbf{W}^{k+1}$ and $\mathbf{T}^{k+1} := \tilde{\mathbf{T}}\mathbf{T}^{k+1}$.

Notice that in order to compute the row rank of the matrix $\mathbf{W}^{k+1}$ in step 6(d), we may simply convert the matrix to row echelon form, and then count the number of nonzero rows. This is done similarly to step 7, and requires ranking of at most one column. We will need to rerank the matrix in step 7 only if $L > M$, and subsequently at least one column is removed in step 6(e)vi.

The IRR scheme may reduce the size of the input U^{k+1} only in the case of a forward step, since otherwise the vertices in U^{k+1} are all affinely independent. Nonetheless, the IRR scheme *must* be applied at each iteration in order to maintain the matrices $\mathbf{W}^k$ and $\mathbf{T}^k$.

The efficiency of the scheme relies on the fact that only a small number of vertices are either added to or removed from the representation. The potentially computationally expensive steps are: step 5(b)—replacing the reference vertex, step 6(d)—finding the row rank of $\mathbf{W}^{k+1}$, step 6(e)i—solving the system of linear equalities, step 6(e)vi—removing columns corresponding with the vertices eliminated from the representation, and step 7—the ranking of the resulting matrix $\mathbf{W}^{k+1}$. Step 5(b) can be implemented without explicitly using matrix multiplication and therefore has a computational cost of $O(n^2)$. Since $\mathbf{W}^k$ was in row echelon form, step 6(d) requires a row elimination procedure, similar to step 7, to be conducted only on the last column of $\mathbf{W}^{k+1}$, which involves at most $O(n)$ operations and an additional $O(n^2)$ operation for updating $\mathbf{T}^{k+1}$. Moreover, since $\mathbf{W}^k$ was full column rank, the IRR scheme guarantees that in step 6(e)i the vector $\boldsymbol{\lambda}$ has a unique solution, and since $\mathbf{W}^{k+1}$ is in row echelon form, it can be found in $O(n^2)$ operations. Moreover, in step 6(e)i, the specific choice of α ensures that the reference vertex $\mathbf{v}^1$ is not eliminated from the representation, and so there is no need to change the reference vertex at this stage. Furthermore, it is reasonable to assume that the set I satisfies $|I| = O(1)$, since otherwise the vector $\mathbf{x}^{k+1}$, produced by a forward step, can be represented by significantly less vertices than $\mathbf{x}^k$, which, although possible, is numerically unlikely. Therefore, assuming that indeed

$|I| = O(1)$, the matrix $\tilde{\mathbf{T}}$, calculated in step 7, applies a row elimination procedure to at most $O(1)$ rows (one for each column removed from $\mathbf{W}^{k+1}$) or one column (if a column was added to $\mathbf{W}^{k+1}$). Conducting such an elimination on either row or column takes at most $O(n^2)$ operations, which may include row switching and at most n row addition and multiplication. Therefore, the total computational cost of the IRR scheme amounts to $O(n^2)$.

References

1. Beck, A., Teboulle, M.: A conditional gradient method with linear rate of convergence for solving convex linear systems. Math. Methods of Oper. Res. **59**(2), 235–247 (2004)
2. Beck, A., Teboulle, M.: Gradient-based algorithms with applications to signal recovery problems. In: Palomar, D., Eldar, Y. (eds.) Convex Optimization in Signal Processing and Communications, pp. 139–162. Cambridge University Press, Cambridge (2009)
3. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont, MA, USA (1999)
4. Bertsimas, D., Tsitsiklis, J.N.: Introduction to Linear Optimization, vol. 6. Athena Scientific, Belmont (1997)
5. Canon, M.D., Cullum, C.D.: A tight upper bound on the rate of convergence of Frank-Wolfe algorithm. SIAM J. Control **6**(4), 509–516 (1968)
6. Dunn, J., Harshbarger, S.: Conditional gradient algorithms with open loop step size rules. J. Math. Anal. Appl. **62**(2), 432–444 (1978)
7. Epelman, M., Freund, R.M.: Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. Math. Program. **88**(3), 451–485 (2000)
8. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Nav. Res. Logist. Quart. **3**(1–2), 95–110 (1956)
9. Freund, R.M., Grigas, P., Mazumder, R.: An extended frank-wolfe method with "in-face" directions, and its application to low-rank matrix completion. arXiv preprint; arXiv:1511.02204 (2015)
10. Garber, D., Hazan, E.: A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. arXiv preprint; arXiv:1301.4666 (2013)
11. Goldfarb, D., Todd, M.J.: Chapter ii: Linear programming. In: Nemhauser, G., Kan, A.R., Todd, M. (eds.) Optimization, volume 1 of Handbooks in Operations Research and Management Science, pp. 73–170. Elsevier, Amsterdam (1989)
12. Guelat, J., Marcotte, P.: Some comments on Wolfe's away step. Math. Program. **35**(1), 110–119 (1986)
13. Güler, O.: Foundations of Optimization. Graduate Texts in Mathematics, vol. 258. Springer, New York (2010)
14. Hoffman, A.J.: On approximate solutions of systems of linear inequalities. J. Res. Natl. Bur. Stand. **49**(4), 263–265 (1952)
15. Jaggi, M.: Sparse Convex Optimization Methods for Machine Learning. Ph.D. thesis, ETH Zurich (2011)
16. Lacoste-Julien, S., Jaggi, M.: An affine invariant linear convergence analysis for Frank-Wolfe algorithms. In: NIPS 2013 Workshop on Greedy Algorithms, Frank-Wolfe and Friends (2014)
17. Lacoste-Julien, S., Jaggi, M.: On the global linear convergence of frank-wolfe optimization variants. In: Advances in Neural Information Processing Systems, pp. 496–504 (2015)
18. Levitin, E., Polyak, B.T.: Constrained minimization methods. USSR Comput. Math. Math. Phys. **6**(5), 787–823 (1966)
19. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course, vol. 87. Springer, Berlin (2004)
20. Pena, J., Rodriguez, D.: Polytope conditioning and linear convergence of the frank-wolfe algorithm. arXiv preprint; arXiv:1512.06142 (2015)
21. Luo, Z.Q., Tseng, P.: Error bounds and convergence analysis of feasible descent methods: a general approach. Ann. Oper. Res. **46–47**(1), 157–178 (1993)
22. Rockafellar, R.T.: Convex Analysis, 2nd edn. Princeton University Press, Princeton (1970)
23. Wang, P.-W., Lin, C.-J.: Iteration complexity of feasible descent methods for convex optimization. J. Mach. Learn. Res. **15**, 1523–1548 (2014)
24. Wolfe, P.: Chapter 1:Convergence Theory in Nonlinear Programming. In: Abadie J. (ed.) Integer and nonlinear programming, pp. 1–36. North-Holland Publishing Company, Amsterdam (1970)