A comparative note on the relaxation algorithms for the linear semi-infinite feasibility problem *

A. Ferrer, † M.A. Goberna‡ E. González-Gutiérrez§
M.I. Todorov¶

November 25, 2015

Abstract

The problem (LFP) of finding a feasible solution to a given linear semi-infinite system arises in different contexts. This paper provides an empirical comparative study of relaxation algorithms for (LFP). In this study we consider, together with the classical algorithm, implemented with different values of the fixed parameter (the step size), a new relaxation algorithm with random parameter which outperforms the classical one in most test problems whatever fixed parameter is taken. This new algorithm converges geometrically to a feasible solution under mild conditions. The relaxation algorithms under comparison have been implemented using the Extended Cutting Angle Method (ECAM) for solving the global optimization subproblems.

Keywords: Linear semi-infinite systems, feasibility problem, relaxation method, cutting angle method.

*This research was partially supported by MICINN of Spain, Grant MTM2014-59179-C2-1-P and Sistema Nacional de Investigadores, Mexico.

†Departament de Matemàtica Aplicada I, Universitat Politècnica de Catalunya, Spain, E-Mail: alberto.ferrer@upc.edu

‡Dep. of Statistics and Operations Research, Alicante University, 03071 Alicante, Spain. E-Mail: mgoberna@ua.es.

§School of Engineering, Polytechnic University of Tulancingo, Tulancingo, Hidalgo, MX. E-Mail: enrique.gonzalez@upt.edu.mx

¶Dep. of Physics and Mathematics, UDLAP, Puebla, MX. On leave from IMI-BAS, Sofia, BG. E-Mail: maxim.todorov@udlap.mx

1 Introduction

This paper deals with linear feasibility problems of the form

$$(LFP) \text{ Find } x \in \mathbb{R}^n \text{ such that } a(t)^\top x \geq b(t), \ \forall t \in T, \tag{1}$$

where T is an infinite index set, $a(t) := (a_1(t), ..., a_n(t)) \in \mathbb{R}^n$ and $b(t) \in \mathbb{R}$ for all $t \in T$. We say that (LFP) is *semi-infinite* as the number of unknowns is finite while the number of constraints is infinite. We denote by

$$F = \left\{ x \in \mathbb{R}^n : a(t)^\top x \geq b(t), \ \forall t \in T \right\}$$

the set of solutions to (LFP).

Let us mention some fields where linear feasibility problems arise in a natural way. A problem like (LFP) has to be solved to get a starting point when one applies a feasible direction method to some linear semi-infinite program (an updated list of documented applications of linear semi-infinite programming can be found in [26, Remark 1.3.3]). Some interesting applications of (LFP) also include the image recovery problem [18] and the robust optimization problem [12]. In particular, the feasibility of a robust linear optimization problem can be reformulated as an example of (LFP) [13]. For more recent development for robust linear multi-objective optimization problem see [23] and [24]. Observe also that any convex (possibly semi-infinite) feasibility problem

$$\text{Find } x \in \mathbb{R}^n \text{ such that } g_s(x) \leq 0, \ \forall s \in S,$$

can be linearized in different ways (e.g., as in [25, (7.10)] or [19, pp. 117-118]) giving rise to a problem like (LFP). Thus, numerical methods for (LFP) could be used to get a starting point when solving convex programs through feasible direction methods (there exists a wide literature on the applications of convex programming). Still in the framework of convex programming, a particular instance of (LFP) arises at each step of the subgradient methods (which are slower than the Newton-like methods but allow to solve non-differentiable convex programs). Indeed, given a convex non-differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, such methods require the computation at step r of a subgradient at the current iterate x^r, i.e., they require to solve (LFP) with T being the domain of f, $a(t) := x^r - t$, and $b(t) := f(x^r) - f(t)$. Analogously, the computation of $\varepsilon-$subgradients and certain variational inequalities can be reformulated in terms of (LFP).

It is well-known that the linear finite feasibility problem can be solved by means of any linear programming method. Unfortunately, the same is not true when T is infinite. The ellipsoid algorithm for finding a feasible point in a convex set could potentially be adapted to solve (LFP), but no implementation is known up to know (even though the ingredients for the complexity analysis of such an implementation are already available [16]). The adaptation of numerical methods conceived for different problems seems also possible but not without difficulties. So, a natural way to tackle (LFP) consists of reformulating it as convex finite feasibility problems by replacing the infinitely many constraints $a(t)^\top x \geq b(t)$, $t \in T$, by a single convex inequality $\varphi(x) \leq 0$, where $\varphi(x) := \max_{t\in T}\left(b(t) - a(t)^\top x\right)$. Applying any convex programming method to minimize φ, one could either find the aimed solution of (LFP) or conclude that no such solution exists. The drawback with this approach is that minimizing φ is usually intractable as its Lipschitz constant cannot be estimated or, even worst, it is not Lipschitz continuous (unless one can replace $\mathbb{R}^n$ with some polytope). Another potential approach consists of extending to infinitely many sets (in this case the half-spaces $\left\{x \in \mathbb{R}^n : a(t)^\top x \geq b(t)\right\}$, $t \in T$) the Douglas-Rachford method for finite families of closed convex sets [15], but proving the convergence could be a hard task.

For all the reasons above, the unique available algorithms for solving (LFP) are semi-infinite variants of the classical relaxation method introduced in 1954, independently, by Agmon and by Motzkin and Schoenberg, for the linear finite feasibility problem. It is well-known that this method either generates a finite sequence whose last element is a feasible solution or generates an infinite sequence which converges geometrically to some feasible solution. Variants of the relaxation algorithm have strongly polynomial time for special classes of the linear finite feasibility problems (see [6], [14] and references therein). The semi-infinite fixed step relaxation algorithm can be briefly described as follows: select a (relaxation) parameter $\lambda \in (0, 2]$ and, if the current iterate at step $r \in \mathbb{N}$ is $x^r \notin F$, compute the next iterate as

$$x^{r+1} := x^r + \lambda\varepsilon_r \frac{a(t_r)}{\|a(t_r)\|}, \tag{2}$$

where ε_r approximates the supremum μ_r of the distance from x^r to the hyperplane $H_r = \left\{x \in \mathbb{R}^n : a(t_r)^\top x = b(t_r)\right\}$ determined by some constraint

$a(t_r)^\top x \geq b(t_r)$, $t_r \in T$, violated by x^r : $\lambda = 1$ in [33] and [34], while $\lambda \in (0, 2]$ in [27], [28], [29], and [30]. If $\varepsilon_r = \mu_r$, x^{r+1} is the projection of x^r onto H_r when $\lambda = 1$ and the symmetric of x^r with respect to H_r when $\lambda = 2$. All the mentioned works are focused on the convergence analysis and provide few numerical examples (if any).

In this paper we propose a new relaxation algorithm where the user could select a parameter $\nu \in (0, 2)$ and replace the fixed parameter λ in (2) by some $\lambda_r \in [\nu, 2]$ depending on r. The sequence $\{\lambda_r\} \subset [\nu, 2]$ can be either predetermined by the user or generated at random. In all our implementations of the latter algorithm λ_r is a random variable uniformly distributed on $[\nu, 2]$. This is also the first work comparing the numerical efficiency of the relaxation algorithms for (LFP), with different values of the relevant parameters from the efficiency point of view: λ in the case of relaxation with fixed step length and ν in the case of relaxation with random step length.

Section 2 contains the necessary notation, the expression of the assumptions of the convergence theorems in terms of the data. We also mention some features of the Extended Cutting Angle Method (ECAM) used to check the feasibility of the current iterate x^r and to construct the new iterate x^{r+1} (two global optimization subproblems). Section 3 shows the convergence of the new algorithm under some mild conditions while Section 4 shows its geometric convergence. Section 5 describes the numerical experiments to compare the computational efficiency of several implementations of the classical and the new relaxation algorithm, Finally, Section 6 provides the conclusions of this comparative study. For the sake of completeness we include a first appendix providing complementary information on ECAM and a second one containing a brief introduction to the performance profiles used to interpret the numerical experiments.

2 Preliminaries

We start this section by introducing the necessary notation. The Euclidean norm of $x \in \mathbb{R}^n$ is represented by $\|x\|$, the corresponding open ball centered at x and radius $\varepsilon > 0$ by $B_\varepsilon(x)$, and the zero vector by 0_n. The Euclidean distance in $\mathbb{R}^n$ is denoted by d. The L_1 norm of $x \in \mathbb{R}^n$ is represented by $\|x\|_1$. Given $X \subset \mathbb{R}^n$, $\operatorname{cl} X$ and $\operatorname{bd} X$ denote the closure and the boundary of X, $\operatorname{span} X$ the linear span of X, $\operatorname{aff} X$ the affine hull of X, $\operatorname{conv} X$ the *convex hull* of X, and $\operatorname{cone} X := \mathbb{R}_+ \operatorname{conv} X$ the *convex conical hull* of $X \cup \{0_n\}$. If X

is convex, $\dim X$, $\operatorname{ri} X$, and $\operatorname{extr} X$ denote the dimension, the relative interior, and the set of extreme points of X, respectively. We also denote by $\mathbb{R}^{(T)}$ the space of mappings $\xi : T \to \mathbb{R}$ with finite *support* $\{t \in T : \xi(t) \neq 0\}$, and by $\mathbb{R}^{(T)}_+$ its positive cone.

The graph of a real-valued function f is denoted by $\operatorname{gph} f$ and its domain by $\operatorname{dom} f$; moreover, given $x \in \operatorname{dom} f$, the gradient and the convex subdifferential of f at x are denoted by $\nabla f(x)$ and $\partial f(x)$, when they exist.

We associate with (LFP), corresponding to the linear system

$$\left\{a(t)^\top x \geq b(t),\ t \in T\right\},$$

the so-called *reference cone*

$$K(a,b) := \operatorname{cl}\operatorname{cone}\{(a(t), b(t)),\ t \in T; (0_n, -1)\},$$

where $a \in (\mathbb{R}^n)^T$ and $b \in \mathbb{R}^T$ are the functions $t \mapsto a(t)$ and $t \mapsto b(t)$, respectively. The existence theorem for linear semi-infinite systems establishes that $F \neq \emptyset$ if and only if $(0_n, 1) \notin K(a,b)$ while the Farkas lemma asserts that, given $F \neq \emptyset$ and $(c,d) \in \mathbb{R}^{n+1}$, $c^\top x \geq d$ holds for all $x \in F$ if and only the coefficient's vector $(c,d) \in K(a,b)$ [25, Chapter 3]. Consequently,

$$\operatorname{aff} F = \bigcap_{(c,d)\in\mathcal{H}} \left\{x \in \mathbb{R}^n : c^\top x = d\right\},$$

where $\mathcal{H} := \{(c,d) \in \mathbb{R}^{n+1} : \operatorname{span}\{(c,d)\} \subset K(a,b)\}$. Then, $\dim F = n$ if and only if $\mathcal{H} = \{0_{n+1}\}$ if and only if $K(a,b)$ contains no line [25, Corollary 3.1.1 and Theorem 5.8]. Thus, the condition for the convergence of the relaxation algorithm with arbitrary starting point x^0, $\dim F = n$ (or the weakest one that $x^0 \in \operatorname{aff} F$) can be expressed in terms of the data, but unfortunately, it can hardy be verified in practice.

We solve the global optimization subproblems in the implementations of the relaxation algorithms by means of the Extended Cutting Angle Method (ECAM in short). ECAM solves optimization problems of the form

$$\inf\{f(x) : x \in X\}, \tag{3}$$

where f is Lipschitz continuous with known Lipschitz constant and $X \subset \mathbb{R}^n$ is a polytope (i.e., a bounded convex polyhedral set). We denote by $\inf_X f \in \mathbb{R}$ the optimal value of (3). ECAM is briefly described in Appendix 1. We shall

use the following two lemmas to get the Lipschitz constants for the functions involved in the subproblems to be solved by the relaxation algorithms in this paper. The first lemma deals with the generation of Lipschitz continuous functions from functions of the same class while the second lemma exploits the smoothness of the functions and the convexity of their domains. The proofs can be found in the standard literature on the subject (see, e.g., [21, Chapter 12] and [32, Proposition 5.1]).

Lemma 1 *Let $f_1, f_2 : T \to \mathbb{R}$ be Lipschitz continuous on T with constants L_1, L_2. Then the following statements hold:*
(i) If $\sup_T |f_i| \leq M_i < +\infty$, $i = 1, 2$, then the product $f_1 f_2$ is Lipschitz continuous on T with Lipschitz modulus (the smallest Lipschitz constant) at most $M_1 L_2 + M_2 L_1$.
(ii) If $0 < m_1 \leq \inf_T |f_1|$ and $\sup_T |f_2| \leq M_2 < +\infty$, then, $\frac{f_2}{f_1}$ is Lipschitz continuous on T with Lipschitz modulus at most $\frac{L_2}{m_1} + \frac{M_2 L_1}{m_1^2}$.

With $f \in \mathcal{C}^1(T)$ we mean that f is continuously differentiable on an open set containing $T \subset \mathbb{R}^m$, $m \in \mathbb{N}$.

Lemma 2 *Let $T \subset \mathbb{R}^m$ be a non-singleton compact convex set and $f \in \mathcal{C}^1(T)$. Then, f is Lipschitz continuous on T with Lipschitz modulus at most $\max_T \|\nabla f\|$.*

3 Convergence of the extended relaxation algorithm

From now on we assume that $a(t) \neq 0_n$ for all $t \in T$, so that the function $g(\cdot, x) := a(\cdot)^\top x - b(\cdot)$ is well-defined for all $x \in \mathbb{R}^n$. Moreover, $g(\cdot, x)$ satisfies $\inf_T \frac{g(\cdot,x)}{\|a(\cdot)\|} \neq -\infty$ as, in the contrary, there exists a sequence $\{t_k\} \subset T$ such that $\frac{g(t_k,x)}{\|a(t_k)\|} \to -\infty$ as $k \to \infty$ and, taking into account that $\left\| \frac{a(t_k)^\top x}{\|a(t_k)\|} \right\| \leq \|x\|$, we have $\frac{b(t_k)}{\|a(t_k)\|} \to +\infty$, which in turn implies that $\frac{a(t_k)}{b(t_k)} \to 0_n$ as $k \to \infty$. So, $(0_n, 1) \in K(a, b)$ (contradiction). Consequently, the extended relaxation algorithm (**ERA** in short) described in Table 1, where the step length is not necessarily predetermined, is well-defined too.

Table 1: Extended relaxation algorithm, ERA

Procedure: ERA

Initialization:
Select $M > 1$, $\beta_0 > 0$, $\tau > 0$ (precision), $\nu \in]0, 2]$;
Choose $x^0 \in \mathbb{R}^n$;
$r := 0$ (set to zero initial iteration),
$\beta := \beta_0$ (value for the initial β-global optimal solution),
non_stop:=true (binary variable);

begin
while (non_stop) **do**
Obtain (via ECAM) ε_r, a β-global optimal solution by solving the problem:

$$\mu_r - \beta < \varepsilon_r := \frac{b(t_r) - a(t_r)^\top x^r}{\|a(t_r)\|} \leq \mu_r := -\inf\left\{\frac{a(t)^\top x^r - b(t)}{\|a(t)\|} : t \in T\right\} \qquad (4)$$

if($\varepsilon_r \geq \tau$) **then**
if($\beta < \varepsilon_r(M-1)$) **then**
Choose $\lambda_r \in [\nu, 2]$ (in some way);
$x^{r+1} := x^r + \lambda_r \varepsilon_r \frac{a(t_r)}{\|a(t_r)\|}$
$r := r + 1$
else
$\beta := \beta/2$
endif
else
non_stop:=false;
endif
endwhile
return x^r, a feasible solution;
end

Before to proceed further, we shall make some comments. The parameter β represents the accuracy level required for the subproblem (4) to be solved at step r, whose exact optimal value is denoted by μ_r, while t_r and ε_r are, an approximation of the current optimal global solution, and a β-global approximation of the optimal value, respectively. If $\beta < \varepsilon_r (M-1)$ one can compute directly the new iterate; if not, β is replaced by a smaller positive scalar of the form $\frac{\beta}{2^k}$, $k \in \mathbb{N}$, until the previous inequality holds. The necessity of enforcing $\beta < \varepsilon_r (M-1)$ at each step comes from the fact that this inequality guarantees that $\mu_r \to 0$, which is the main ingredient of the convergence proof of Theorem 6 below.

Obviously, ERA can be implemented in different ways, e.g., by taking $\lambda_r = \lambda$ (a fixed parameter in $[\nu, 2]$) for each $r = 0, 1, 2, \ldots$ (the classical fixed step relaxation algorithm **FISRA**), by choosing a predetermined sequence $\{\lambda_r\} \subset [\nu, 2]$, or by picking up the parameter λ_r at random in some subinterval of $[\nu, 2]$ (the new random step relaxation algorithm **RASRA**). Iteration r of ERA requires a $\beta-$optimal solution of $-\inf \left\{ \frac{g(\cdot, x^r)}{\|a(\cdot)\|} : t \in T \right\}$, where x^r is the current iterate. This can be done via ECAM provided that these functions are Lipschitz continuous with known Lipschitz constants on a polytope T contained in some Euclidean space (in most practical applications the index set T is a low dimensional box, usually with $\dim T \in \{1, 2\}$).

The next two results can be useful in order to apply ECAM to the subproblems of ERA. The first one involves the constants

$$B := \inf_{t \in T} \|a(t)\|, \ N := \sup_{t \in T} \|a(t)\|, \text{ and } P := \sup_{t \in T} |b(t)|. \tag{5}$$

The first two constants, B and N, play an important role in the proof of the convergence Theorem 11, where we shall assume that $B > 0$ and $N < +\infty$. Observe that $B > 0$ and $N, P < +\infty$ whenever T is a compact set, $a : T \to \mathbb{R}^n$ and $b : T \to \mathbb{R}$ are continuous.

Proposition 3 *Let $b, a_1, ..., a_n$ be Lipschitz continuous on $T \subset \mathbb{R}^m$, with Lipschitz constants $L_0, L_1, ..., L_n$, and assume that $B > 0$ and $N, P < +\infty$. Denote $L := (L_1, ..., L_n) \in \mathbb{R}^n$ and let $x^r = (x_1^r, ..., x_n^r) \in \mathbb{R}^n$. Then $\frac{g(\cdot, x^r)}{\|a(\cdot)\|}$ is Lipschitz continuous on T with Lipschitz modulus at most*

$$\frac{1}{B} (L_0 + \|L\| \|x^r\|) + \frac{N}{B^3} (P + N \|x^r\|) \|L\|_1. \tag{6}$$

Proof. Since $g(\cdot, x^r) = \sum_{i=1}^{n} x_i^r a_i(\cdot) - b(\cdot)$ is a linear combination of $n+1$ Lipschitz continuous functions with Lipschitz constants $L_1, ..., L_n$ and L_0, we get that $g(\cdot, x^r)$ is Lipschitz continuous on T with Lipschitz modulus at most $L_0 + \|L\| \|x^r\|$.
By Lemma 1(i), for each $i = 1, ..., n$, $a_i(\cdot)^2$ is Lipschitz continuous on T with Lipschitz modulus at most $2NL_i$. So, $\|a(\cdot)\|^2 = \sum_{i=1}^{n} a_i(\cdot)^2$ is Lipschitz continuous on T with Lipschitz modulus at most $2N\|L\|_1$. Thus, if $t, s \in T$, we have

$$|\|a(t)\| - \|a(s)\|| = \left| \frac{\|a(t)\|^2 - \|a(s)\|^2}{\|a(t)\| + \|a(s)\|} \right| \leq \left(\frac{N\|L\|_1}{B} \right) \|t - s\|,$$

which shows that $\|a(\cdot)\|$ is Lipschitz continuous on T with Lipschitz modulus at most $\frac{N\|L\|_1}{B}$.

Observe also that, given $t \in T$,

$$|g(t, x^r)| \leq |b(t)| + \|x^r\| \|a(t)\| \leq P + N\|x^r\|. \tag{7}$$

Now we apply Lemma 1(ii) to the functions $f_1 = \|a(\cdot)\|$ and $f_2 = g(\cdot, x^r)$, with $0 < B \leq \inf_T |f_1|$ and

$$\sup_T |f_2| = \sup_T |g(\cdot, x^r)| \leq P + N\|x^r\| < +\infty$$

by (7). Then we get (6). ■

Let us introduce two additional constants when $b, a_1, ..., a_n \in \mathcal{C}^1(T)$ and T is compact:

$$Q := \max_{i=1,...,n; t \in T} \|\nabla a_i(t)\| \text{ and } R := \max_{t \in T} \|\nabla b(t)\|.$$

Proposition 4 *Let $T \subset \mathbb{R}^m$ be a non-singleton compact convex set, $b, a_1, ..., a_n \in \mathcal{C}^1(T)$, and assume that $B > 0$. Then, given $x^r \in \mathbb{R}^n$, $\frac{g(\cdot, x^r)}{\|a(\cdot)\|}$ is Lipschitz continuous on T with Lipschitz modulus at most*

$$\left(\frac{N}{B^3} \right) \left[B\left(\|x^r\|_1 Q + R \right) + \left(P + N\|x^r\| \right) nQ \right]. \tag{8}$$

Proof. Observe that

$$\nabla_t \frac{g(t, x^r)}{\|a(t)\|} = \frac{\|a(t)\| \nabla_t g(t, x^r) - g(t, x^r) \nabla \left(\|a(t)\| \right)}{\|a(t)\|^2}.$$

Since $\max_{t\in T}\|a(t)\| \leq N$, then

$$\max_{t\in T}\|\nabla_t g(t,x^r)\| \leq \|x^r\|_1 Q + R,$$

and

$$\max_{t\in T} as\,|g(t,x^r)| \leq P + N\|x^r\|.$$

Since

$$\|\nabla(\|a(t)\|)\| = \|a(t)\|^{-1}\left\|\sum\nolimits_{i=1,\ldots,n} a_i(t)\nabla a_i(t)\right\| \leq nB^{-1}NQ,\ \forall t\in T,$$

one has $\max_{t\in T}\|\nabla(\|a(t)\|)\| \leq nB^{-1}NQ$, which together with $\min_{t\in T}\|a(t)\|^2 \geq B^2$ shows that $\max_{t\in T}\left\|\nabla_t\left(\frac{g(t,x^r)}{\|a(t)\|}\right)\right\|$ is not greater than the real number in (8). Lemma 2 yields the aimed conclusion. ■

Example 5 *In robust linear programming with uncertain constraints (see, e.g., [24, Section 3]), one assumes that the objective function $x \mapsto c^\top x$ is deterministic while the coefficient vectors of the p given constraints take values on given (generally infinite) uncertainty sets $\mathcal{U}_j$, $j = 1,\ldots,p$. The robust feasible solutions are the feasible solutions of the so-called* robust counterpart *problem*

$$\min\left\{c^\top x : a_j^\top x \geq b_j,\ \forall(a_j,b_j)\in\mathcal{U}_j,\ j=1,\ldots,p\right\}.$$

So, computing a robust feasible solution is the linear feasibility problem

$$(LFP)\ \ \textit{Find } x\in\mathbb{R}^n \textit{ such that } a^\top x \geq b,\ \forall(a,b)\in T, \tag{9}$$

where $T = \bigcup\limits_{j=1,\ldots,p}\mathcal{U}_j$, which can be written as (LFP) in (1), with $a: T\to\mathbb{R}^n$ such that $a(t_1,...,t_{n+1}) = (t_1,...,t_n)$ and $b: T\to\mathbb{R}$ such that $b(t_1,...,t_{n+1}) = t_{n+1}$. Observe that T is compact whenever $\mathcal{U}_j$ is compact for all $j = 1,\ldots,p$.

Obviously, the projection functions $a_i(\cdot)$ and $b(\cdot)$ are Lipschitz continuous with Lipschitz moduli equal to 1. *Most robust decision makers choose uncertainty sets of the form*

$$\mathcal{U}_j := (\overline{a}_j,\overline{b}_j) + \alpha_j\mathcal{U},\quad j=1,\ldots,p, \tag{10}$$

where $(\overline{a}_j,\overline{b}_j)\in\mathbb{R}^{n+1}$ are deterministic vectors and $\alpha_j \geq 0$, $j = 1,\ldots,p$, while $\mathcal{U}$ denotes the closed unit ball for some norm on $\mathbb{R}^{n+1}$. For simplicity

we consider here the affine constraint data perturbations model (9)-(10) with $\mathcal{U} = B_1(0_{n+1})$. *ERA is well defined provided that* $a(t) \neq 0_n$ *for all* $t \in T$, *i.e.,*

$$\|\overline{a}_j\| > \alpha_j, j = 1, \ldots, p,$$

or, equivalently, $B := \inf_{j=1,\ldots,p}(\|\overline{a}_j\| - \alpha_j) > 0$. *If* $x^r \in \mathbb{R}^n$ *is the present iterate, by Proposition 3,* $\frac{g(\cdot,x^r)}{\|a(\cdot)\|}$ *is Lipschitz continuous on* T *with Lipschitz modulus at most*

$$\frac{1}{B}\left(1 + \sqrt{n}\,\|x^r\|\right) + \frac{nN}{B^3}\left(P + N\,\|x^r\|\right),$$

where

$$N = \sup_{j=1,\ldots,p}(\|\overline{a}_j\| + \alpha_j) \text{ and } P = \sup_{j=1,\ldots,p} \max\left\{\left|\overline{b}_j - \alpha_j\right|, \left|\overline{b}_j + \alpha_j\right|\right\}.$$

Other Lipschitz constants can be obtained for other norms in a similar way.

When the unit ball $\mathcal{U}$ *is a polytope (e.g. for the* L_1 *and the* L_∞ *norms), one can reformulate* (LFP) *in (9)-(10) as*

$$(LFP) \text{ Find } x \in \mathbb{R}^n \text{ such that } t^\top (x, -1) \geq 0,\ t \in T,$$

where $T = \operatorname{conv}\left[\bigcup_{j=1,\ldots,p}\left((\overline{a}_j, \overline{b}_j) + \alpha_j \operatorname{extr}\mathcal{U}\right)\right]$ *is a polytope in* $\mathbb{R}^{n+1}$. *Observe that* $Q = R = 1$, *but Proposition 4 does not applies as* T *is the union of* p *closed balls and, so, generally non-convex for* $p \geq 2$.
According to [24, Theorem 4], (LFP) *has solutions, i.e.* $F \neq \emptyset$, *whenever* $\max_{j=1,\ldots,p} \alpha_j$ *is less than the distance from the so-called* hypographical set

$$\operatorname{conv}\left\{(\overline{a}_j, \overline{b}_j), j = 1, \ldots, p\right\} + \mathbb{R}_+\left\{(0_n, -1)\right\}$$

to the origin 0_{n+1}. *This distance can be computed by solving a quadratic programming problem. Unfortunately, the assumption that* $\dim F = n$ *in the convergence theorems below, which can be expressed in terms of the data as the requirement that the convex cone*

$$\operatorname{cl}\left(\sum_{i=1,..,p} \mathbb{R}_+ \operatorname{cone}\left\{(\overline{a}_j, \overline{b}_j) + \alpha_j \mathcal{U}\right\} + \mathbb{R}_+\left\{(0_n, -1)\right\}\right)$$

contains no line , is not checkable. In other words, the user must apply ERA assuming that $\dim F = n$ *and conclude that* $\dim F < n$ *for those feasibility problems for which the generated sequence* $\{x^r\}$ *does not converge to some feasible solution.*

Now, we shall modify the proof of the convergence of FISRA, [29, Theorem 3], in order to get the new proof of the convergence of ERA.

Theorem 6 (Convergence) *Assume that* $\dim F = n$. *Let* $x^0 \in \mathbb{R}^n$ *and* $\nu \in]0, 2[$. *If for each* $r = 0, 1, 2, \ldots$ *we chose an arbitrary* $\lambda_r \in [\nu, 2]$, *ERA either ends after a finite number of steps, or generates an infinite sequence* $\{x^r\}$ *converging to some element of* F.

Proof. Observe that, if ERA generates a finite sequence, the last point is an approximate solution to (LFP). So, we can assume w.l.o.g. that $\{x^r\}$ is an infinite sequence of infeasible points.

For each $t \in T$ we denote $H_t = \{x \in \mathbb{R}^n : a(t)^\top x = b(t)\}$. Given $r \in \mathbb{N}$, we have $\mu_r > 0$, i.e., $x^r \notin H_{t_r}$ for some $t_r \in T$. Thus, x^{r+1} belongs to the half-line emanating from x^r in the direction of $a(t_r)$, with $d(x^{r+1}, x^r) = \lambda_r \varepsilon_r$.

By hypothesis, there exist $z \in \mathbb{R}^n$ and $\delta > 0$ such that

$$B_\delta(z) \subset F \subset \{x \in \mathbb{R}^n : a(t_r)^\top x \geq b(t_r)\}, \quad r = 1, 2, \ldots$$

and $\rho_{t_r} := d(z, H_{t_r}) \geq \delta$.

By construction, the line determined by x^r and x^{r+1}, $\operatorname{aff}\{x^r, x^{r+1}\}$, is orthogonal to H_{t_r}. Let $h_r = d(z, \operatorname{aff}\{x^r, x^{r+1}\})$. We select a coordinate system in the hyperplane $\operatorname{aff}\{x^r, x^{r+1}, z\}$ such that the abscissa axis is the line $\operatorname{aff}\{x^r, x^{r+1}\}$, oriented in the direction from x^r to x^{r+1}, the axis of ordinates is the line orthogonal to $\operatorname{aff}\{x^r, x^{r+1}\}$, oriented in such a way that z belongs to the first quadrant, and the origin is located at $H_{t_r} \cap \operatorname{aff}\{x^r, x^{r+1}\}$. With this oriented system, the coordinates of the x^r are $(-\varepsilon_r, 0)$, the coordinates of x^{r+1} are $((\lambda_r - 1)\varepsilon_r, 0) = (\xi_r \varepsilon_r, 0)$, with $\lambda_r - 1 = \xi_r \in]-1, 1]$, and the coordinates of z are (ρ_{t_r}, h_r), with $h_r \geq 0$ (the case when $\dim \operatorname{aff}\{x^r, x^{r+1}, z\} = 1$ and $h_r = 0$ is trivial). Figure 1 illustrates the notations, which are the same as in [29, Theorem 3].

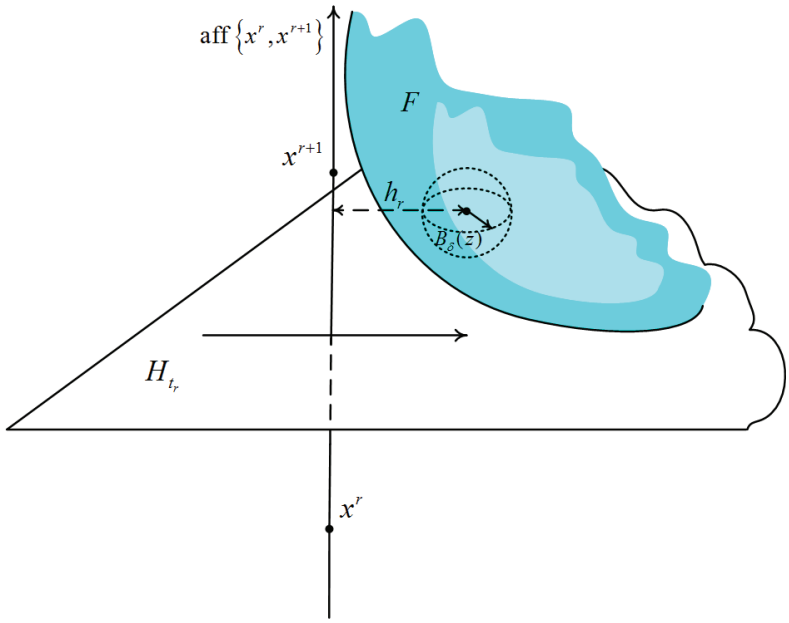

Figure 1: Coordinate system involving aff$\{x^r, x^{r+1}\}$ and the hyperplane H_{t_r}.

Following exactly the same steps as in the beginning of the proof of [29, Theorem 3], we obtain the following inequality

$$\sum_{k=0}^{r-1}(1+\xi_k)\varepsilon_k \leq \frac{1}{2\delta}\|x^0 - z\|^2.$$

Since $\xi_k = \lambda_k - 1$ and $\nu \leq \lambda_k$ for all $k = 0, \ldots, r-1$, one gets

$$\sum_{k=0}^{r-1}\nu\varepsilon_k \leq \sum_{k=0}^{r-1}\lambda_k\varepsilon_k \leq \frac{1}{2\delta}\|x^0 - z\|^2, \tag{11}$$

which gives

$$\sum_{k=0}^{r-1}\varepsilon_k \leq \frac{1}{2\delta\nu}\|x^0 - z\|^2. \tag{12}$$

Defining $\eta_{r-1} := \sum_{k=0}^{r-1}\varepsilon_k$, and $K := \frac{1}{2\nu\delta}\|x^0 - z\|^2$, from (12) we get $0 \leq \eta_{r-1} \leq K$ for all $r \in \mathbb{N}$. As the sequence $\{\eta_r\}$ is bounded and increasing, it is convergent, with $0 \leq \lim_r \eta_r \leq K$. Hence, $\sum_{r=0}^{\infty}\varepsilon_r$ converges as well (and $\lim_r \varepsilon_r = 0$).

We have that $|\mu_r - \varepsilon_r| < \beta$, for every $r = 1, 2, ...$, and require at each step that $\beta < \varepsilon_r(M-1)$, which is equivalent to $\beta + \varepsilon_r < \varepsilon_r M$. But, we have that $\beta + \varepsilon_r > \mu_r$, whereby

$$\frac{\mu_r}{M} < \varepsilon_r \leq \mu_r,$$

i.e., $0 < \mu_r < \varepsilon_r M$, so, we get $\lim_r \mu_r = 0$.

From (11) we have

$$\sum_{k=0}^{r-1} \lambda_k \varepsilon_k \leq \frac{1}{2\delta} \|x^0 - z\|^2,$$

but, from the definition of ERA, we have

$$\lambda_k \varepsilon_k = \|x^r - x^{r+1}\|.$$

So,

$$\sum_{k=0}^{r-1} \|x^r - x^{r+1}\| \leq \frac{1}{2\delta} \|x^0 - z\|^2$$

and then the series $\sum_{r=0}^{\infty} \|x^r - x^{r+1}\|$ converges. Therefore, $\sum_{r=0}^{\infty}(x^r - x^{r+1})$ is absolutely convergent (see, e.g., [5, Theorem 26.7]), and we conclude the existence of some $\hat{x} \in \mathbb{R}^n$ such that $\lim_r x^r = \hat{x}$.

It remains to show that $\hat{x} \in F$. For any $t \in T$, and for all $r \in \mathbb{N}$ we have

$$-\frac{g(t, x^r)}{\|a(t)\|} = \frac{b(t) - a(t)^\top x^r}{\|a(t)\|} \leq \begin{cases} \mu_r, & \text{if } g(t, x^r) < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Passing to the limit in (13) as $r \to \infty$ we get $\frac{b(t) - a(t)^\top \hat{x}}{\|a(t)\|} \leq 0$, for all $t \in T$, and this proves that $\hat{x} \in F$. ■

Observe that when $\dim F = n$ and ERA generates an infinite sequence $\{x^r\}$, its limit $\hat{x} \in \operatorname{bd} F$ as $x^r \in \mathbb{R}^n \diagdown F$ for all $r \in \mathbb{N}$.

The next example shows that the non-degeneracy assumption that $\dim F = n$ in Theorem 6 is not superfluous. Even more, the computational experience in Section 4 shows that the convergence is quite slow whenever the *condition number* of F (assumed to be bounded), say $\operatorname{cond}(F)$, defined as the quotient of the smallest width of F by the greatest one, is small. Obviously, for a compact convex set set F, $\dim F < n$ if and only if $\operatorname{cond}(F) = 0$.

Example 7 *The simple feasibility problem*

$$(LFP)\ \ \textit{Find } x \in \mathbb{R}^2 \ \textit{s.t.}\ \ -d\left(\cos t\right) x_1 - c\left(\sin t\right) x_2 \geq -cd,\ \ \forall t \in [0, 2\pi],$$

where c and d are two given positive numbers, illustrates the difficulties encountered by ERA when solving feasibility problems when $\dim F = n$ *but* $\operatorname{cond}(F)$ *is very small. It is easy to see that* $F = \left\{x \in \mathbb{R}^2 : \frac{x_1^2}{c^2} + \frac{x_2^2}{d^2} \leq 1\right\}$, *with* $\operatorname{cond}(F) = \min\left\{\frac{c}{d}, \frac{d}{c}\right\}$. *Assuming that ERA generates an infinite sequence* $\{x^r\}$ *whose limit* $\hat{x} \neq (\pm c, 0)$, *and that* $0 < d < c$, $x_2^r \neq 0$ *for sufficiently large* r *because* $x_2^r \to \hat{x}_2 \neq 0$ *(as the unique points* $x \in \operatorname{bd} F$ *such that* $x_2 = 0$ *are* $(\pm c, 0)$*),* $\frac{d}{dt}\left(\frac{g(x^r,t)}{\|a(t)\|}\right)_{t=0} = -\frac{c}{d}x_2^r$, *and*

$$\left|\frac{d}{dt}\left(\frac{g\left(x^r, t\right)}{\|a\left(t\right)\|}\right)_{t=0}\right| = \frac{|x_2^r|}{\operatorname{cond}\left(F\right)} \to +\infty \ \textit{as}\ \operatorname{cond}\left(F\right) \to 0.$$

Hence the Lipschitz modulus of $\frac{g(x^r,t)}{\|a(t)\|}$ *tends to* $+\infty$ *too as* $\operatorname{cond}(F)$ *tends to zero, making ECAM to become inefficient to solve the global optimization subproblems. This theoretical observation is coherent with the empirical results shown in Table 2 (see Subsection 5.2).*

Consider now the limit case that $d = 0$ *while* $c > 0$. *Obviously,* $F = \mathbb{R} \times \{0\}$ *with* $\dim F < n = 2$. *Recall that ERA selects at step* r *a parameter* $\lambda_r \in (0, 2]$ *and, if the current iterate is* $x^r \notin F$, *computes the next iterate by (2), with* ε_r *approximating the supremum* $\mu_r = d\left(x^r, H_r\right)$, *where* $H_r = \left\{x \in \mathbb{R}^n : a\left(t_r\right)^\top x = b\left(t_r\right)\right\}$ *is the hyperplane determined by some constraint violated by* x^r. *Consider* (LFP) *with* $d = 0$ *and take* $\varepsilon_r = \mu_r$ *for all* r. *Given* $x^r \notin F$ *(i.e.,* $x_2^r \neq 0$*),* $H_r = F$ *(the* x *axis), and* $\operatorname{gph} g\left(\cdot, x^r\right)$ *is the curve in red (in blue) in Figure 2 whenever* $x_2^r > 0$ *(*$x_2^r < 0$, *respectively), so that*

$$\operatorname{argmin} g\left(t, x^r\right) = \begin{cases} \left\{\frac{3\pi}{2}\right\}, & \textit{if } x_2^r > 0, \\ \left\{\frac{\pi}{2}\right\}, & \textit{if } x_2^r < 0. \end{cases}$$

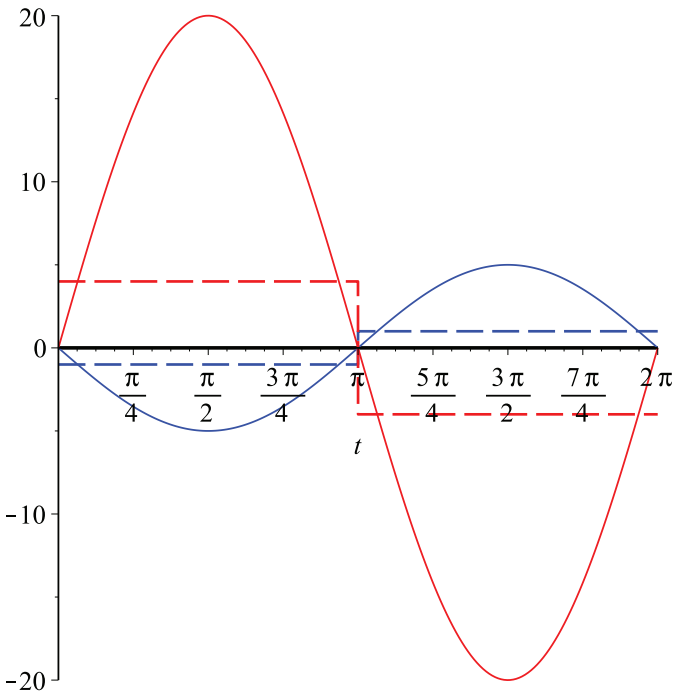

Figure 2: Graphs of the functions $g(\cdot, x^r)$ and $\frac{g(\cdot,x^r)}{\|a(\cdot)\|}$.

Figure 2 shows the graph of the piecewise linear function $g(\cdot, x^r)$ (represented with dashed points) and the graph of the smooth function to be minimized at step r, $\frac{g(\cdot,x^r)}{\|a(\cdot)\|}$, in both cases in red (blue) whenever $x_2^r > 0$ ($x_2^r < 0$, respectively). We now apply FISRA with different choices of the step size λ and the initial point x^0. As Figure 3 shows, the results are as follows:

i) If $\lambda = 0.5$ and $x^0 = (-7, 4)$, then FISRA generates an infinite sequence $\{x^r\} \to x^* \in F$ contained in the open half-plane $x_2 > 0$.

ii) If $\lambda = 1.0$ and $x^0 = (-3, 4)$, then FISRA provides a point of $x^* \in F$ in just one iteration.

iii) If $\lambda = 1.5$ and $x^0 = (3, 4)$, then FISRA generates again an infinite sequence $\{x^r\} \to x^* \in F$, whose even (odd) terms are contained in the open half-plane $x_2 > 0$ ($x_2 < 0$, respectively).

iv) If $\lambda = 2.0$ and $x^0 = (7, 4)$, then FISRA fails (the oscillating sequence $x^r = \left(7, (-1)^{r+1} 4\right)$ does not converge).

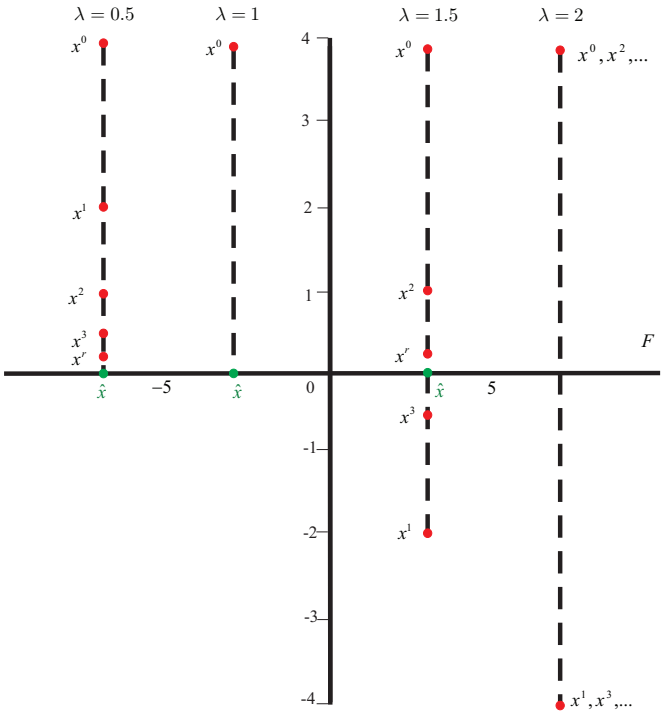

Figure 3: Different choices of the step size λ and the initial point x^0

Remark 8 *ERA can be conceptually adapted to the unrealistic situation in which dim $F = m < n$ and the affine hull of F is known, i.e.,* aff $F = p + V$, *for a given $p \in \mathbb{R}^n$ and a given linear subspace V of dimension m. We thus have:*

- *The translation $x = z + p$ allows us to replace F with a closed convex set*
$$\overline{F} := \left\{ z \in \mathbb{R}^n : a(t)^\top z - \overline{b}(t) \geq 0,\ \forall t \in T \right\},$$
with $\overline{b}(\cdot) = b(\cdot) - a(\cdot)^\top p$, so that $F = \overline{F} + p$ and aff $\overline{F} = V$.

- *We can complete an arbitrary basis $\{v^1, \ldots, v^m\}$ of V with $n - m$ linearly independent vectors $\{w^{m+1}, \ldots, w^n\}$ to get a basis of $\mathbb{R}^n$. Thus, $\mathbb{R}^n = V \oplus W$.*

- *We can find a $n \times n$ non-singular matrix $B = [B_1 \mid B_2]$, with $B_1 (n \times m)$ and $B_2 (n \times (n-m))$, such that $z = B \begin{pmatrix} y^1 \\ y^2 \end{pmatrix}$, where $y = (y^1, y^2) \in \mathbb{R}^m \times \mathbb{R}^{n-m}$ is the vector formed by the coordinates of $z \in \mathbb{R}^n$ in the basis $\{v^1, \ldots, v^m; w^{m+1}, \ldots, w^n\}$. Observe that $y^2 = 0$ for all $y \in V =$* aff $\overline{F}$.

- *The result of replacing $z = B \begin{pmatrix} y^1 \\ 0_{n-m} \end{pmatrix}$ in the linear system*
$$\left\{ a(t)^\top z - \overline{b}(t) \geq 0,\ \forall t \in T \right\}$$
is the system $\left\{ \widetilde{a}(t)^\top y_1 - \overline{b}(t) \geq 0,\ \forall t \in T \right\}$, with $\widetilde{a}(\cdot) = B_1^\top a(\cdot)$.

- *Then ERA allows to compute an element $\widehat{y}^1$ of*
$$\widetilde{F} := \left\{ y^1 \in \mathbb{R}^m : \widetilde{a}(t)^\top z - \overline{b}(t) \geq 0,\ \forall t \in T \right\}$$
as dim $\widetilde{F} =$ dim $\overline{F} = m$ *(i.e. $\widetilde{F}$ has full dimension in $\mathbb{R}^m$). So, $\widehat{x} := p + B \begin{pmatrix} \widehat{y}^1 \\ 0_{n-m} \end{pmatrix} \in F$.*

4 Rate of convergence of ERA

The objective of this section is to show that, taking $\lambda_r \in [\nu, \mu] \subset (0, 2)$ for all $r \in \mathbb{N}$, the rate of convergence of ERA is geometric. To prove it we need two lemmas.

Lemma 9 *[1, Lemma 2.1]Let $\lambda \in [0, 2]$ and $x, y \in \mathbb{R}^n$ be separated by the hyperplane $H = \{x \in \mathbb{R}^n : a^\top x = b\}$, that is $a^\top x < b$ and $a^\top y \geq b$. Then*

$$\|x + \lambda (x_H - x) - y\|^2 \leq \|x - y\|^2 - \lambda (2 - \lambda) \|x_H - x\|^2, \tag{14}$$

where x_H is the orthogonal projection of x on H. The equality holds if $\lambda = 0$, or $\lambda = 2$ and $y \in H$.

We also need the following extension of [33, Lemma 1], whose assumptions involve the smallest and greatest distances from 0_n to the set $\{a(t) : t \in T\}$ introduced in (5): $B := \inf_{t \in T} \|a(t)\| \in \mathbb{R}_+$ and $N := \sup_{t \in T} \|a(t)\| \in \mathbb{R}_+ \cup \{+\infty\}$, respectively.

Lemma 10 *[27, Lemma 5] Assume that ERA generates an infinite sequence $\{x^r\}$. If $\dim F = n$, $B > 0$ and $N < +\infty$, then there exists a constant $0 < \gamma < 1$ such that $\mu_r \geq \gamma d(x^r, F)$ for all $r = 0, 1, 2, \ldots$.*

Now, we are ready to prove the following theorem on the rate of convergence of ERA.

Theorem 11 (Geometric convergence) *Let $\lambda_r \in [\nu, \mu] \subset (0, 2)$ for all $r = 0, 1, 2, \ldots$, with $\nu < \mu$, and assume that ERA generates an infinite sequence $\{x^r\}$. If $\dim F = n$, $B > 0$ and $N < +\infty$, then there exist $M > 1$, $0 < \theta < 1$, and $\bar{x} \in F$ such that $\bar{x} = \lim_r x^r$ and*

$$\|x^r - \bar{x}\| \leq \theta^r \|x^0 - \bar{x}\| \tag{15}$$

for all r big enough.

Proof. From the definition of ε_r, we have $\varepsilon_r = \|x^r - x_{H_{t_r}}\|$, where $x_{H_{t_r}}$ is the orthogonal projection of x^r on the hyperplane H_{t_r}. We know that

$$\varepsilon_r > \frac{\mu_r}{M}, \ r = 0, 1, \ldots. \tag{16}$$

Let us replace x by x^r, y by y^r and λ by λ_r in the inequality (14), where y^r be the point in F such that $\|x^r - y^r\| = d(x^r, F)$, that is, y^r is the projection of x^r on F. From Lemma 9, Lemma 10 and the fact that $\|x^{r+1} - y^{r+1}\|^2 \leq \|x^{r+1} - y^r\|^2$, we get

$$\begin{aligned}\|x^{r+1} - y^{r+1}\|^2 &\leq \|x^{r+1} - y^r\|^2 \leq \|x^r - y^r\|^2 - \lambda_r(2-\lambda_r)\|x^r - x_{H_{t_r}}\|^2 \\ &= \|x^r - y^r\|^2 - \lambda_r(2-\lambda_r)\varepsilon_r^2 \\ &\leq \|x^r - y^r\|^2 - \frac{\lambda_r(2-\lambda_r)\mu_r^2}{M^2} \\ &\leq \|x^r - y^r\|^2 - \frac{\lambda_r(2-\lambda_r)\gamma^2}{M^2}\|x^r - y^r\|^2 \\ &= \|x^r - y^r\|^2(1 - \lambda_r(2-\lambda_r)\gamma^2 M^{-2}). \end{aligned} \tag{17}$$

Let us define $\zeta := \min[\nu(2-\nu), \mu(2-\mu)]$. Then $0 < \zeta \leq \lambda_r(2-\lambda_r) \leq 1$, $r = 0, 1, ...$Thus, for a sufficiently large M we have $0 < \sigma = (1 - \zeta\gamma^2 M^{-2})^{\frac{1}{2}} < 1$ and, making use of (17) repeatedly, we get

$$\|x^{r+1} - y^{r+1}\| \leq \sigma^{r+1}\|x^0 - y^0\|.$$

Since $\bar{x}$ and x^r are in the ball $B_{\|x^r - y^r\|}(y^r)$ for each $r = 0, 1, 2, ...$, we finally obtain

$$\frac{1}{2}\|x^{r+1} - \bar{x}\| \leq \|x^{r+1} - y^{r+1}\| \leq \sigma^{r+1}\|x^0 - y^0\| \leq \sigma^{r+1}\|x^0 - \bar{x}\|, \tag{18}$$

which proves the theorem for any θ such that $\sigma < \theta < 1$. ■

Remark 12 *From (18), it would be convenient estimating the smallest σ such that (15) holds for any θ such that $\sigma < \theta < 1$, for sufficiently large values of r. Assuming $M > 1$, we can chose $M > \max\left\{1, \zeta^{\frac{1}{2}}\gamma\right\} = 1$, because $\zeta \in (0,1)$ and $\gamma \in (0,1)$. This means that $(1 - \zeta\gamma^2)^{\frac{1}{2}} < \sigma < 1$.*

5 Numerical results

In this section we present the results of numerical experiments to compare different implementations of FISRA (depending on the fixed value of $\lambda \in (0, 2]$) and RASRA (depending on the chosen distribution for λ_r). In the latter case, we have chosen uniform distributions on intervals of the form $[\nu, 2]$, with $0 < \nu < 2$, but other distributions on subintervals of $(0, 2]$ could be used. Observe that, for the chosen distribution of λ_r, RASRA converges, but the convergence could be slow as we may have $\lambda_r = 2$.

5.1 Test problems

A total of 27 linear feasibility test problems have been selected satisfying the assumption guaranteeing the convergence of the relaxation algorithms ($\dim F = n$) and the conditions allowing to check the feasibility of the current iterate though ECAM (T polyhedral and Lipschitzian data functions). From the test problems, and by considering several distances from the randomly generated initial point to the origin, we have obtained 41 different test instances (see Tables 2 and 3). These distances are significative in this study because they increase the computational time. Nevertheless, in practice we don't know how far the initial point is from F. So, we do not consider necessary to work with initial points far from the origin since this fact increases the complexity of the functions to be optimized. In our experiments we have selected distances 10, 20 and 50, just to illustrate the difficulties associated with high distances. Instances from No. 1 to No. 12 have been generated from ellipses with decreasing condition number ($\mathrm{cond}\,(F)$), which is indicated between parenthesis. Instances from No. 13 to No. 21 come from [31, Examples 8, 9 and 10]. Finally, instances from No. 22 to No. 41 have been generated by using the procedure described in [22]. In this latter case we can generate test problems without limitations on the number, n, of variables and the dimension, $m := \dim T$, of the index set.

5.2 Computational results

The numerical experiments, which are summarized in four tables, were carried out on a PC with Processor Intel(R) Core(TM) i5-4200U CPU $1.60-2.30$ GHz and 8 GB of RAM (MS Windows7 enterprise). In Tables 2 and 3, *Num* denotes the number assigned to the instance, *Name* indicates the name of the instance, and *Iter* and *Time* represent the number of iterations and the CPUTime required for obtaining a feasible solution, respectively. Table 2 describes instances with $\lambda_r = \lambda$ for all $r \in \mathbb{N}$ (constant sequences) while Table 3 describes instances with random values of λ_r. The maximum number of iterations was limited to 400 for all instances. When the algorithm needs more than 400 iterations to attain a solution of (LFP), then we consider that the solver has failed in solving the problem. The failure of a solver is indicated with a star $(*)$, in the column indicating the number of iterations.

Table 2: Fixed value of $\lambda_r \in]0, 2]$

				$\lambda_r = 0.1$		$\lambda_r = 0.4$		$\lambda_r = 0.7$		$\lambda_r = 1.0$		$\lambda_r = 1.2$		$\lambda_r = 1.5$		$\lambda_r = 1.8$		$\lambda_r = 2.0$	
Num	Name (cond (F))	n	m	Iter	Time	Iter	Time	Iter	Time	Iter	Time	Iter	Time	Iter	Time	Iter	Time	Iter	Time
1	elps.5.4.10 (0.8)	2	1	324	16.084	68	2.387	30	1.045	1	0.047	1	0.047	1	0.046	1	0.047	1	0.031
2	elps.5.4.20	2	1	334	16.240	70	2.433	31	1.092	3	0.109	1	0.047	1	0.031	2	0.063	2	0.078
3	elps.5.4.50	2	1	344	16.848	72	3.510	32	1.592	3	0.156	2	0.093	2	0.094	4	0.171	6	0.266
4	elps.5.3.10 (0.6)	2	1	325	15.921	68	2.964	30	1.326	3	0.140	1	0.062	1	0.047	1	0.047	2	0.062
5	elps.5.3.20	2	1	332	15.943	70	3.105	31	1.419	3	0.156	1	0.063	2	0.078	2	0.093	3	0.125
6	elps.5.3.50	2	1	345	16.676	72	3.464	31	1.513	3	0.156	2	0.093	3	0.125	5	0.219	8	0.327
7	elps.5.1.10 (0.2)	2	1	326	19.936	69	3.027	30	1.326	1	0.047	2	0.093	2	0.063	3	0.093	4	0.094
8	elps.5.1.20	2	1	334	20.546	70	4.305	31	1.903	2	0.125	2	0.109	3	0.110	5	0.187	9	0.312
9	elps.5.1.50	2	1	344	20.780	72	3.510	31	1.528	2	0.110	2	0.078	4	0.124	8	0.219	22	0.639
10	elps.5.01.10 (0.002)	2	1	328	18.798	69	2.886	30	1.264	2	0.093	3	0.125	6	0.203	12	0.296	401*	72.915
11	elps.5.01.20	2	1	335	19.235	66	2.777	31	1.310	1	0.047	4	0.172	7	0.234	17	0.421	401*	74.459
12	elps.5.01.50	2	1	345	20.545	67	2.823	32	1.389	2	0.109	4	0.125	7	0.171	20	0.406	401*	68.359
13	GT14.ex8.10	2	1	334	8.666	70	1.607	31	0.702	1	0.031	1	0.031	2	0.063	2	0.047	3	0.078
14	GT14.ex8.20	2	1	336	10.327	70	1.607	31	0.733	6	0.125	2	0.047	2	0.062	14	0.109	4	0.109
15	GT14.ex8.50	2	1	344	8.830	72	1.872	32	0.795	3	0.063	3	0.078	3	0.078	5	0.109	8	0.203
16	GT14.ex9.10	2	1	335	3.994	70	0.702	31	0.296	1	0.032	3	0.062	7	0.172	23	0.343	401*	72.540
17	GT14.ex9.20	2	1	341	3.963	71	0.795	31	0.359	1	0.016	5	0.156	11	0.234	159	1.825	401*	83.132
18	GT14.ex9.50	2	1	346	4.025	72	0.671	31	0.281	1	0.015	3	0.047	53	0.437	25	0.374	401*	61.527
19	GT14.ex10.10	2	2	591	12.776	141	3.074	61	1.279	2	0.062	1	0.047	1	0.047	1	0.047	1	0.046
20	GT14.ex10.20	2	2	571	12.620	144	2.715	63	1.138	2	0.063	1	0.031	1	0.031	1	0.031	1	0.032
21	GT14.ex10.50	2	2	514	10.218	149	2.403	54	0.904	3	0.063	2	0.031	2	0.031	2	0.031	2	0.032
22	FPftpea.20	3	1	335	19.547	70	4.337	31	1.950	3	0.218	2	0.156	7	0.437	7	0.437	12	0.764
23	FPftpea.20	5	1	336	20.468	70	4.290	31	1.981	2	0.171	2	0.156	9	0.593	7	0.453	13	0.826
24	FPftpea.20	10	1	335	24.149	70	4.695	31	2.278	2	0.187	2	0.156	13	0.842	8	0.531	12	0.795
25	FPftpea.20	15	1	335	26.364	70	5.102	31	2.527	1	0.125	2	0.156	26	1.731	9	0.624	12	0.827
26	FPftpea.20	25	1	335	27.441	70	6.770	31	2.512	3	0.374	1	0.125	178	13.541	12	0.967	13	1.061
27	FPftpea.20	50	1	335	44.563	70	10.358	31	4.836	3	0.687	1	0.312	3	0.515	17	2.012	15	1.825
28	FPftpea.20	75	1	333	77.113	71	14.149	31	5.632	2	0.530	1	0.312	3	0.593	13	2.137	14	2.262
29	FPftpea.20	100	1	336	74.319	70	14.118	31	6.021	1	0.281	2	0.468	3	0.609	6	1.216	10	1.919
30	FPftpea.20	500	1	335	350.712	69	157.716	29	27.487	3	3.214	2	2.231	3	3.198	5	5.195	10	10.155
31	FPftpea.20	1000	1	334	761.235	149	2.403	30	68.609	2	4.914	2	5.226	3	7.489	5	12.058	10	23.634
32	FPftpeaT2.20	3	2	297	39.047	66	8.128	34	4.118	4	0.577	2	0.266	3	0.374	5	0.577	10	1.092
33	FPftpeaT2.20	5	2	297	43.368	60	8.424	29	6.771	3	0.483	2	0.359	3	0.718	5	0.920	10	2.059
34	FPftpeaT2.20	10	2	243	45.053	72	10.858	31	8.829	3	0.749	2	0.577	3	0.812	5	1.482	10	2.636
35	FPftpeaT2.20	15	2	344	54.729	73	16.552	30	7.394	2	0.655	2	0.577	3	0.936	5	1.514	9	2.511
36	FPftpeaT2.20	25	2	291	73.051	73	21.856	26	6.302	3	0.920	2	0.733	3	1.061	5	1.622	9	2.964
37	FPftpeaT2.20	50	2	270	87.807	64	25.007	30	9.782	3	1.107	2	1.077	3	1.170	5	2.215	9	3.744
38	FPftpeaT2.20	75	2	254	114.344	61	28.361	25	12.277	2	1.186	2	1.092	3	1.607	5	2.589	9	4.353
39	FPftpeaT2.20	100	2	283	160.884	61	33.384	26	14.398	1	0.780	2	1.326	1	0.047	5	2.871	9	5.070
40	FPftpeaT2.20	500	2	262	641.210	61	133.295	26	55.443	1	3.198	2	5.148	1	0.031	5	11.435	10	22.199
41	FPftpeaT2.20	1000	2	282	1324.955	54	2374.227	27	125.783	3	16.723	2	11.279	3	15.584	5	25.818	10	49.437

Table 3: Random values of $\lambda_r \in [\nu, 2]$ for fixed ν

				$\nu = 0.01$		$\nu = 0.4$		$\nu = 1.0$		$\nu = 1.5$		$\nu = 1.9$	
Num	Name (cond (F))	n	m	Iter	Time	Iter	Time	Iter	Time	Iter	Time	Iter	Time
1	elps.5.4.10 (0.8)	2	1	2	0.094	1	0.031	1	0.047	1	0.031	1	0.031
2	elps.5.4.20	2	1	1	0.078	2	0.094	2	0.078	2	0.093	2	0.094
3	elps.5.4.50	2	1	3	0.156	4	0.172	1	0.047	3	0.125	5	0.202
4	elps.5.3.10 (0.6)	2	1	1	0.046	1	0.047	1	0.047	1	0.047	1	0.047
5	elps.5.3.20	2	1	1	0.062	3	0.063	2	0.078	3	0.109	3	0.094
6	elps.5.3.50	2	1	2	0.094	2	0.078	3	0.109	4	0.156	6	0.203
7	elps.5.1.10 (0.2)	2	1	2	0.156	2	0.125	3	0.141	3	0.124	4	0.141
8	elps.5.1.20	2	1	3	0.125	2	0.125	3	0.125	5	0.187	7	0.265
9	elps.5.1.50	2	1	5	0.266	4	0.171	5	0.172	8	0.234	12	0.359
10	elps.5.01.10 (0.002)	2	1	6	0.515	6	0.468	4	0.281	9	0.452	23	0.843
11	elps.5.01.20	2	1	5	0.373	7	0.233	6	0.233	13	0.414	57	1.282
12	elps.5.01.50	2	1	5	0.482	7	0.289	2	0.115	13	0.378	42	1.038
13	GT14.ex8.10	2	1	2	0.094	2	0.047	1	0.031	1	0.031	1	0.031
14	GT14.ex8.20	2	1	4	0.171	2	0.063	1	0.031	2	0.062	3	0.094
15	GT14.ex8.50	2	1	5	0.172	4	0.109	1	0.031	3	0.078	6	0.156
16	GT14.ex9.10	2	1	5	0.219	5	0.078	4	0.109	12	0.187	76	0.936
17	GT14.ex9.20	2	1	6	0.266	8	0.187	10	0.218	16	0.281	68	1.014
18	GT14.ex9.50	2	1	30	0.390	4	0.125	10	0.265	14	0.359	82	1.435
19	GT14.ex10.10	2	2	1	0.016	2	0.047	2	0.046	2	0.063	2	0.031
20	GT14.ex10.20	2	2	4	0.140	3	0.094	2	0.046	2	0.047	2	0.063
21	GT14.ex10.50	2	2	32	0.437	30	0.390	4	0.094	4	0.078	10	0.265
22	FPftpea.20	3	1	6	0.374	3	0.156	5	0.219	7	0.296	9	0.359
23	FPftpea.20	5	1	9	0.582	7	0.390	5	0.374	6	0.328	10	0.530
24	FPftpea.20	10	1	4	0.344	4	0.249	5	0.328	4	0.312	8	0.468
25	FPftpea.20	15	1	4	0.359	1	0.093	3	0.219	5	0.327	8	0.593
26	FPftpea.20	25	1	6	0.437	3	0.265	3	0.281	5	0.421	8	0.655
27	FPftpea.20	50	1	7	0.999	4	0.515	1	0.156	3	0.358	8	0.952
28	FPftpea.20	75	1	3	0.570	3	0.581	3	0.560	5	0.893	8	1.313
29	FPftpea.20	100	1	1	0.297	3	0.639	4	0.796	5	1.045	8	1.545
30	FPftpea.20	500	1	2	2.652	3	3.104	5	5.148	4	4.259	8	7.972
31	FPftpea.20	1000	1	1	3.136	5	13.088	2	5.476	6	15.693	8	20.436
32	FPftpeaT2.20	3	2	3	0.515	3	0.437	4	0.546	4	0.499	8	1.014
33	FPftpeaT2.20	5	2	17	2.949	15	2.231	3	0.452	5	0.858	8	1.544
34	FPftpeaT2.20	10	2	8	2.144	2	0.586	3	1.023	5	1.347	8	2.266
35	FPftpeaT2.20	15	2	3	1.264	4	1.030	3	0.936	6	1.825	8	2.527
36	FPftpeaT2.20	25	2	3	1.342	4	1.435	1	0.437	4	1.357	7	3.089
37	FPftpeaT2.20	50	2	4	2.932	3	2.138	4	2.979	4	2.715	8	5.288
38	FPftpeaT2.20	75	2	4	3.636	1	1.077	3	3.011	4	3.853	7	7.191
39	FPftpeaT2.20	100	2	2	3.276	4	4.836	5	5.336	5	5.756	9	10.655
40	FPftpeaT2.20	500	2	2	10.421	3	15.943	23	15.023	5	24.819	8	46.442
41	FPftpeaT2.20	1000	2	3	29.421	2	24.570	2	34.648	5	59.717	8	66.471

The results shown in Tables 2 and 3 are compared in Tables 4 and 5. The precise meaning of the entries in the latter tables, $\rho_s(1)$ (probability of success in solving a problem) and ρ_s^* (probability of win over the rest) is explained in the Appendix. For the sake of brevity and clarity, we have just included two figures, Figures 4 and 5 (with different scales in the axis of abscissas, corresponding to the No. of iterations), which plot the performance profile of the results, for the number of iterations, for FISRA and for RASRA.

Table 4: Results for fixed value of λ_r

	Time		Iter	
λ_r	$\rho_s(1)$	ρ_s^*	$\rho_s(1)$	ρ_s^*
0.1	0.0%	100.0%	0.0%	100.0%
0.4	2.4%	100.0%	0.0%	100.0%
0.7	0.0%	100.0%	0.0%	100.0%
1.0	26.2%	100.0%	21.4%	100.0%
1.2	57.14%	100.0%	71.43%	100.0%
1.5	14.3%	100.0%	51.2%	100.0%
1.8	7.1%	100.0%	0.0%	100.0%
2.0	9.5%	85.7%	0.0%	85.7%

Table 5: Results for random values of λ_r

	Time		Iter	
ν	$\rho_s(1)$	ρ_s^*	$\rho_s(1)$	ρ_s^*
0.01	23.8%	100.0%	40.5%	100.0%
0.4	40.5%	100.0%	40.5%	100.0%
1.0	35.7%	100.0%	45.2%	100.0%
1.5	11.9%	100.0%	14.3%	100.0%
1.9	0.0%	100.0%	0.0%	100.0%

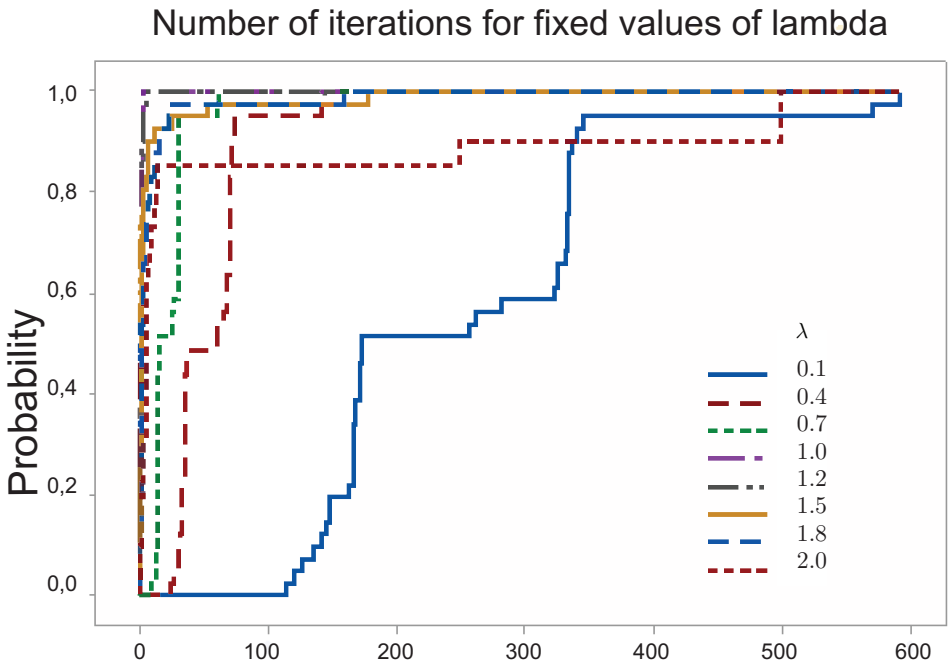

Figure 4: Iterations for fixed values of λ_r.

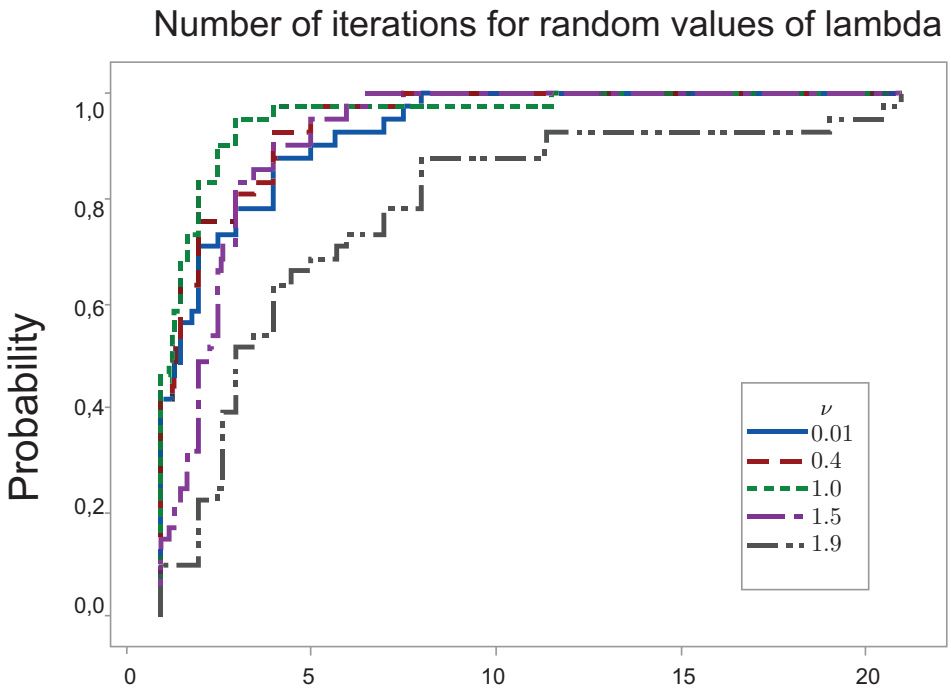

Figure 5: Iterations for random values of λ_r.

6 Conclusions

This paper reports on the implementation of the relaxation algorithm ERA for solving (LFP) which combines different step size iterations with ECAM. It is clear that the main computational difficulty to solve semi-infinite feasibility problems comes from the non-convex optimization problems that must be solved efficiently at each iteration. An innovation of this paper consists of tackling these hard global optimization subproblems with the so-called Cutting Angle Method, an efficient global optimization procedure for solving Lipschitz programming problems. Two variants of ERA with fixed and random step sizes, FISRA and RASRA, have been implemented in C++ and run on Visual Studio 2013.

The preliminary numerical considerations are as follows. From the summary results of Tables 2 and 3, we can conclude that, in general, the number of iterations needed to attain a solution of (LFP) is lower for RASRA than for FISRA. Tables 4 and 5 (and Figures 4 and 5) show the probability of win of each implementation over the rest and the probability of success in solving a problem. As we can see in Table 2, FISRA with constant $\lambda_r = 2.0$ fails in solving six of the instances (i.e., it only solves the 85.7% of the instances). So, we can deduce that the random election of λ_r is a more stable procedure in the sense that it solves the 100% of the instances. Nevertheless, when we consider the best case for RASRA, i.e. $\nu = 0.4$, and the best case for FISRA, i.e., $\lambda_r = 1.2$, then FISRA uses less iterations than RASRA (observe that the best fixed step size for FISRA, $\lambda_r = 1.2$, is the middle point of the best interval $[0.4, 2]$ for the random variable λ_r in RASRA). Indeed, by using the corresponding performance profiles to compare the best cases, FISRA with $\lambda_r = 1.2$ and RASRA with $\nu = 0.4$ we obtain that the probability of win for fixed value of $\lambda_r = 1.2$ is 95.1% and the probability of win for $\nu = 0.4$ is 24.4%.

The results obtained in the reported experiments are promising enough to suggest that suitable implementations of RASRA, which combines a relaxation method that uses random election of λ_r together ECAM, could outperform FISRA for solving semi-infinite feasibility problems. In particular, the above empirical observations suggest to replace the uniform distribution of RASRA used in this paper with unimodal symmetric distributions on intervals of the form $[1.2 - \varepsilon, 1.2 + \varepsilon]$, for small values of $\varepsilon > 0$. This could be object of further empirical studies.

Appendix: Extended Cutting Angle Method

The Extended Cutting Angle Method (ECAM in short) due to Beliakov solves very hard optimization problems of the form

$$\inf \{f(x) : x \in X\}, \tag{19}$$

where f is Lipschitz continuous and X is a polytope. For simplicity, we assume that $\dim X = n$. Since any full dimensional polytope can be expressed as the finite union of non-overlapping simplices, X will be a simplex in this appendix.

In ECAM the objective function is optimized by building a sequence of piecewise linear underestimates. ECAM is inspired in the classical Cutting Plane method by Kelley [35] and Cheney and Golstein [17] to solve linearly constrained convex programs of the form (3), where X is the solution set of a given linear system and $f : \mathbb{R}^n \to \mathbb{R}$ is convex. Since f is lower semicontinuous, it is the upper envelope of the set of all its affine minorants, i.e.

$$f = \sup \{h \ : \ h \text{ affine function}, h \leq f\}. \tag{20}$$

Indeed, it is enough to consider in (20) the affine functions of the form $h(x) = f(z) + \langle u, x - z\rangle$, where $u \in \partial f(z)$, the graph of h being a hyperplane which supports the epigraph of f at $(z, f(z))$. Let $x^1, ..., x^k \in X$ be given and consider the affine functions $h^j(x) = f(x^j) + \langle u^j, x - x^j\rangle$, for some $u^j \in \partial f(x^j)$, $j = 1, ..., k$. The function

$$f_k := \max_{j=1,...,k} h^j \tag{21}$$

is a convex piecewise affine underestimate of the objective function f, in other words, a polyhedral convex minorant of f. The k-th iteration of the Cutting Plane method consists of computing an optimal solution x^{k+1} of the approximating problem $\inf \{f_k(x) : x \in X\}$ which results of replacing f with f_k in (3) or, equivalently, solving the linear programming problem in $\mathbb{R}^{n+1}$

$$\inf \left\{x_{n+1} : x \in X, x_{n+1} \geq h^j(x), j = 1, ..., k\right\}, \tag{22}$$

where $x = (x_1, ..., x_n)$. Then the next underestimate of f,

$$f_{k+1} := \max \left\{f_k, h^{k+1}\right\}, \tag{23}$$

is a more accurate approximation to f, and the method iterates.

The Generalized Cutting Plane method for (3), where $f : \mathbb{R}^n \to \mathbb{R}$ is now a non-convex function while $X = \left\{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\right\}$ is the unit simplex, follows the same script, except that the underestimate f_k is built using the so-called H-subgradients (see [36]) instead of ordinary subgradients, so that minimizing f_k on S is no longer a convex problem. The Cutting Angle method ([3],[4]), of which ECAM is a variant, is an efficient numerical method for minimizing the underestimates when f belongs to certain class of abstract convex functions. Assume that f is Lipschitz continuous with Lipschitz constant $M > 0$ and take a scalar $\gamma \geq M$. Let $x^1, ..., x^k \in S$ be given. For $j = 1, ..., k$, we define the *support vector* $l^j \in \mathbb{R}^n$ by

$$l^j_i := \frac{f(x^j)}{\gamma} - x^j_i, \ i = 1, \ldots, n, \tag{24}$$

and the *support function* h^j by

$$h^j(x) := \min_{i=1,...,n} (f(x^j) - \gamma(x^j_i - x_i)) = \min_{i=1,...,n} \gamma(l^j_i + x_i). \tag{25}$$

Since the functions h^j are concave piecewise affine underestimates of f (i.e. polyhedral concave minorants of f), the underestimate f_k defined in (21) is now a saw-tooth underestimate of f and its minimization becomes a hard problem as (22) is no longer a linear program. ECAM locates the set V^k of all local minima of the function f_k which, after sorting, yields the set of global minima of f_k (see [9] and [10] for additional information). A global minimum x^{k+1} of f_k is aggregated to the set $\left\{x^1, ..., x^k\right\}$ and the method iterates with $f_{k+1} := \max\left\{f_k, h^{k+1}\right\}$.

As shown in [9, 10], a necessary and sufficient condition for a point $x^* \in \text{ri } X$ to be a local minimizer of f_k given by (25),(21) is that there exist an index set $J = \{k_1, k_2, \ldots, k_{n+1}\}$, such that

$$d = f_k(x^*) = \gamma(l^{k_1}_1 + x^*_1) = \gamma(l^{k_2}_2 + x^*_2) = \ldots = \gamma(l^{k_{n+1}}_n + x^*_{n+1}),$$

and $\forall i \in \{1, \ldots, n+1\}$,

$$(l^{k_i}_i + x^*_i) < (l^{k_i}_j + x^*_j), j \neq i.$$

Let x^* be a local minimizer of f_k, which corresponds to some index set J satisfying the above conditions. Form the ordered combination of the

support vectors $L = \{l^{k_1}, l^{k_2}, \ldots, l^{k_{n+1}}\}$ that corresponds to J. It is helpful to represent this combination with a matrix L whose rows are the support vectors l^{k_i}:

$$L := \begin{pmatrix} l_1^{k_1} & l_2^{k_1} & \ldots & l_{n+1}^{k_1} \\ l_1^{k_2} & l_2^{k_2} & \ldots & l_{n+1}^{k_2} \\ \vdots & \vdots & \ddots & \vdots \\ l_1^{k_{n+1}} & l_2^{k_{n+1}} & \ldots & l_{n+1}^{k_{n+1}} \end{pmatrix}, \tag{26}$$

so that its components are given by $L_{ij} = \frac{f(x^{k_i})}{\gamma} - x_j^{k_i}$.

Let the support vectors $l^k, k = 1, \ldots, K$ be defined as in (24). Let x^* denote a local minimizer of f_k and $d = f_k(x^*)$. Then the matrix (26) corresponding to x^* enjoys the following properties (see [10]):

1) $\forall i, j \in \{1, \ldots, n+1\}, i \neq j : l_i^{k_j} > l_i^{k_i}$,

2) $\forall r \notin \{k_1, k_2, \ldots, k_{n+1}\}\ \exists i \in \{1, \ldots, n+1\} : L_{ii} = l_i^{k_i} \geq l_i^r$,

3) $d = \frac{\gamma}{n+1}\,(\mathrm{Trace}(L) + 1)$, and

4) $x_i^* = \frac{d}{\gamma} - l_i^{k_i}$, $i = 1, \ldots, n+1$.

Property 1 reads that the diagonal elements of the matrix L are *dominated* by their respective columns, and Property 2 reads that no support vector l^r (which is not part of L) strictly dominates the diagonal of L. The approach taken in [8, 9] is to enumerate all combinations L with the Properties 1-2, which will give the positions of local minima x^* and their values d by using Properties 3-4.

From (23), combinations of L-matrices can be built incrementally, by taking initially the first $n+1$ support vectors (which yields the unique combination $L = \{l^1, l^2, \ldots, l^{n+1}\}$), and then adding one new support vector at a time. Suppose, we have already identified the local minima of f_k, i.e., all the required combinations. When we add another support vector l^{k+1}, we can *inherit* most of the local minima of f_{k+1} (a few will be lost since Property 2 may fail with l^{k+1} playing the role of l^r), and we only need to add a few new local minima, that are new combinations necessarily involving l^{k+1}. These new combinations are simple modifications of those combinations because Property 2 fails with $l^r = l^{k+1}$.

When ECAM is applied for solving the global optimization subproblem (4) at step r of ERA, the procedure finishes when $f_{best} - d^* > \beta$ so, a β-global optimal solution is obtained.

Remark 13 *Notice that the transformation of variables*

1) $\bar{x}_i = x_i - a_i$, $i = 1, \ldots, n$, $d = \sum_{i=1}^{n}(b_i - a_i)$ *with* $\bar{x}_i \geq 0$ *and* $\sum_{i=1}^{n} \bar{x}_i \leq d$

2) $z_i = \frac{\bar{x}_i}{d}$, $i = 1, \ldots, n$, $z_{n+1} = \sum_{i=1}^{n} z_i$,

allows us to replace the program

$$\min\{f(x) : x \in [a, b]\}$$

by the following one:

$$\min\{g(z_1, \ldots, z_{n+1}) : (z_1, \ldots, z_{n+1}) \in X\},$$

where S *denotes the unit simplex in* $\mathbb{R}^{n+1}$.

Appendix: Performance profiles

In this paper we compare, on the one hand, 8 implementations of the classical fixed step relaxation algorithm corresponding to 8 choices of λ on a battery of 27 feasibility problems and, on the other hand, 5 implementations of the new relaxation algorithm with variable step size corresponding to 5 choices of υ on the same set of test problems. Denote by $\mathcal{S}$ the set of implementations to be compared, so that the cardinality of $\mathcal{S}$, denoted by size $\mathcal{S}$ is 8 and 5 for the classic and for the new relaxation algorithms, respectively. Denote also by $\mathcal{P}$ the set of test feasibility problems, with size $\mathcal{P} = 27$ for both algorithms.

The notion of *performance profile* [20] allows us to compare the performance of the implementations from $\mathcal{S}$ on $\mathcal{P}$. For each pair $(p, s) \in \mathcal{P} \times \mathcal{S}$ we define

$f_{p,s} :=$ number of function evaluations required to solve problem p by solver s.

Consider a fixed problem $p \in \mathcal{P}$. The performance of a solver $s \in \mathcal{S}$ able to solve p is compared with the best performance of any solver of $\mathcal{S}$ on the same problem through the *performance ratio*

$$r_{p,s} := \frac{f_{p,s}}{\min\{f_{p,s} : \ s \in \mathcal{S}\}} \geq 1.$$

Obviously, $r_{p,s} = 1$ means that s is a winner for p, as it is at least as good, for solving p, as any other solver of $\mathcal{S}$. For any solver s unable to solve problem p we define $r_{p,s} = r_M$, where r_M denotes an arbitrary scalar such that

$$r_M > \max\{r_{p,s} : \ s \text{ solves } p, \ (p,s) \in \mathcal{P} \times \mathcal{S}\}.$$

The evaluation of the overall performance of $s \in \mathcal{S}$ is based on the stepwise non-decreasing function $\rho_s : \mathbb{R}_+ \to [0,1]$, called *performance profile* of s, defined as follows:

$$\rho_s(t) = \frac{\text{size}\{p \in \mathcal{P} : \ r_{p,s} \leq t\}}{\text{size}\mathcal{P}}, \ t \geq 0.$$

Obviously, $\rho_s(t) = 0$ for all $t \in [0,1[$ and $\rho_s(1)$ is the relative frequency (which could be interpreted as a probability when p is taken at random from $\mathcal{P}$) of wins of solver s over the rest of the solvers. We say in brief that $\rho_s(1)$ is the *probability of win* for s.

Analogously, for $t > 1$, $\rho_s(t)$ represents the probability for solver $s \in \mathcal{S}$ that a performance ratio $r_{p,s}$ is within a factor $t \in \mathbb{R}$ of the best possible ratio, so that ρ_s can be interpreted as a distribution function and the number

$$\rho_s^* := \lim_{t \searrow r_M} \rho_s(t)$$

as the *probability of solving a problem* of $\mathcal{P}$ with $s \in \mathcal{S}$.

Acknowledgments

The authors are grateful to the referees for their constructive comments and helpful suggestions which have contributed to the final preparation of the paper.

References

[1] Agmon, S. (1954). The relaxation method for linear inequalities. *Can. J. Math.* 6, 382-392.

[2] Auslender, A., Ferrer, A., Goberna, M.A. & López, M.A. (2015). Comparative study of RPSALG algorithm for convex semi-infinite programming. *Comput. Optim. Appl.* 60, 59-87.

[3] Bagirov A.M. & Rubinov A.M. (2000). Global minimization of increasing positively homogeneous functions over the unit simplex. *Ann. Oper. Res.* 98, 171-187.

[4] Bagirov, A.M. & Rubinov, A.M. (2001). Modified versions of the cutting angle method. In: N. Hadjisavvas & P.M. Pardalos (Eds.), *Advances in convex analysis and global optimization* (pp. 245-268). Dordrecht: Kluwer.

[5] Bartle, R.G. (1964). *The elements of real analysis*. New York: Wiley.

[6] Basu, A., De Loer, J.A. & Junod, M. (2013). On Chubanov's method for Linear Programming. *INFORMS J. Comput.* 26, 336-350.

[7] Beliakov, G. (2003). Geometry and combinatorics of the cutting angle method. Optimization 52, 379-394.

[8] Beliakov, G. (2004). Cutting angle method. A tool for constrained global optimization. *Optim. Method. Softw.* 19, 137-151.

[9] Beliakov, G. (2005). A review of applications of the cutting angle method. In: A.M. Rubinov & V. Jeyakumar (Eds.), *Continuous optimization* (pp. 209-248). New York: Springer.

[10] Beliakov, G. (2008). Extended cutting angle method of global optimization. *Pacific J. Optim.* 4, 153-176

[11] Beliakov, G. & Ferrer, A. (2010). Bounded lower subdifferentiability optimization techniques: applications. *J. Global Optim.* 47, 211-231.

[12] Ben-Tal, A., El Ghaoui, L. & Nemirovski, A. (2009). *Robust optimization*. Princeton: Princeton U.P.

[13] Ben-Tal, A. & Nemirovski, A. (1999). Robust solutions of uncertain linear programs. Oper. Res. Lett. 25 (1999), no. 1, 1–13.

[14] Betke, U. (2004), Relaxation, new combinatorial and polynomial algorithms for the Linear Feasibility Problem. *Discrete Comput. Geom.* 32, 317-338.

[15] Borwein, J.M. & Tam, M.K. (2014). A Cyclic Douglas-Rachford Iteration Scheme. *J Optim Theory Appl.* 160, 1-29.

[16] Cánovas, M.J., López, M.A., Parra, J. & Toledo, F.J. (2005). Distance to ill-posedness and the consistency value of linear semi-infinite inequality systems. *Math. Programming* 103A, 95-126.

[17] Cheney, E.W. & Goldstein, A.A. (1959). Newton method for convex programming and Tchebycheff approximation. *Numer. Math.* 1, 253-268.

[18] Combettes, P.L. (1996). *The convex feasibility problem in image recovery.* In In: Hawkes P (ed.), Advances in Imaging and Electron Physics, vol. 95, pp. 155-270, New York: Academic Press.

[19] Dinh, N., Goberna, M.A. & López, M.A. (2006). From linear to convex systems: Consistency, Farkas Lemma and applications. *J. Convex Anal.* 13, 279-290.

[20] Dolan, E.D. & Moré, J.J. (2002). Benchmarking optimization software with performance profiles. *Math. Programming* 91, 201-213.

[21] Eriksson, K., Estep, D. & Johnson, C. (2004). *Applied mathematics: body and soul.* Berlin: Springer.

[22] Ferrer, A. & Miranda, E. (2013). Random test examples with known minimum for convex semi-infinite programming problems. *E-prints UPC*. http://hdl.handle.net/2117/19118. Accessed 28 February 2015.

[23] Goberna, M.A., Jeyakumar, V., Li, G. & Vicente-Pérez, J. (2014). Robust solutions of uncertain multi-objective linear semi-infinite programming. *SIAM J. Optim.* 24, 1402-1419.

[24] Goberna, M.A., Jeyakumar, V., Li, G. & Vicente-Pérez, J. (2015). Robust solutions to multi-objective linear programs with uncertain data, *European J. Oper. Res.* 242, 730-743.

[25] Goberna, M.A. & López, M.A. (1998). *Linear semi-infinite optimization.* Chichester: Wiley.

[26] Goberna, M.A. & López, M.A. (2014). *Post-optimal analysis in linear semi-infinite optimization*, Springer Briefs, New York: Springer.

[27] González-Gutiérrez, E., Rebollar, L.A. & Todorov M.I. (2011). Rate of convergence of a class of numerical methods solving linear inequality systems. *Optimization* 60, 947-957.

[28] González-Gutiérrez, E., Rebollar, L.A. & Todorov M.I. (2011). Under and over projection methods for solving linear inequality systems. *Comptes Rendus de la Academie Bulgare des Sciences* 64, 785-790.

[29] González-Gutiérrez, E., Rebollar, L.A. & Todorov M.I. (2012). Relaxation methods for solving linear inequality systems: converging results. *Top* 20, 426-436.

[30] González-Gutiérrez, E. & Todorov, M.I. (2012). A relaxation method for solving systems with infinitely many linear inequalities. *Optimization Letters* 6, 291-298.

[31] González-Gutiérrez E. & Todorov, M.I. (2014). Generalized step iteration in the relaxation method for the feasibility problem. IMPA, Rio de Janeiro, BR, Preprint serie D113/2014.
https://institucional.impa.br/preprint/lista.action?serie=4

[32] Horst, R., Pardalos, P. & Thoai, N.V. (2000). *Introduction to global optimization* (First ed.). Dordrecht: Kluwer.

[33] Hu, H. (1994). A Projection method for solving infinite systems of linear inequalities. In: D-Z. Du & J. Sun (Eds.), *Advances in optimization and approximation (*pp 186-194). Dordrecht: Kluwer.

[34] Jeroslow, R.G. (1979). Some relaxation methods for linear inequalities, *Cahiers du Cero* 21, 43-53.

[35] Kelley, J.E., Jr. (1960). The cutting-plane method for solving convex programs. *J. Soc. Indust. Appl. Math.* 8, 703-712.

[36] Rubinov, A.M. (2000). *Abstract convexity and global optimization.* Dordrecht/Boston: Kluwer.