Solving worrying simplex's instances in polynomial time

Babacar M.Ndiaye[a,1], *Ivan Lavallée*[b], *Diaraf Seck*[a,c]

[a] *Laboratory of Mathematics of Decision and Numerical analysis, FASEG-UCAD BP 45087 Dakar-Fann Senegal*
[b] *Laboratory of Informatics and Complex Systems, LaISC-CHArt Université Paris 8, France*
Unité Mixte Internationale UMMISCO IRD 209, France

Abstract. This paper presents a new approach, namely sénégaulois algorithm, to solve linear programming problem in relatively short polynomial time from a basic idea which allows to obtain an approximate solution with desired accuracy. A key issue of this approach is to find basic feasible solutions using some dichotomic translations of an hyperplane. Our aim is to show the potential and efficiency of such an approach by performing test on three problem sets arising in Klee-Minty's Linear Problem, for which the simplex algorithm takes an exponential number of iterations, i.e. is outside in time complexity. Computational experiments indicate that our approach solves theses instances in very short time.

Keywords: Linear programming (LP), Hyperplane, Dichotomy, Basic feasible solutions, Klee-Minty's LP, Polynomial complexity.

---

1 Introduction

While the simplex method, introduced by Dantzig (1951), works very well in practice for linear optimization problems. In 1972, Klee and Minty (1979) gave a theoretical example for which the simplex method takes an exponential number of iterations. More precisely, they considered maximization

[1] Corresponding author. Tel.:+33 6 14 02 17 35
Email addresses: babacarm.ndiaye@ucad.edu.sn (Babacar M.Ndiaye), ivan.lavallee@gmail.com (Ivan Lavallée), diaraf.seck@ucad.edu.sn (Diaraf Seck)

problem over an n-dimensional "squashed" cube and proved that a variant of the simplex method visits all its 2^n vertices. Thus, the time complexity is not polynomial for the worst case, since $2^n - 1$ iterations are necessary for this n-dimensional linear optimization problem. The pivot rule used in Klee-Minty example was the most negative reduced cost but variants of the Klee-Minty n-cube allow to prove exponential running time for most pivot rules; see Terlaky and Zhang (1993) and the references therein.

The Klee-Minty worst-case example partially stimulated the search for a polynomial algorithm and, in 1979, Khachiyan (1979) ellipsoïd method proved that linear programming is indeed polynomially solvable[2]. The ellipsoïd algorithm runs in polynomial time in worst case analysis (in degeneracy case) but not the simplex one.

In 1984, an Indian, Karmarkar (1984), proposed a more efficient polynomial algorithm that sparked the research on polynomial interior point methods. While the simplex method goes along the edges of the polyhedron corresponding to the feasible region, interior point methods pass through the interior of this polyhedron. Starting at the analytic center, most interior point methods follow the so-called central path and converge to the analytic center of the optimal face; see e.g. Roos et al.(1997); Wright (1997); Ye (1997);Akoa (2005); Wachter and Biegler (2006). This algorithm is called an inner one and its also a polynomial one. In 1989, appear a new inner algorithm due to Chang and Murty (1989). The algorithm was improved in 2006 (see Murty [2006, a,b]). Compared with the simplex, it is a border algorithm.

In the present paper we give an efficient method, namely the *sénégaulois algorithm*, based on the basic idea of Lavallé et al.(2011) which solve Klee-Minty LP instances in very short time, that is to say in sub-polynomial global iterations. Our method can be called outer-inner because we use a translation hyperplane, which can be a cutting plane or not (in a dichotomic argument). It gives an approached (or optimal) solution with a desired accuracy. The theoretical foundation for our approach appears in Lavallé et al.(2011) where

[2]Nevertheless during year 1965 an other soviet researcher has produced a polynomial [see Levin (1965)] but with a worry complexity of $O(n^6, m)$ but this result was ignored

algorithmic and geometric questions concerning dichotomy are studied and the vacuity test of the polytope is investigated.

Let us give a short overview of basic notions, polytopes, explicit descriptions of convex optimization and Hahn-Banach Theorems. More detailed account of the main aspects of our implementation are given in Section 3. In the following $\mathbb{N}$ is the set of integer numbers, $\mathbb{E}$ is a $\mathbb{R}$-vector space; where $\mathbb{R}$ is the set of real numbers. We call *Linear Programming* (LP) *problem* any problem which can be stated as follows

$$\max f(x) = \sum_{j=1}^{n} c_j x_j \quad c_j \in \mathbb{R},\ n \in \mathbb{N}^*$$

$$s.t. \begin{cases} \sum_{j=1}^{n} a_{ij} x_j & \leq b_i \\ x_j & \geq 0 \end{cases} \tag{1}$$

$$a_{ij}, b_i \in \mathbb{R}, i = 1, \ldots, m \subset \mathbb{N},$$

$$j = 1, \ldots, n \subset \mathbb{N}$$

Recall that convex polyhedron on $\mathbb{E}$ is a set P as:

$$P = \{x \in R^n :\ Ax \leq b\}, \tag{2}$$

where $A :\ \mathbb{E} \to\ \mathbb{R}$ is a linear application ($m \in \mathbb{N};\ if\ m = 0,\ P = \mathbb{E}$), $b \in \mathbb{R}^m$ and the inequality $Ax \leq b$ must be considered step by step (row by row) in $\mathbb{R}^m$ $(Ax)_i \leq b_i, \forall_i \in 1, \ldots, m$. Without loss of generality we call *polytope* a convex and bounded polyhedron; and we can restrict attention to inequality systems on the form $\{Ax \leq b\}$. In case of equalities as $Ax = d$, it is always possible to return to the form (2) by replacing an equality by two opposite inequalities $Ax \leq d$ and $-Ax \leq -d$. From a geometrical point of view, a polyhedron is the intersection of a finite number of hyperplanes on $\mathbb{E}$. If $\mathbb{E}$ is a finite-dimensional space, we can suppose that $\mathbb{E} = \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$. To apply Theorems 1 and 2, we define for the n the polyhedron:

$$P = \{x \in R^n :\ Ax = b,\ x \geq 0\}, \tag{3}$$

Recall that all polyhedron of the form (2) can be stated as (3) by introducing slack variables.

Remark 1. Representations (2) and (3) are called duals because they use linear applications of the dual set of the $\mathbb{E}$.

Consider polytope $\mathbb{D}$ (see Figure 1) which may contains the origin. This is always possible by introducing slack variables.

In the context of the separation theorem for locally convex spaces, attention was first devoted to separating hyperplane, convex optimization definitions and Hahn-Banach theorems, which will be used in the rest of this paper. Let f be a linear functional on $\mathbb{E}$, $\alpha \in \mathbb{R}$, $C_1 \subset \mathbb{E}$ and $C_2 \in \mathbb{E}$.

$$f(x) = \sum_{j=1}^{n} k_j x_j, \; n \in \mathbb{N}^*, \; k_j \in \mathbb{R}, \; x = (x_1, x_2, \dots, x_n)$$

Definition 1. *Consider two nonempty sets $C_1, \; C_2 \in \mathbb{R}^n$. C_1 and C_2 are separated by a hyperplane if and only if (iff) there exists a $\nu \in \mathbb{R}^n, \; \nu \neq 0$, and an $r \in \mathbb{R}$ such that $\nu.a \geq r \geq \nu.b$ for every $a \in C_1, \; b \in C_2$. C_1 and C_2 are strictly separated by a hyperplane iff there exists $\nu \neq 0$ and a $r \in \mathbb{R}$ such that $\nu.a > r > \nu.b$ for every $a \in C_1, \; b \in C_2$*

In Figure 2 the line drawn is a hyperplane strictly separating C_1 and C_2.

Definition 2. *A set $C \in \mathbb{R}^n$ is convex iff for any $x, \; y \in C$ and $\lambda \in [0, 1]$, the point $\lambda x + (1 - \lambda)y$ is also in C.*
In the geometrical point of view, C is convex iff C contains the line segment joining any two points in C:

$$[x, y] = [\lambda x + (1 - \lambda)y, \; 0 \leq \lambda \leq 1]$$

Definition 2. *We define hyperplane $\mathcal{H}$ with equation $[f = \alpha]$ the linear set:*

$$\mathcal{H} = \{x \in \mathbb{R}^n : \; f(x) = \alpha\} \tag{4}$$

Now we briefly recall some concepts, such as the **Hahn-Banach theorems**. The proofs can be obtained in Gilbert (2005-2006); Brezis (1999), and in Achmanov (1984) for polyhedron. These results are known in the literature as the Hahn-Banach separation theorems for locally convex spaces or separation theorem of convex sets. In the first one, it is possible to strictly separate two convex sets using two sets which one is closed and the other compact. In the second theorem, it is possible to separate (but not strictly) two ordinary convex sets of finite size.

Theorem 1(Hahn-Banach I.) *Let $\mathbb{E}$ be a finite-dimensional vector space with inner product $\langle ., . \rangle$. Let C_1 and $C_2 \subset \mathbb{E}$ two nonempty convex sets such as $C_1^{\infty} \cap C_2^{\infty} = \{0\}$ (where C_1^{∞} and C_2^{∞} are asymptotic cones of C_1 and C_2, respectively.) Then, it is possible to separate C_1 and C_2. There exists a vector $\xi \in \mathbb{E}$ such that:*

$$\sup_{x_1 \in C_1} \langle \xi, x_1 \rangle \leq \inf_{x_2 \in C_2} \langle \xi, x_2 \rangle$$

Corollary 1. *Suppose that C_1, $C_2 \subset \mathbb{E}$ are nonempty closed convex sets with C_1 compact, and C_2 closed. If $C_1 \cap C_2 = \emptyset$ then C_1 and C_2 are strictly separated by a hyperplane.*

Theorem 1(Hahn-Banach II.) *Let $\mathbb{E}$ be a finite-dimensional vector space with inner product $\langle ., . \rangle$. Let C_1 and $C_2 \subset \mathbb{E}$ two nonempty disjoint convex subsets. Then there exists a nonzero vector $\xi \in \mathbb{E} \setminus \{0\}$ such that:*

$$\sup_{x_1 \in C_1} \langle \xi, x_1 \rangle \leq \inf_{x_2 \in C_2} \langle \xi, x_2 \rangle$$

Corollary 2.
- *Let C be a nonempty closed convex set. $\forall u \in \partial C$ (the border of C), C is supported at u.If C is a convex set of a topological vector space, any point of the border of C is in a supporting hyperplane.*

- *Let C be a nonempty closed convex set, then*

$$C = \bigcap_{\prod \in p} \prod$$

where p is the set of the half space containing C.

As mentioned above, these theorems deal with the question of existence of a hyperplane that separates two given disjoint convex subsets.

The outline of this paper is as follows: In section 2 we describe the worrying instances of the simplex algorithm and review solution approaches for Klee-Minty's LP instances. In section 3 we describe the main ingredients of our algorithm. In section 4 we extensively test the implementation and the comparison on Klee-Minty's LP problem sets. We close with some conclusions in Section 5.

2 Worrying instances for the *symplex*

2.1 *Klee-Minty's LP problems*

In Klee and Minty (1969,1979); Liebling (1972) it is showed that some instances of *linear programming* for which Klee-Minty's LP requires an exponential number of iterations by Dantzig's simplex method. These instances are not isolated or specific ones, they are a class of problems and need specific treatment because the simplex computes in a great number of iterations.

Example 1 (Cycles in simplex)

$$Max\ z = 2x_1 + 3x_2 - x_3 - 12x_4$$
$$s.t. \begin{cases} -2x_1 - 9x_2 + x_3 + 9x_4 + x_5 & = 0 \\ \frac{1}{3}x_1 + x_2 - \frac{1}{3}x_3 - 2x_4 + x_6 & = 0 \\ -x_1 + x_2 & \leq 2 \end{cases} \tag{5}$$
$$x_i \geq 0,\ i = \overline{1,6}$$

With some patience we obtain the following cycle of basis.

$$B_0 = [5,6,7];\ B_1 = [2,5,7];\ B_2 = [1,2,7];\ B_3 = [1,4,7];\ B_4 = [3,4,7];$$
$$B_5 = [3,5,7]\ and\ B_6 = [5,6,7] = B_0 \tag{6}$$

The example (6) is not an isolated one, it is possible to show other similarity instances (see problems (7),(8) and (9)). The policy which use the Dantzig criteria is not optimal in terms of steps. The generalization of the example (6) shows a complete class of such problems (see Klee and Minty (1969, 1979)).

$$Max\ z = 2^{n-1}x_1 + \ldots + 2^2x_{n-2} + 2x_{n-1} + x_n$$
$$s.t. \begin{cases} x_1 & \leq 5 \\ 4x_1 + x_2 & \leq 25 \\ 8x_1 + 4x_2 + x_3 & \leq 125 \\ \vdots & \leq \vdots \\ 2^nx_1 + 2^{n-1}x_2 + \ldots + x_n & \leq 5^n \end{cases} \tag{7}$$

Geometrically this linear programming problem with n variables, is a polytope with 2^n nodes and none will be visited as basis of the simplex. A second Klee-Minty example is

$$Max\ z = \sum_{j=1}^{n} 10^{n-j} x_j$$

$$s.t. \begin{cases} 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i & \leq 100^{i-1} \qquad \forall\ 1 \leq i \leq n \\ x_j & \geq 0 \qquad \forall\ 1 \leq j \leq n \end{cases} \tag{8}$$

The simple variant of Klee-Minty's LP is much simpler than the original LP. It has an additional property that for any integer $k \in [0, 2^{n-1}]$ there exists exactly one basic feasible solution whose objective function value is k. The variant of Klee-Minty's LP is represented as:

$$Max\ z = \sum_{i=1}^{n} x_i$$

$$s.t. \begin{cases} x_i & \leq 1 \\ 2\sum_{i=1}^{k-1} x_i + x_k & \leq 2^k - 1 \qquad \forall\ 2 \leq k \leq n \\ x_i & \geq 0 \qquad \forall\ 1 \leq i \leq n \end{cases} \tag{9}$$

The standard form using a vector $y = (y_1, \dots, y_n)^T$ of slack variable is written as:

$$Max\ z = \sum_{i=1}^{n} x_i$$

$$s.t. \begin{cases} x_1 + y_1 & = 1 \\ 2\sum_{i=1}^{k-1} x_i + x_k + y_k & = 2^k - 1 \qquad \forall\ 2 \leq k \leq n \\ x_i,\ y_i & \geq 0 \qquad \forall\ 1 \leq i \leq n \end{cases} \tag{10}$$

The variant as well as Klee-Minty's LP has the following properties:

• for each $i \in \{1, 2, , \dots, n\}$ at any basic feasible solution, exactly one of x_i and y_i is a basic variable

• the problem has 2^n feasible solutions

• each component of any basic feasible solution is an integer, whose value is easily calculated from the equality constraints, because the constraint matrix is lower triangular

- the problem is nondegenerate

The optimal basic feasible solution of (10) is $x^* = (0,\ 0, \ldots,\ 0,\ 2^{n-1})^T$, $y^* = (1,\ 2^2-1, \ldots,\ 2^{n-1}-1, 0)^T$, and the optimal value is $2^n - 1$

2.2 Iterations of the simplex algorithm on the worrying instances

As a test set for models (7), (8) and (9) with the Dantzig's simplex method we use $n = 2, 3, 4, 5, 6, 12, 30$ and 50. Note finally that the size in numerical tests considered is not very large ($n \leq 50$); which allow to observe the behavior of our algorithm for this particular type of problems. Also, given the size of the objective function value, we restricted the choices which values not exceeding 1.0e38 (limits possible in C++ code) corresponding to $n = 50$, 19 for the first to instances. The same value $n = 50$ is tested for problem (9). As the number of iterations in simplex is exponential, we reported value corresponding to 2^n for $n \geq 12$. As an illustration, in the case $n = 6$, problem (7) becomes:

$$Max\ z = 32x_1 + 16x_2 + 8x_3 + 4x_4 + 2x_5 + x_6$$

$$s.t. \begin{cases} x_1 \leq 5 \\ 4x_1 + x_2 \leq 25 \\ 8x_1 + 4x_2 + x_3 \leq 125 \\ 16x_1 + 8x_2 + 4x_3 + x_4 \leq 625 \\ 32x_1 + 16x_2 + 8x_3 + 4x_4 + x_5 \leq 3125 \\ 64x_1 + 32x_2 + 16x_3 + 8x_4 + 4x_5 + x_6 \leq 15625 \end{cases}$$

In the case $n = 6$, problem (8) becomes problem (11).
Finally in the case $n = 6$ the problem (9) becomes

$$Max\ z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

$$s.t. \begin{cases} x_1 \leq 1 \\ 4x_1 + x_2 \leq 3 \\ 8x_1 + 4x_2 + x_3 \leq 7 \\ 16x_1 + 8x_2 + 4x_3 + x_4 \leq 15 \\ 32x_1 + 16x_2 + 8x_3 + 4x_4 + x_5 \leq 31 \\ 64x_1 + 32x_2 + 16x_3 + 8x_4 + 4x_5 + x_6 \leq 63 \end{cases}$$

Let us give meaning of the notation. We denoted by: dim=number of variables, $iter$=number of iterations of simplex, $objval$=objective function value and $solution$=optimal solution. Iterations and optimal values are both integers. Using the simplex method, problems (7) and (8) requires 2^n pivots to find an optimal solution; the stimulation results are given in Tables 1 and 2. Table 3 shows the results for problem (9).

Recall that there exist alternative solution approaches for solving these instances. Today's LP solvers are turned towards quickly finding an optimal solution of a feasible LP. Sometimes their bases are not really optimal, but this has only a negligible effect on the objective function value, see Koch (2004). When checking infeasibility, however, small errors can lead to completely wrong decisions. The answer depends on the particular instance, the solution method of the LP solver, its parameters, e.g., the precision (usually around 1.0e-06), and often also the preprocessing and starting basis.

In the following section we will discuss a way to solve Klee-Minty's LP instances via the sénégaulois algorithm and present computational results for different data sets.

3 Sénégaulois algorithm

The main approach of this algorithm appears in Lavallé et al.(2011), where the algorithm is deeply studied. We add a hyperplane which separates the whole space (one containing the solution), and it consists on restricting more and more the space containing solutions, as done by Khachiyan (1979), with the hyperplane rather than ellipsoïde. Our algorithm, namely *sénégaulois algorithm*, solves linear programs of the form (1).

3.1 The sketch of the algorithm Lavallé et al.(2011)

Let $\mathcal{H}$ be a hyperplane defined by (4) and K be the polytope defined by the set of constraints of problem (1). Since f is a linear functional, then $\mathcal{H}$ is a convex set. Here we are looking for separation of convex sets not by just arbitrary hyperplanes but by closes hyperplanes. This means that the associated linear functionals must be continuous. Nevertheless, since f is a continuous function so $\mathcal{H}$ is closed hyperplane.

Using the separation theorem, every such hyperplane $\mathcal{H}$ divides the whole space in two convex sets:

$$\mathcal{H}_l = \{x \in \mathbb{R}^n \ : \ f(x) \leq \alpha\}$$

and

$$\mathcal{H}_r = \{x \in \mathbb{R}^n \ : \ f(x) \geq \alpha\}$$

known as *halfspaces*. On part contains the polytope K defined from the constraints in (4).

Then the following cases hold:

1. Hyperplane $\mathcal{H}$ does not divide the whole polytope K of constraints, then $\mathcal{H}$ is outside the feasible solution, *i.e.* $\mathcal{H} \cap K = \emptyset$ ($\mathcal{H} \cap K$ does not contain any solution). In such case, it is necessary to choose another hyperplane $\mathcal{H}' // \mathcal{H}$ (*parallel*) such that $\mathcal{H}'$ lies between the origin 0 and $\mathcal{H}$

2. Hyperplane $\mathcal{H}$ divides the whole polytope K of constraints, then there is non empty intersection between $\mathcal{H}$ and K, *i.e.* $\mathcal{H} \cap K \neq \emptyset$. In this situation, the problem contains some feasible solutions and K is separated by $\mathcal{H}$. If the stopping test fails, then a new separation hyperplane $\mathcal{H}^* // \mathcal{H}$ is chosen such that $\mathcal{H}^*$ lies in the half space which does not contain the origins 0.

3. Polytope K is empty, the problem has no solution.

$$\begin{aligned} &Max\ z = 100000x_1 + 10000x_2 + 1000x_3 + 100x_4 + 10x_5 + x_6 \\ s.t. &\begin{cases} x_1 \leq 1 \\ 20x_1 + x_2 \leq 1e+02 \\ 200x_1 + 20x_2 + x_3 \leq 1e+04 \\ 2000x_1 + 200x_2 + 20x_3 + x_4 \leq 1e+06 \\ 20000x_1 + 2000x_2 + 200x_3 + 20x_4 + x_5 \leq 1e+08 \\ 200000x_1 + 20000x_2 + 2000x_3 + 200x_4 + 20x_5 + x_6 \leq 1e+10 \end{cases} \end{aligned} \tag{11}$$

Table 1

A Klee-Minty example (7)

Number	dim	iter	objval	solution
1	2	4	25	$(x_1 = 0,\ x_2 = 25)$
2	3	8	125	$(x_1 = x_2 = 0,\ x_3 = 125)$
3	4	16	625	$(x_1 = x_2 = x_3 = 0,\ x_4 = 625)$
4	5	32	3125	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 3125)$
5	6	64	15625	$(x_1 = x_2 = x_3 = x_4 = x_5 = 0,\ x_5 = 15625)$
6	12	4096	2.44140625e+08	$(x_i = 0\ \forall i = 1, \ldots,\ 11, x_{12} = 2.44140625e + 08)$
7	20	1.048576e+06	9.5367431640625e+13	$(x_i = 0\ \forall i = 1, \ldots,\ 19, x_{20} = 9.5367431640625e + 13)$
8	30	1.073741824+09	9.31322574615479e+20	$(x_i = 0\ \forall i = 1, \ldots,\ 29, x_{30} = 9.31322574615479e + 20)$
9	50	1.12589990684262e+15	8.88178419700125e+34	$(x_i = 0\ \forall i = 1, \ldots,\ 49, x_{50} = 8.88178419700125e + 34)$

Table 2

Second Klee-Minty example (8)

Number	dim	iter	objval	solution
1	2	4	1e+02	$(x_1 = 0,\ x_2 = 1e + 02)$
2	3	8	1e+04	$(x_1 = x_2 = 0,\ x_3 = 1e + 04)$
3	4	16	1e+06	$(x_1 = x_2 = x_3 = 0,\ x_4 = 1e + 06)$
4	5	32	1e+08	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 1e + 08)$
5	6	64	1e+10	$(x_1 = x_2 = x_3 = x_4 = x_5 = 0,\ x_5 = 1e + 10)$
6	12	4096	1e+22	$(x_i = 0\ \forall i = 1, \ldots,\ 11, x_{12} = 1e + 22)$
7	19	5.24288e+05	1e+36	$(x_i = 0\ \forall i = 1, \ldots,\ 18, x_{19} = 1e + 36)$

Table 3

Simple variant Klee-Minty's LP (7)

Number	dim	iter	objval	solution
1	2	4	3	$(x_1 = 0,\ x_2 = 3)$
2	3	6	7	$(x_1 = x_2 = 0,\ x_3 = 7)$
3	4	10	15	$(x_1 = x_2 = x_3 = 0,\ x_4 = 15)$
4	5	16	31	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 31)$
5	6	26	63	$(x_1 = x_2 = x_3 = x_4 = x_5 = 0,\ x_5 = 63)$
6	12	466	4.095e+3	$(x_i = 0\ \forall i = 1, \ldots,\ 11, x_{12} = 4.095e + 3)$
7	20	2.1892e+04	1.048575e+06	$(x_i = 0\ \forall i = 1, \ldots,\ 19, x_{20} = 1.048575e + 06)$
8	30	6.35626e+05	1.073741823e+09	$(x_i = 0\ \forall i = 1, \ldots,\ 29, x_{30} = 1.073741823e + 09)$
9	50	1.655698e+06	1.12589990684262e+15	$(x_i = 0\ \forall i = 1, \ldots,\ 49, x_{50} = 1.12589990684262e + 15)$

3.2 The algorithm

Suppose $\mathcal{H}\ :\ f(x) = \alpha$ and K the initial polytope defined by the set of constraints of problem (1). If $\mathcal{H} \cap K = \emptyset$, then we state $\alpha := \frac{\alpha}{2}$. That is to say that we divide the gap between 0 and α into two parts. It is the main idea of the algorithm.
Now suppose that we have two values of α denoted by α_0 and α_1 such that:

- $\mathcal{H}\ :\ f(x) = \alpha_0$ *and* $\mathcal{H} \cap K \neq \emptyset$
- $\mathcal{H}\ :\ f(x) = \alpha_1$ *and* $\mathcal{H} \cap K = \emptyset$

Then we take a new value of

$$\alpha = \frac{\alpha_0 + \alpha_1}{2} \tag{12}$$

In addition, two cases are possible:
1. $\mathcal{H}\ :\ f(x) = \alpha_0$ and $\mathcal{H} \cap K = \emptyset$, then put $\alpha_1 := \alpha$
2. $\mathcal{H}\ :\ f(x) = \alpha_0$ and $\mathcal{H} \cap K \neq \emptyset$, then put $\alpha_0 := \alpha$
And so on.

Remark 2. Notation $i := i + 1$ which has no sense in mathematical field is a programming language notation which means that the memory cell value i, at which the value 1 is added, is assigned to the memory cell i.

In this algorithm the hyperplane defined by:

$$\mathcal{H} = \left\{ x \in \mathbb{R}^n \ : \ \sum_{i=1}^{n} c_i x_i = \alpha \right\} \tag{13}$$

where $\sum_{i=1}^{n} c_i x_i$ is the objective function (the one to maximize in our purpose) of the LP.

Remark 3. From a geometrical point of view, at each step of the algorithm, the new hyperplane is obtained from the previous one by the translation. Thus, we search vectors that characterizes polytope $\mathcal{H}$ defined bye (13); then we search an unitary vector $\overrightarrow{\nu}$ as vectorial step.

3.3 A key subproblem

In the general sénégaulois algorithm, a major problem necessary to solve is the one of the test of the intersection between the hyperplane $\mathcal{H}$ and the polytope generated by constraints. Here we solve this as follows:
When the separating hyperplane $\mathcal{H}_p$ is chose, the step of the algorithm consists to launch an iteration of the simplex algorithm on the new instance. If there is no solution, we choose a new hyperplane $\mathcal{H}_{p+1}$ between $\mathcal{H}_{p-1}$ and $\mathcal{H}_p$.

3.4 The Stopping Test

As stated above, our algorithm works with successive approximations but the result is obtained with desired accuracy[3]. That is to say that it is the operator who gives the acceptable tolerance. Suppose that margin tolerance is $\epsilon \in \mathbb{Q}$, with equation notations in (12), then the stopping test is:

$$|\alpha_o - \alpha_1| \leq \epsilon \tag{14}$$

[3] It is also the case of the Katchiyan's, Karmarkar's, and Murty's algorithms. And the simplex algorithm is also an approximate one due to the limitation of the coding numbers on computers.

3.5 To Provide a Solution

When both $\mathcal{H} \cap K \neq \emptyset$ and $|\alpha_o - \alpha_1| \leq \epsilon$ are true, a feasible solution is obtained in the part of the polytope as we can see it in Figure 3.

Geometrical explanations

Unless we have obtained the *optimum*, the hyperplane $\mathcal{H}_0 \ : \ f(x) = \alpha$ separate the other constraints. Basis are vertices of the residual polytope. There are n such vertices associated with the separating hyperplane satisfying the constraints. But all interior points of the residual polytope satisfy the constraints and are in the permitted interval. All convex combination of these points is a valid solution[4].

4 The improvement of the simplex

In this section, we describe our algorithm for solving instances (7), (8) and (9). We also conducted experiments with the same data used in Section 2.

4.1 Transformation of the worrying instances

Before presenting computational results, we want to present the influence of the separating hyperplane

$$\mathcal{H} = \left\{ x \in \mathbb{R}^n \ : \ \sum_{i=1}^{n} c_i x_i = \alpha \right\}$$

The basic question is whether $\mathcal{H} \cap K$ is empty or not for a given best-first value α_0.

As an illustration, for problem (9), we add a hyperplane of the form $\sum_{i=1}^{n} x_i = \alpha$. we obtain the desired problem:

$$Max\ z = \sum_{i=1}^{n} x_i$$

[4]It's true because the set of solutions is convex.

$$s.t. \begin{cases} x_i & \leq 1 \\ 2\sum_{i=1}^{k-1} x_i + x_k & \leq 2^k - 1 \qquad \forall\ 2 \leq k \leq n \\ \sum_{i=1}^{n} x_i & = \alpha \\ x_i & \geq 0 \qquad \forall\ 1 \leq i \leq n \end{cases} \tag{15}$$

For $n = 2$ and $n = 16$, we obtain:

$$Max\ z = x_1 + x_2$$

$$s.t. \begin{cases} x_1 & \leq 1 \\ 2x_1 + x_2 & \leq 3 \\ x_1 + x_2 & = 16 \\ x_1 \geq 0,\ x_2 \geq 0 & \end{cases} \tag{16}$$

Table 4

Sénégaulois algorithm for solving problem (7)

Nb	dim	iter	objval	solution	α_0
1	2	3	25	$(x_1 = 0,\ x_2 = 25)$	100
2	2	4	25	$(x_1 = 0,\ x_2 = 25)$	40
3	2	11	24.951	$(x_1 = 0.0245,\ x_2 = 24.902)$	70
4	3	4	125	$(x_1 = x_2 = 0,\ x_3 = 125)$	200
5	3	12	124.966	$(x_1 = 0,\ x_2 = 0.017,\ x_3 = 124.932)$	190
6	3	16	124.955	$(x_1 = 0,\ x_2 = 0.025,\ x_3 = 124.99)$	190
7	4	4	625	$(x_1 = x_2 = x_3 = 0,\ x_4 = 625)$	1000
8	4	15	624.994	$(x_1 = x_3 = 0,\ x_2 = 0.0015,\ x_4 = 624.988)$	1100
9	5	4	3125	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 3125)$	5000
10	5	18	3124.992	$(x_1 = x_3 = x_4 = 0,\ x_2 = 0.001,\ x_5 = 3124.984)$	9000
11	6	4	15625	$(x_1 = x_2 = x_3 = x_4 = x_5 = 0,\ x_6 = 15625)$	25000
12	6	21	15624.99	$(x_1 = x_3 = x_4 = x_5 = 0,\ x_2 = 0.000625,\ x_6 = 15624.98)$	70000
13	12	40	244140625	$(x_i = 0\ \forall i = 1,\ldots,11,\ x_{12} = 244140625)$	2.9e+08
14	12	34	244140624.977	$(x_i = 0\ \forall i = 1,3,\ldots,11,\ x_2 = 0.000022,\ x_{12} = 244140624.954)$	5.2e+08
15	12	40	244140625	$(x_i = 0\ \forall i = 1,\ldots,11,\ x_{12} = 244140625)$	5.2e+08
16	20	52	9.5367431640625e+13	$(x_i = 0\ \forall i = 1,\ldots,19,\ x_{20} = 9.5367431640625e+13)$	1e+14
17	20	48	9.5367431640625e+13	$(x_i = 0\ \forall i = 1,\ldots,19,\ x_{20} = 9.5367431640625e+13)$	1.5e+14
18	30	52	9.31322574615479e+20	$(x_i = 0\ \forall i = 1,\ldots,29,\ x_{30} = 9.31322574615479e+20)$	1.8+21
19	30	52	9.31322574615479e+20	$(x_i = 0\ \forall i = 1,\ldots,29,\ x_{30} = 9.31322574615479e+20)$	1e+23
20	50	51	8.88178419700125e+34	$(x_i = 0\ \forall i = 1,\ldots,49,\ x_{50} = 8.88178419700125e+34)$	9e+34
21	50	52	8.88178419700125e+34	$(x_i = 0\ \forall i = 1,\ldots,49,\ x_{50} = 8.88178419700125e+34)$	2e+35
22	50	59	8.88178419700125e+34	$(x_i = 0\ \forall i = 1,\ldots,49,\ x_{50} = 8.88178419700125e+34)$	1e+37
23	50	63	8.88178419700125e+34	$(x_i = 0\ \forall i = 1,\ldots,49,\ x_{50} = 8.88178419700125e+34)$	1e+38

Table 5

Sénégaulois algorithm for solving problem (8)

Nb	dim	iter	objval	solution	α_0
1	2	4	1e+02	$(x_1 = 0,\ x_2 = 1e + 02)$	160
2	2	13	99.998	$(x_1 = 0,\ x_2 = 99.996)$	270
3	3	2	1e+05	$(x_1 = x_2 = 0,\ x_3 = 1e + 05)$	2e+05
4	3	4	1e+05	$(x_1 = x_2 = 0,\ x_3 = 1e + 05)$	1.6e+04
5	4	2	1e+06	$(x_1 = x_2 = x_3 = 0,\ x_4 = 1e + 06)$	2e+06
6	4	15	999999.979	$(x_1 = x_2 = 0,\ x_3 = 0.0021,\ x_4 = 999999.958)$	1.9e+06
7	5	2	1e+08	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 1e + 08)$	2e+08
8	5	4	1e+08	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 1e + 08)$	1.6e+08
9	5	22	99999999.988	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 99999999.988)$	3.5e+08
10	6	2	1e+10	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 1e + 10)$	1e+10
11	6	4	1e+10	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 1e + 10)$	1.6e+10
12	6	42	9999999999.941	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 9999999999.941)$	1.9e+10
13	12	7	1e+22	$(x_i = 0\ \forall i = 1, \ldots, 11,\ x_{12} = 1e + 22)$	5e+22
14	12	9	1e+22	$x_i = 0\ \forall i = 1, \ldots, 11,\ x_{12} = 1e + 22)$	1.7e+23
15	12	19	1e+22	$(x_i = 0\ \forall i = 1, \ldots, 11,\ x_{12} = 1e + 22)$	1e+26
16	19	6	1e+36	$(x_i = 0\ \forall i = 1, \ldots, 18,\ x_{19} = 1e + 36)$	3e+36
17	19	9	1e+36	$(x_i = 0\ \forall i = 1, \ldots, 18,\ x_{19} = 1e + 36)$	8e+36
18	19	11	1e+36	$(x_i = 0\ \forall i = 1, \ldots, 18,\ x_{19} = 1e + 36)$	5e+37

We also conducted experiments with the same data n =2, 3, 4, 5, 6, 12, 20, 30 and 50 for the problems (7) and (8).

4.2 Numerical simulations

In the following we will discuss computational results of our algorithm implementation for the Klee-Minty's LP problems. The algorithm was implemented in C++ and computations were performed on a 2 GHz 2×Intel(R) Core(TM)2 Duo machine with 4 GB of main memory and 2MB cache running Linux. We experiments with solving the problem sets (7), (8) and (9) by our approach and then comparing with the simplex algorithm. Simulation results for problem (7),(8) and (9) are illustrated in Table 4,5 and 6, respectively. The notation is as in Tables 1, 2 and 3 of Section 2.2.

Two stopping tests are possible: (a) $\alpha_{k+1} = \alpha_k$ (this is the case where the value of $\alpha = \alpha_r - \alpha_l$ is exactly equal to α_{opt}), (b) with accuracy ϵ well selected, an approximate solution or even optimal is provided.

Table 6

Sénégaulois algorithm for solving problem (9)

Nb	dim	iter	objval	solution	α_0
1	2	2	3	$(x_1 = 0,\ x_2 = 3)$	6
2	2	5	3	$(x_1 = 0,\ x_2 = 3)$	16
3	3	2	7	$(x_1 = x_2 = 0,\ x_3 = 7)$	14
4	3	5	7	$(x_1 = x_2 = 0,\ x_3 = 7)$	16
5	3	9	6.934	$(x_1 = 0,\ x_2 = 0.066,\ x_3 = 6.868)$	25
6	4	2	15	$(x_1 = x_2 = x_3 = 0,\ x_4 = 15)$	30
7	4	3	15	$(x_1 = x_2 = x_3 = 0,\ x_4 = 15)$	20
8	4	4	15	$(x_1 = x_2 = x_3 = 0,\ x_4 = 15)$	120
9	5	2	31	$(x_1 = x_2 = x_3 = x_4 = 0,\ x_5 = 31)$	62
10	5	12	30.996	$(x_1 = x_3 = x_4 = 0,\ x_2 = 0.004,\ x_5 = 30.992)$	120
11	6	2	63	$(x_1 = x_2 = x_3 = x_4 = x_5 = 0,\ x_6 = 63)$	126
12	6	12	62.959	$(x_1 = x_3 = x_4 = x_5 = 0,\ x_2 = 0.041,\ x_6 = 62.918)$	140
13	6	17	62.943	$(x_1 = x_3 = x_4 = x_5 = 0,\ x_2 = 0.057,\ x_6 = 62.886)$	5e+03
14	6	24	63	$(x_1 = x_2 = x_3 = x_4 = x_5 = 0,\ x_6 = 63)$	5e+03
15	12	14	4.095e+3	$(x_i = 0\ \forall i = 1, \ldots, 11,\ x_{12} = 4.095e + 3)$	6e+03
16	12	17	4.095e+3	$(x_i = 0\ \forall i = 1, \ldots, 11,\ x_{12} = 4.095e + 3)$	1.5e+04
17	20	25	1.048575e+06	$(x_i = 0\ \forall i = 1, \ldots, 19,\ x_{20} = 1.048575e + 06)$	1.8e+06
18	20	25	1.048575e+06	$(x_i = 0\ \forall i = 1, \ldots, 19,\ x_{20} = 1.048575e + 06)$	7e+06
19	20	38	1.048575e+06	$(x_i = 0\ \forall i = 1, \ldots, 19,\ x_{20} = 1.048575e + 06)$	1e+10
20	30	35	1.073741823e+09	$(x_i = 0\ \forall i = 1, \ldots, 29,\ x_{30} = 1.073741823e + 09)$	5e+09
21	30	54	1.073741823e+09	$(x_i = 0\ \forall i = 1, \ldots, 29,\ x_{30} = 1.073741823e + 09)$	1e+13
22	50	49	1.12589990684262e+15	$(x_i = 0\ \forall i = 1, \ldots, 49,\ x_{50} = 1.12589990684262e + 15)$	3.5e+15
23	50	62	1.12589990684262e+15	$(x_i = 0\ \forall i = 1, \ldots, 49,\ x_{50} = 1.12589990684262e + 15)$	2e+20
24	50	81	1.12589990684262e+15	$(x_i = 0\ \forall i = 1, \ldots, 49,\ x_{50} = 1.12589990684262e + 15)$	1e+25

Table 7
Simplex and sénégaulois algorithms

		simplex			sénégaulois	
dim	(7)	(8)	(9)	(7)	(8)	(9)
2	4	4	4	3	4	2
3	8	8	4	4	2	2
4	16	16	6	4	2	2
5	32	32	10	4	2	2
6	64	64	16	4	2	2
12	4096	4096	26	34	7	14
19	-	524288	-	-	6	-
20	1.048576e+06	-		48	-	25
30	1.073741824e+09	-		52	-	35
50	1.12589990684262e+15	-		51	-	49

First, for very small epsilon, one can obtain the optimal solution. Second, for a choice of α such that at iteration k $\alpha_{opt} = \alpha_k$ (exactly), we obtain the optimal value for all ϵ. It is the latter case which is found in several examples of such Klee-Minty's LP instances. It is why, we give a very small value of ϵ in the range of 1.0e-03 to 1.0e-06. Nevertheless, we display the value of ϵ for which this solution is obtained for $\alpha_{opt} = \alpha_k$.

One varies the value of α and discuss the influence of this parameter. We begin with the value α_0: the first value of α. The value α_0 is arbitrary selected without a prior idea of the optimal value of f. Solutions found by our approach have several properties.

For problem (7): it can be observed that we could compute optimal solutions for examples 1, 2, 4, 7, 9, 11, 13 and 15-23. On the other side examples 3, 5, 6, 8, 10, 12 and 14 gives approximate solutions for each run, which can be improved to optimal solution with a value of epsilon much smaller. The corresponding iterations are not exponential. The variation of ϵ gives an improvement of the solution as in examples 5 and 6. We choose three values of α_0 (100, 40 and 70) for n =2. The first two examples give the same solution as in Table 4, but our algorithm makes fewer iterations that the simplex. For the same accuracy, we obtain an approximate solution with 11 iterations. With n =3 and α_0=200, we obtain four the simplex algorithm found eight. On the other side, certain values of α_0 give an approximate solution. For $n \geq 4$, all combinations give a number iterations less than the simplex. For $n = 50$ (number 23) even a significant increase of α_0 does not greatly increase the number of iterations.

For problem (8): it can be expected that optimal solutions are found for examples 1, 3, 4, 5, 7, 8, 10, 11 and 13-18 compared for example 2, 6, 9 and 12 where approximate solutions are computed. For $n \leq 6$ solutions are computed with $\epsilon = 1.0e - 01$. Our algorithm can get four or two times less iterations than the simplex algorithm, see Table 5.

For problem (9): tests with $n \leq 6$ are carried out with $\epsilon = 1.0e - 01$ excepted number 14. One varies the tolerance for obtaining good approximate solution (or optimal one) and setting the smallest possible value. It can be observed that an optimal solution is found with $\epsilon = 1.0e - 03$. It can be noticed that optimal solutions are computed for examples 1, 2, 3, 4, 6, 7, 8, 9, 11 and 14-24; while approximate solutions are found for examples 5, 10, 12 and 13. Then, significant increase of α_0 does not greatly increase the number of iterations for $n = 30$ and 50. With $n = 6$, so even a very large value of α_0 provides an optimal solution with less iterations compared to the simplex algorithm.

Our results are, however, comparable to the ones shown in Section 2.2 for simplex algorithm to solve the Klee-Minty's instances. Table 7 compares the number of iterations of the different methods, where (7), (8) and (9) represent problem (7), problem (8) and problem (9) respectively. It displays only problems for which the optimal solutions could be found. The smallest resulting values are only reported. It turns out that all variants find the same final primal solutions. However, our approach is faster compared to the simplex algorithm. We can reduce the number of iterations with respect to α_0.

We conclude that the complete problem sets could be solved to optimality with small number of iterations.

5 Conclusions

In this paper we described the sénégaulois algorithm implementation of the Klee-Minty's LP problem sets. With respect to the problem data, the considered instances can be solved to optimality with very short iterations. Of course, various sizes instead of $n \leq 6$ can be taken in order to obtain large set of problems. An interesting open issue is whether the vacuity test could improve the performance of the implementation for very large scale problems.

6 Acknowledgments

Our acknowledgments to both French CNRS-institute INS2I and INSIS and for OSD chair (CNRS-LIX-Microsoft) which support this work.

7 References

Achmanov, S., 1984, Programmation Linéaire. Édition Mir. Traduit de Russe, édition Russe 1981.

Akoa, B.F., 2005. Interior-Point Approaches and DC Programming for nonconvex optimization, PhD Thesis, Rouen, National Institute for Applied Sciences, Rouen, France.

Brezis, H., 1999, Analyse fonctionelle: théorie et applications. Dunod, Paris, ISBN : 2-10-049336-1.

Chang, S.Y., Murty, K.G., 1989, The Steepest Descent Gravitational Method for Linear Programming, Discrete Applied Mathematics 25, 211-239.

Dantzig, G.B., 1951, Minimization of linear function of a variables subject to linear inequalities. In Koopman, T.C. editor, Activity, Analysis of production and allocation, John Wiley, New York, 339-347.

Gilbert, J.C., 2005-2006, Optimization différentielle: Théorie et algorithmes. École Nationale Supérieure des Techniques Avancées, Cours.

Karmarkar, N., 1984, A New Polynomial-Time Algorithm for Linear Programming, STOC, 1(11), 302-311

Khachiyan, L.G., 1979, A polynomial algorithm in linear programming. En russe, translation in Soviet Mathematics Doklady 20:1 (1979), 191-194. Doklady Akademii Nauk SSSR, novaia seria, 244 (5), 1093-1096

Klee, V., Minty, G.J., 1972, How Good is the Simplex Algorithm? In O.Shisha editor, Inequalities III, Academic press, New York, 159-175.

Klee, V., Minty, G.J., 1972, How Good is the Simplex Algorithm? In Shisha, Oved. editor, Inequalities III, Proceedings of the Third Symposium on Inequalities held. University of California.

Koch, T., 2004. The final Netlib-LP results. Operation Research Letter 32, 138-142.

Lavallée, I., Ndiaye, B.M.,Seck, D., 2011. A New Way in Linear Programming. In LISS, Logistics Management 2, 220-227.

Levin, A.Y., 1965. On an algorithm for the minimization of convex functions. Soviet Mathematics Doklady 6, 286-290.

Liebling, T.M., 1972. On the number of iterations of the simplex method. Methods of Operations Research XVII, 248-264.

Murty, K.G., 2006. A New Practically Efficient Interior Point Method for LP. Algorithmic Operations Research 1, 3-19.

Murty, K.G., 2006. Linear equations, inequations, LPs, and an efficient new algorithm. In INFORMS 2006, Tutorial, 3-19.

Roos, C., Terlaky, T., Vial, J. Ph., 1997. Theory and algorithms for linear optimization: an interior point approach. Discrete Mathematics and Optimization, Wiley-Interscience Series.

Terlaky, T., Zhang, S., 1993. Pivot rules for linear programming - a survey. Annal of Operation Research 46, 203-233.

Wachter, A., Biegler, L.T., 2006. On the implementation of a primal-dual interior point filter line search algorithm for large scale nonlinear programming. Mathematical Programming 106 (1), 25-27.

Wright, S.J., 1997. Primal-Dual Interior-Point Methods. SIAM Publications.

Ye, Y., 1997. Interior-Point Algorithms: Theory and Analysis. Discrete Mathematics and Optimization, Wiley-Interscience series.