A Feasible-Point Sagitta Approach in Linear Programming

Á. Santos Palomo [1] and P. Guerrero García [2]

Abstract. A second scheme for the *sagitta* method is presented. This method uses a "global" viewpoint of the linear problem, and, in this feasible-point version, it also takes advantage of the additional "local" information that a feasible point supplies. The computational results obtained are highly encouraging.

Key Words. Linear programming, active set methods, range space methods.

1. Introduction

We consider the linear problem

$$(ILP) \qquad \underset{x \in R^n}{Minimize} \qquad \ell(x) = c^T x$$
$$subject\ to \qquad A^T x \geq b$$

where A is an $n \times m$ matrix. The condition $c \neq \mathbf{0}$ is added and we use a_i^T to denote the ith row of A^T.

A first scheme of the *sagitta* method for solving this problem has been recently presented [1]. This new method, which is an intent to resolve the "myopia" of the *simplex* method [2, 3, 4, 5, 6], is similar to non-simplex active-set methods [7, 4], but it has innovative characteristics.

The *sagitta* method attempts to determine a set of constraints possibly active at an optimal solution of the (ILP), using a global viewpoint of the problem. The first scheme does not use an initial feasible point and it works with a candidate set (which we call *foreactive set*) and a corresponding *null-space descent direction.* The constraint addition to the foreactive set attempts initially to determine if a direction of the feasible region exists and, hence, if the (ILP) problem is unbounded below. When there is not a null-space descent

[1] Dpt. Applied Mathematics, Univ. of Málaga, 29071 Málaga (Spain), santos@ctima.uma.es
[2] Dpt. Applied Mathematics, Univ. of Málaga, 29071 Málaga (Spain), pablito@ctima.uma.es

direction, an iteration point is determined (usually exterior to the feasible region) and, also, a multiplier vector associated. Then, the method continues undertaking a primal-feasibility search loop, by modifying conveniently the foreactive set and trying that, if the iteration point is feasible, the associated multiplier vector is nonnegative.

This work presents a feasible-point *sagitta* approach that tries, maintaining this previous basic methodology, to take advantage of the "local" information upon the feasible region that a feasible point supplies. As feasible-point or primal method, the method steps down the feasible point whenever possible.

This work is organized as follows. In the following section we sum up basic preliminary results. In section 3 the second scheme of the *sagitta* method and relative formulae are provided, as well as a convergence theorem. In section 4 some linear programs are solved as an example. A particular range space implementation of the algorithm is described briefly in section 5. Then in section 6 we give the results of comparative computational tests carried out, and finally, a summary of remarks and conclusions are presented in section 7.

2. Preliminaries

Basically, the *sagitta* method is an active-set method since it tries to determine a linearly independent subset $\mathcal{A}_*$ of the active set $\mathcal{A}(x^*)$ of active constraints at an optimal solution x^* of the (ILP) problem, working with a sequence of candidate sets $\mathcal{A}_j$, $\mathcal{A}_j \subseteq \mathcal{M}$ for $\mathcal{M} = \{1,\ 2,\ \ldots,\ m\}$. However, the *sagitta* method presents some innovative characteristics.

Commonly, an active-set method generates a sequence of pairs $(x^{(j)}, \mathcal{A}_j)$. The set $\mathcal{A}_j$ (usually called active-set or working-set) is a set of constraint indices with two properties:

(i) if constraint p is in $\mathcal{A}_j$, constraint p is active at the iteration point $x^{(j)}$;

(ii) the normal vectors of constraints in $\mathcal{A}_j$ are linearly independent.

Well now, each set $\mathcal{A}_j$ defined by the *sagitta* method (which we call *foreactive set*) is a set of constraint indices with only property (ii). And, although in a feasible-point approach this new method has available a feasible solution $x^{(j)}$, non-active constraints at $x^{(j)}$ are added to the set $\mathcal{A}_j$. A foreactive set $\mathcal{A}_j$ is not required to contain n indices.

As in a general active-set method, the constraints in the foreactive set are treated as equalities in order to choose, if it is possible, a *null-space descent direction* $d^{(j)}$. That is to say, $d^{(j)}$ as null-space direction satisfies

$$A_j^T d^{(j)} = \mathbf{0} \tag{1}$$

and, as descent direction, it is such that

$$c^T d^{(j)} < 0 \tag{2}$$

where A_j is the *foreactive-set matrix*. However, no movement takes place along the direction $d^{(j)}$ in the first scheme of the *sagitta* method [1]. But, in a feasible-point sagitta approach, we consider of interest (not essential) such a movement along $d^{(j)}$. The interest is twofold: firstly, the approximation of the feasible point to an optimal solution, and secondly, the availability –as far as possible– of a varying boundary information. So, if the null-space descent direction $d^{(j)}$ is a feasible direction at $x^{(j)}$, regardless of the way in which the foreactive set $\mathcal{A}_j$ has been determined, a movement takes place from the iteration $x^{(j)}$ to a new iteration $x^{(j+1)}$ defined by

$$x^{(j+1)} = x^{(j)} + \alpha_j d^{(j)}$$

for a non-negative step length α_j.

The *sagitta* method selects the constraints to be added to the current foreactive set $\mathcal{A}_j$ from among the *contrary constraints* to the current direction $d^{(j)}$, where a *contrary constraint* to the direction $d^{(j)}$ is such that

$$a_i^T d^{(j)} < 0. \tag{3}$$

Moreover, even though the *sagitta* method tries to determine an active set $\mathcal{A}_*$ at an optimal solution of the (ILP), the candidate foreactive set $\mathcal{A}_j$ can be such that the corresponding null-space descent direction $d^{(j)}$ is feasible for all the constraints, that is to say, such that

$$a_i^T d^{(j)} \geq 0 \qquad \text{for all } i \in \mathcal{M}$$

or, in other terms, a null-space descent direction $d^{(j)}$ such that it is a *direction of the feasible region* (see [2, p. 82] or [3, p. 58]). Then, it is possible to conclude that the objective function is unbounded below.

Theorem 2.1 *Consider the (ILP) problem and suppose that there is a feasible solution. If a descent direction d exists such that*

$$A^T d \geq \mathbf{0},$$

then the objective function is unbounded below in the feasible region.

Proof: It is straightforward.□

Well now, if the (ILP) has an optimal solution, the determination of a null-space descent direction corresponding to a foreactive set $\mathcal{A}_j$ not always will be possible. But it is well-known that if we have no such direction, as the foreactive-set matrix A_j has full column rank, the system

$$A_j \mu = c \tag{4}$$

is compatible (see, for example, [4, p. 377]). Also, it is compatible the set of equations

$$a_i^T x = b_i \text{ for all } i \in \mathcal{A}_j,$$

or, equivalently, the system

$$A_j^T x = b^{(j)} \tag{5}$$

where $b^{(j)}$ is the subvector of b with elements b_i for all $i \in \mathcal{A}_j$. Then, the resolution of the systems (4) and (5) provide another iteration point $\hat{x}^{(j)}$, usually exterior to the feasible set, and a multiplier vector $\mu^{(j)}$ associated with it.

If the system (5) is underdetermined, alternative options for the iteration point $\hat{x}^{(j)}$ are the minimum norm solution of (5) or the solution of (5) closest to $x^{(j)}$, that is to say, the solution of the least-distance problem

$$\begin{array}{ll} \underset{x\in R^n}{Minimize} & \| x^{(j)} - x \|_2 \\ subject\ to & A_j^T x = b^{(j)}. \end{array}$$

The following theorem characterizes the minimum norm solution of the system (5).

Theorem 2.2 *Suppose that B is an $n \times k$ matrix with full column rank and that the system $B^T t = v$ is underdetermined. Then a solution t^* of $B^T t = v$ is the minimun norm solution of this system if and only if t^* lies in the range-space of B.*

Proof: See, for example, [8, Theorem 4.3.2].□

According with this theorem, an option for $\hat{x}^{(j)}$ is

$$\hat{x}^{(j)} = A_j (A_j^T A_j)^{-1} b^{(j)}. \tag{6}$$

On the other hand, a well-known result from linear algebra is that, if $\tilde{x}$ is a particular solution of (5), then $\tilde{x} + z$ is also a solution of (5) if and only if z lies in the null-space of A_j^T, that is, z is a solution of the corresponding homogeneous system

$$A_j^T x = \mathbf{0}.$$

Then, if $\tilde{x}$ is selected as the minimum norm solution and Z_j is an $n \times (n - \text{cardinal}(\mathcal{A}_j))$ matrix whose columns form a basis for the null-space of A_j^T, the solution of the above least-distance problem can be written in the form

$$\hat{x}^{(j)} = A_j (A_j^T A_j)^{-1} b^{(j)} + Z_j (Z_j^T Z_j)^{-1} Z_j^T x^{(j)}. \tag{7}$$

Finally, we note that an iteration point $\hat{x}^{(j)}$ has a multiplier vector $\mu^{(j)}$ associated and it is an optimal solution of the following subproblem

$$
\begin{array}{lll}
(ELP)_j & \underset{x\in R^n}{Minimize} & \ell(x) = c^T x \\
 & subject\ to & a_i^T x = b_i \text{ for all } i \in \mathcal{A}_j
\end{array}
$$

and, also, if $\mu^{(j)} \geq \mathbf{0}$, then $\hat{x}^{(j)}$ is an optimal solution of the subproblem $(ILP)_j$, linear program with the same objective function ℓ and the inequality constraints

$$a_i^T x \geq b_i \text{ for all } i \in \mathcal{A}_j$$

with the foreactive set $\mathcal{A}_j$ a set of active constraints at $\hat{x}^{(j)}$. That is why, when a first iteration point $\hat{x}^{(j)}$ is determined, the *sagitta* method basically pursuits to carry $\hat{x}^{(j)}$ towards the feasible region by modifying conveniently the foreactive set and trying that, when we have done it, the associated multiplier vector is non-negative, therefore, a feasible dual vector.

3. The Feasible-Point Sagitta Method

The first and basic scheme of the *sagitta* method [1] does not use an initial feasible point, starts with a foreactive set and a corresponding null-space descent direction and it repeats the constraint addition to the current foreactive set $\mathcal{A}_j$ until there is not a current null-space descent direction $d^{(j)}$. When an iteration point is determined (usually an exterior point to the feasible region), the method continues undertaking a loop of additions or exchanges to $\mathcal{A}_j$ of violated constraints at the exterior iteration point, until it concludes that the (ILP) has no solution or it finds a feasible solution that it is accustomed to be an optimal solution of the (ILP). The restarting of the global process is taken into account, with a contraint dropping from $\mathcal{A}_j$, but it seldom takes place.

The feasible-point approach presented below maintains the basic *sagitta* methodology, but it tries to take advantage of the "local" information upon the feasible region that a feasible point $x^{(j)}$ supplies; for example, selecting active constraints at this point to be added to the foreactive set. Also, as feasible-point or primal method, $x^{(j)}$ is moved approximating it to an optimal solution. The method is provided without a description of addition/exchange strategies for reasons of clarity and generality.

The Feasible-Point Sagitta Method

Let $x^{(1)}$ be a feasible solution; $\mathcal{A}_1 \leftarrow \varnothing$; $d^{(1)} \leftarrow -c$ $(\ c \neq \mathbf{0})$
$\mathcal{M} \leftarrow \{1,\ 2,\ \ldots,\ m\}$; $j \leftarrow 1$

While $\varnothing \neq \mathcal{C} \leftarrow \{\ i \in \mathcal{M} - \mathcal{A}_j \mid\ a_i^T d^{(j)} < 0\ \}$ **Do**
    Determine a step length α
    $x^{(j+1)} \leftarrow x^{(j)} + \alpha d^{(j)}$
    Select $\mathcal{P} \subseteq \mathcal{C}$ to be added to $\mathcal{A}_j$ with $\mathcal{A}_j \cup \mathcal{P}$ a linearly independent set.
    $\mathcal{A}_{j+1} \leftarrow \mathcal{A}_j \cup \mathcal{P}$.
    If there is not a null-space descent direction **Then**
        $j \leftarrow j+1$
        Determine a solution $\hat{x}^{(j)}$ of the equations $a_i^T x = b_i$ for all $i \in \mathcal{A}_j$.
        Determine the solution $\mu^{(j)}$ of $A_j \mu = c$.
        While $\varnothing \neq \mathcal{V} \leftarrow \{\ i \in \mathcal{M} - \mathcal{A}_j \mid\ a_i^T \hat{x}^{(j)} - b_i < 0\ \}$ **Do**
            If $\ell(\hat{x}^{(j)}) < \ell(x^{(j)})$ **Then**
                Determine a step length $\alpha \in [0,\ 1)$ for the direction $\hat{x}^{(j)} - x^{(j)}$
                $x^{(j+1)} \leftarrow x^{(j)} + \alpha(\hat{x}^{(j)} - x^{(j)})$
            Endif
            Select $p \in \mathcal{V}$ to be added to $\mathcal{A}_j$
            If $a_p \in$ Range–space of A_j **Then**
                Select $q \in \mathcal{A}_j$ to be exchanged for p
                $\mathcal{A}_{j+1} \leftarrow \mathcal{A}_j - \{q\} \cup \{p\}$
            Else
                $\mathcal{A}_{j+1} \leftarrow \mathcal{A}_j \cup \{p\}$.
            Endif
            Determine a solution $\hat{x}^{(j+1)}$ of the equations $a_i^T x = b_i,\ \forall i \in \mathcal{A}_{j+1}$
            Determine the solution $\mu^{(j+1)}$ of $A_{j+1}\mu = c$
            $j \leftarrow j+1$.
        Endwhile
        If $\ell(\hat{x}^{(j)}) \leq \ell(x^{(j)})$ **Then** $x^{(j+1)} \leftarrow \hat{x}^{(j)}$ **Endif**
        If $\mu^{(j)} \geq 0$ **Then Stop** *(Problem (ILP) has an optimal solution* $x^{(j)}$*)*
        Select $\mathcal{Q}$ to be dropped from $\mathcal{A}_j$
        $\mathcal{A}_{j+1} \leftarrow \mathcal{A}_j - \mathcal{Q}$
    Endif
    Determine a null-space descent direction $d^{(j+1)}$
    $j \leftarrow j+1$
Endwhile
Stop *(The objective function for the* (ILP) *is unbounded below)*

We note that the main explicit difference with respect to the first scheme is the movement of the feasible point $x^{(j)}$, but an implicit difference consists in the specific addition/exchange strategies for this feasible-point method. Also, the internal loop continues as a primal feasibility search loop since the multiplier vector is associated to the exterior iteration point, and, trivially, a check for linear programs with no feasible solution is absent.

The initial selection of $\mathcal{A}_1 = \varnothing$ and $d^{(1)} = -c$ could be changed by another foreactive set and a corresponding null-space descent direction.

The multiple addition of constraints (cardinal($\mathcal{P}$) > 1) will be convenient if we try to join the local and global viewpoint. In the external loop specifically, if the set $\mathcal{C}$ of the contrary constraints to the null-space descent direction $d^{(j)}$ is not empty, we adopt to select:

A) *Boundary or local viewpoint*: Or an active constraint at $x^{(j)}$ if the direction $d^{(j)}$ is not feasible and, hence, $\alpha = 0$, or else an active constraint at the new feasible point after a step in the direction $d^{(j)}$ with step length $\alpha > 0$.

B) *Global viewpoint*: A constraint in $\mathcal{C}$, if it does exist, that persists in being a contrary constraint to the updated null-space descent direction, after the constraint selected with the boundary criterion is added to the forective set, or none otherwise.

The "most contrary" thumb rule can be utilized.

The null-space descent direction $d^{(j+1)}$ can be taken as any solution of

$$\begin{bmatrix} A_{j+1}^T \\ c^T \end{bmatrix} d^{(j+1)} = \begin{bmatrix} \mathbf{0} \\ -1 \end{bmatrix}$$

or the *steepest-descent* null-space direction [4, p. 377–378]

$$d^{(j+1)} = -Z_{j+1}(Z_{j+1}^T Z_{j+1})^{-1} Z_{j+1}^T c. \tag{8}$$

The exterior iteration point $\hat{x}^{(j)}$ is a solution of the system (5). If this system is underdetermined, $\hat{x}^{(j)}$ can be computed through (6) or (7).

The determination of a new iteration point $\hat{x}^{(j+1)}$ in the internal loop can be carried out by a simple update

$$\hat{x}^{(j+1)} = \hat{x}^{(j)} + \frac{-r_p(\hat{x}^{(j)})}{a_p^T P_j a_p} P_j a_p \tag{9}$$

where $r_p(\hat{x}^{(j)}) = a_p^T \hat{x}^{(j)} - b_p$ and P_j is the projection operator

$$P_j = I - A_j(A_j^T A_j^T)^{-1} A_j^T, \tag{10}$$

even though $q \in \mathcal{A}_j$ is exchanged for p, but in this case we have to use the foreactive-set matrix A_j updated without the column corresponding to the qth constraint.

The computation of a multiplier vector $\mu^{(j)}$ that solves the compatible system (4) can be carried out without the least difficulty. In the internal loop, when a constraint addition takes place, the update formula of the multiplier vector is

$$\mu^{(j+1)} = \begin{bmatrix} \mu^{(j)} \\ 0 \end{bmatrix};$$

but, when $a_p = A_j^T \eta$ and a constraint exchange occurs, the update formula is

$$\mu^{(j+1)} = \begin{bmatrix} \bar{\mu}^{(j)} - \tau\bar{\eta} \\ \tau \end{bmatrix}$$

where $\bar{\mu}^{(j)}$ and $\bar{\eta}$ are the respective vectors $\mu^{(j)}$ and η without their kth element correspondig to the exchanged constraint q for p in $\mathcal{A}_j$, and $\tau = \mu_k^{(j)}/\eta_k$.

Generally, since the first determined $\hat{x}^{(j)}$ is accustomed to be an exterior point, the internal loop is crucial for the method convergence aimed in the sense that we have an optimal solution the first time that $\hat{x}^{(j)}$ arrives at the boundary of the feasible region. Santos [1] points out a definite parallelism, from an algorithmic viewpoint, of the internal or primal-feasibility search loop with dual methods for solving quadratic programs (see, for example, [9, 10]). And he proved the following convergence theorem for the *sagitta* method, although the *cycling* possibility cannot be ruled out if zero or cuasizero multipliers are numerous.

Theorem 3.1 *Assume that an optimal solution $\hat{x}^{(j)}$ of a subproblem $(ILP)_j$ has been computed. Then, henceforth, the objective function does not decrease in the primal-feasibility search loop if, when p is such that a_p lies in the range-space of A_j, a constraint $q \in \mathcal{A}_j$ is exchanged for p, where q is determined by*

$$\frac{\mu_k^{(j)}}{\eta_k} = \min \left\{ \frac{\mu_h^{(j)}}{\eta_h} \ \textit{for}\ \eta_h > 0,\ h \in \{1,\ 2,\ \ldots,\ \textit{cardinal}(\mathcal{A}_j)\} \right\}, k = h(q). \tag{11}$$

Furthermore, if such loop finishes obtaining a feasible solution, this one is an optimal solution of the (ILP).

In the internal loop and bearing in mind this result, Santos [1] suggests to select the pth constraint to be added to $\mathcal{A}_j$ as the most violated constraint at $\hat{x}^{(j)}$, with normalization in 2-norm, and to use (even when some $\mu_i^{(j)} < 0$) the criterion (11) to determine q to be exchanged for p. The behaviour of the primal-feasibility loop is good, usually convergent.

In this feasible-point method, the availability of a feasible point compels us to consider a "local" strategy to select p. We have used the boundary criterion facilitated for the external loop with $\hat{x}^{(j)} - x^{(j)}$ as direction if $\ell(\hat{x}^{(j)}) < \ell(x^{(j)})$, that is to say if it is a descent direction, or, in other case, the most violated constraint at $\hat{x}^{(j)}$.

We note that none of the aforementioned strategies is sufficient to ensure the method convergence to an optimal solution at the first arrival of $\hat{x}^{(j)}$ at the boundary of the feasible region, as it can be tested using the particular linear programs described by Goldfarb [11], whose feasible region is constructed combinatorially equivalent to the n-cube (see Example 4.3 in §4). Nevertheless, the research continues.

4. Examples

Example 4.1: *(Unbounded objective)*

$$\begin{array}{lrrrrcl}
\textit{Minimize} & -2x_1 & -3x_2 & +x_3 & +12x_4 & & \\
\textit{subject to} & x_1 & & & & \geq & 0 \\
& & x_2 & & & \geq & 0 \\
& & & x_3 & & \geq & 0 \\
& & & & x_4 & \geq & 0 \\
& 2x_1 & +9x_2 & -x_3 & -9x_4 & \geq & 0 \\
& -\frac{1}{3}x_1 & -x_2 & +\frac{1}{3}x_3 & +2x_4 & \geq & 0
\end{array}$$

Solution: This is a linear program constructed by Kuhn (see [4, p. 351]) to show that cycling can occurs in the simplex method. We initiate the feasible-point *sagitta* method with the feasible point $x^{(1)}$ as the origin, which is a degenerate vertex where all six constraints are active. Starting with the foreactive set $\mathcal{A}_1 = \varnothing$ and the descent direction $d^{(1)} = -c$, the indices of the contrary constraints to the direction $d^{(1)}$ are $\{3,\ 4,\ 6\}$, so

$$\frac{a_3^T d^{(1)}}{\|a_3\|_2} = -1, \qquad \frac{a_4^T d^{(1)}}{\|a_4\|_2} = -12, \qquad \frac{a_6^T d^{(1)}}{\|a_6\|_2} = \frac{-28}{\sqrt{47}/3},$$

and we select the constraint 6; since only constraint 4 persists in being contrary to the updated direction $[0,\ -1/3,\ 0,\ -1/6]^T$, we finally add $\mathcal{P} = \{6,\ 4\}$ to the foreactive set and it results $\mathcal{A}_2 = \{6,\ 4\}$. Then, a corresponding null-space descent direction is

$$d^{(2)} = \begin{bmatrix} 1 \\ -1/3 \\ 0 \\ 0 \end{bmatrix}.$$

The indices of the contrary constraints to $d^{(2)}$ are $\{2,\ 5\}$:

$$\frac{a_2^T d^{(2)}}{\|a_2\|_2} = -1/3, \qquad \frac{a_5^T d^{(2)}}{\|a_5\|_2} = \frac{-1}{\sqrt{167}}, \qquad a_i^T d^{(2)} \geq 0 \text{ for } i = \{1,\ 3\}$$

and we select constraint 2; since constraint 5 is not contrary to the updated direction

$$d^{(3)} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

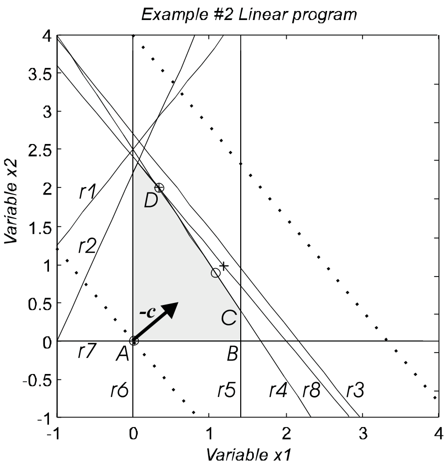

Figure 1: Example #2 Linear Program

we only add $\mathcal{P} = \{2\}$ to the foreactive set and it results $\mathcal{A}_3 = \{6,\ 4,\ 2\}$. This direction $d^{(3)}$ is such that $A^T d^{(3)} \geq \mathbf{0}$ and, therefore, in conformity with Theorem 1 the objective function is unbounded below for this linear program. Note that we can conclude this with no move of the initial feasible point and without the computation of an exterior iteration point.

Example 4.2: *(Non-unique solution)*

$$\begin{array}{llrrcr}
\textit{Minimize} & \ell(x) = & -(6/5)x_1 & -x_2 & & \\
\textit{subject to} & & (5/4)x_1 & -x_2 & \geq & -5/2 \\
 & & (11/5)x_1 & -x_2 & \geq & -11/5 \\
 & & -(3/2)x_1 & -(6/5)x_2 & \geq & -13/4 \\
 & & -(3/2)x_1 & -x_2 & \geq & -5/2 \\
 & & -x_1 & & \geq & -7/5 \\
 & & x_1 & & \geq & 0 \\
 & & & x_2 & \geq & 0 \\
 & & -(6/5)x_1 & -x_2 & \geq & -12/5
\end{array}$$

Solution: The application of the feasible-point *sagitta* method to this linear program is depicted in figure 1. We have marked with ri the line corresponding to the ith constraint, we have shadowed the feasible region and the two parallel dotted lines represent the contour lines $\ell(x) = 0$ and $\ell(x) = -4$. As an aside, note that $A \to B \to C \to D$ is the path followed by the simplex method. Starting with the feasible point $x^{(1)}$ as the origin, the foreactive set $\mathcal{A}_1 = \varnothing$ and the descent direction $d^{(1)} = -c$ (see figure 1), the indices of the contrary constraints to the direction $d^{(1)}$ are $\{3,\ 4,\ 5,\ 8\}$, so

$$\frac{a_3^T d^{(1)}}{\|a_3\|_2} = \frac{-3}{\sqrt{369}/10}, \quad \frac{a_4^T d^{(1)}}{\|a_4\|_2} = \frac{-14/5}{\sqrt{13}/2}, \quad \frac{a_5^T d^{(1)}}{\|a_5\|_2} = -6/5, \quad \frac{a_8^T d^{(1)}}{\|a_8\|_2} = \frac{-61/25}{\sqrt{61}/5},$$

and the min-ratio test gives $\alpha = 25/28$; we then move the feasible-point $x^{(1)}$ in the direction $d^{(1)}$ to obtain $x^{(2)} = [15/14,\ 25/28]^T$ (shown with a circle in figure 1), activating the constraint 4. Now, only constraints in $\{3,\ 8\}$ persists in being contrary to the updated direction $\tilde{d}^{(1)} = [-10/3,\ 5]^T$, so

$$\frac{a_3^T \tilde{d}^{(1)}}{\|a_3\|_2} = \frac{-1}{\sqrt{369}/10}, \qquad \frac{a_8^T \tilde{d}^{(1)}}{\|a_8\|_2} = \frac{-1}{\sqrt{61}/5},$$

and we select the most contrary constraint, in this case constraint 8. Suming up, we add $\mathcal{P} = \{4,\ 8\}$ to the foreactive set and it results $\mathcal{A}_2 = \{4,\ 8\}$. As cardinal$(\mathcal{A}_2) = n = 2$, there is not a corresponding null-space descent direction.

Solving the systems $A_2^T x = b^{(2)}$ and $A_2 \mu = c$ for

$$A_2 = \begin{bmatrix} -3/2 & -6/5 \\ -1 & -1 \end{bmatrix} \text{ and } b^{(2)} = \begin{bmatrix} -5/2 \\ -12/5 \end{bmatrix},$$

we obtain the iteration point $\hat{x}^{(2)}$ (see point D in figure 1, marked with a plus sign) and the multiplier vector $\mu^{(2)}$ associated with it, respectively; resulting

$$\hat{x}^{(2)} = \begin{bmatrix} 1/3 \\ 2 \end{bmatrix} \text{ and } \mu^{(2)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The residual vector associated with $\hat{x}^{(2)}$ is

$$r(\hat{x}^{(2)}) = A\hat{x}^{(2)} - b = \begin{bmatrix} 11/12 & 14/15 & 7/20 & 0 & 16/15 & 1/3 & 2 & 0 \end{bmatrix}^T,$$

so there is no violated constraint at $\hat{x}^{(2)}$ and the algorithm does not enter in the primal feasibility search loop. Since $\ell(\hat{x}^{(2)}) = -12/5 < -61/28 = \ell(x^{(2)})$, we can move the feasible point to obtain $x^{(3)} = \hat{x}^{(2)} = [1/3,\ 2]^T$ (see point D in figure 1, marked with a circle). The multiplier vector $\mu^{(2)}$ is non-negative too, so we have that $x^{(3)}$ is an optimal solution of the linear program and the minimum value of the objective function is $\ell(x^{(3)}) = -12/5$.

Finally, note that we could have adopted the criterion of it does not add to the foreactive set the activated constraint when we make a movement. In this example it implies not to add constraint 4 to the foreactive set and only add constraint 8, so we have no null-space descent direction. When solving the corresponding underdetermined system we obtain the exterior point $\hat{x}^{(2)} = [72/61,\ 60/61]^T$ (marked with a plus sign in figure 1); then, the algorithm cannot move the feasible point $x^{(2)}$ ($\alpha = 0$) and enters in the interior loop to add constraint 4 (the only violated constraint at this exterior point). Next, we determine the point $\hat{x}^{(3)} = [1/3,\ 2]^T$ (D in figure 1), which brings $x^{(2)}$ towards him and stops with optimal solution $\hat{x}^{(3)}$ since it is a feasible point which has non-negative multiplier vector.

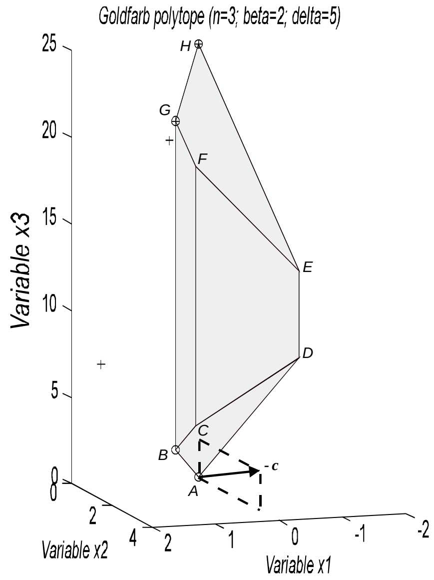

Figure 2: Goldfarb polytope ($n = 3$;$\beta = 2$;$\delta = 5$)

Example 4.3: *(Unique solution)*

$$\begin{array}{rlcl}
Minimize & \beta(2-\beta^2)x_2 + (1-\beta^2)x_3 & & \\
subject\ to & x_1 & \geq & 0 \\
 & -\beta x_1 + x_2 & \geq & 0 \\
 & x_1 - \beta x_2 + x_3 & \geq & 0 \\
 & -x_1 & \geq & -1 \\
 & -\beta x_1 - x_2 & \geq & -\delta \\
 & x_1 - \beta x_2 - x_3 & \geq & -\delta^2
\end{array}$$

Solution: This is a parametric linear program constructed by Goldfarb (see [11]) to show that the simplex method with the steepest-edge pivoting rule can be forced to visit all intervening vertices. The application of both the feasible-point *sagitta* method and the simplex method to a particular case of this linear program is depicted in figure 2, where we have labeled the vertices in lexicographical ordering to indicate the path followed by the simplex method. Note that we have shown four of the six faces of the feasible region:

Constraint	Vertices	Shown
1	$A-D-E-H$	No
2	$A-B-G-H$	No
3	$A-B-C-D$	Yes
4	$B-C-F-G$	Yes
5	$C-D-E-F$	Yes
6	$E-F-G-H$	Yes

We start with the initial feasible point $x^{(1)}$ (A in figure 2) as the origin (in which constraints 1, 2 and 3 are actives), the foreactive set $\mathcal{A}_1 = \varnothing$ and the descent direction $d^{(1)} = -c$ (see figure 2). The indices of the contrary constraints to the direction $d^{(1)}$ are $\{3,\ 5,\ 6\}$; using the strategy given in §3 we select $\mathcal{P} = \{3,\ 6\}$ to add to the foreactive set, so $\mathcal{A}_2 = \{3,\ 6\}$ and a null-space descent direction is

$$d^{(2)} = \begin{bmatrix} 1/2 \\ 1/4 \\ 0 \end{bmatrix}.$$

The indices of the contrary constraints to $d^{(2)}$ are $\{2,\ 4,\ 5\}$ and only constraint 2 is blocking, so $\mathcal{P} = \{2\}$ and $\mathcal{A}_3 = \{3,\ 6,\ 2\}$. As $cardinal(\mathcal{A}_3) = n = 3$, there is not a corresponding null-space descent direction. The first exterior iteration point (marked with a plus sign in figure 2) and its associated multiplier vector are:

$$\hat{x}^{(3)} = \begin{bmatrix} 25/6 \\ 25/3 \\ 25/2 \end{bmatrix} \text{ and } \mu^{(3)} = \begin{bmatrix} -1/6 \\ 17/6 \\ 4/3 \end{bmatrix}.$$

The indices of the violated constraints at $\hat{x}^{(3)}$ are $\{4,\ 5\}$ and the algorithm enters in the primal feasibility search loop. Since $\ell(\hat{x}^{(3)}) = -425/6 < 0 = \ell(x^{(3)})$, we can move the feasible point along the direction $\hat{x}^{(3)} - x^{(3)}$ during a step length $\alpha = 6/25$ to obtain

$$x^{(4)} = [1,\ 2,\ 3]^T$$

which is the B vertex in figure 2. Applying the criterion of Theorem 3 and the strategy given in §3 *for the internal loop* (the FP2 program in §6 uses this strategy), the index of the constraint to be exchanged for $p = 4$ (the activated constraint after we have done the movement) is $q = 3$, because $\eta = [1/6,\ 1/6,\ 2/3]^T$.

The new foreactive set is $\mathcal{A}_4 = \{6,\ 2,\ 4\}$, and the exterior iteration point and its associated multiplier vector would be:

$$\hat{x}^{(4)} = \begin{bmatrix} 1 \\ 2 \\ 22 \end{bmatrix} \text{ and } \mu^{(4)} = \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix},$$

so the exterior iteration point have arrived the feasible region on vertex G in figure 2. Since there is no violated constraints in $\hat{x}^{(4)}$ the algorithm exits the interior loop and forces the feasible point to meet $\hat{x}^{(4)}$:

$$x^{(5)} = \hat{x}^{(4)}$$

but the multiplier vector associated with $\hat{x}^{(4)}$ is not non-negative. Then we have to procced to the restarting procedure, dropping constraint $q = 4$ (most negative multiplier) and recovering a null-space descent direction

$$d^{(5)} = \begin{bmatrix} -1 \\ -2 \\ 3 \end{bmatrix},$$

that corresponds to $\mathcal{A}_5 = \{6,\ 2\}$. The only contrary constraint to $d^{(5)}$ is constraint 1 and there is no blocking constraint, so we can move the feasible point $x^{(5)}$ to the H point

$$x^{(6)} = [0,\ 0,\ 25]^T$$

and we add that constraint to the foreactive set, so $\mathcal{A}_6 = \{6,\ 2,\ 1\}$; then we do without descent direction and a newly exterior iteration point can be determined

$$\hat{x}^{(6)} = \begin{bmatrix} 0 \\ 0 \\ 25 \end{bmatrix} \text{ and } \mu^{(6)} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix},$$

which coincides again with $x^{(6)}$. Now the algorithm can stop with non-negative multiplier vector for the newly exterior iteration point $\hat{x}^{(6)}$.

Finally, note that we could have not adopted *for the internal loop* the strategy given in §3 and keep on considering p as the index of the most violated constraint at $\hat{x}^{(3)}$, instead of the index of the activated constraint after we have done the movement. In this case (the FP1 program in §6 uses this strategy) it would be $p = 5$ (the most violated constraint at $\hat{x}^{(3)}$) instead of $p = 4$; the index of the constraint to be exchanged for p is $q = 3$ too, because $\eta = [2/3,\ 2/3,\ 5/3]^T$. Now the new foreactive set would be $\mathcal{A}_{3bis} = \{6,\ 2,\ 5\}$, and the exterior iteration point $\hat{x}^{(3bis)}$ (close to G and marked with a plus sign in figure 2) and its associated multiplier vector would be:

$$\hat{x}^{(3bis)} = \begin{bmatrix} 5/4 \\ 5/2 \\ 85/4 \end{bmatrix} \text{ and } \mu^{(3bis)} = \begin{bmatrix} 3 \\ 7/4 \\ -1/4 \end{bmatrix}.$$

Constraint 4 would be the unique violated constraint at $\hat{x}^{(3bis)}$ and the algorithm would keep into the primal feasibility search loop. Although $\ell(\hat{x}^{(3bis)}) = -295/4 < -17 = \ell(x^{(3bis)})$ (with $x^{(3bis)} = [1,\ 2,\ 3]^T$), we would not be able to move the feasible point along the direction $\hat{x}^{(3bis)} - x^{(3bis)}$ since $\alpha = 0$ and then $x^{(4)} = x^{(3bis)}$. Once again, the application of the criterion of Theorem 3 and the most violated constraint thumb rule would supply that $q = 5$ would have to be exchanged for $p = 4$ (because $\eta = [0,\ 1/4,\ 1/4]^T$), and the new foreactive set would be $\mathcal{A}_4 = \{6,\ 2,\ 4\}$. Suming up, we would obtain the same foreactive set with the iteration point in the same place, but having determined one additional exterior iteration point and having done one additional iteration.

5. A Range-Space Implementation

We have carried out a first computational implementation of the *sagitta* method as a range-space method. The initial election of $\mathcal{A}_j = \varnothing$ justifies this option. The slack or artificial variables are unnecessary.

We make use of the QR factorization of the foreactive-set matrix A_j,

$$A_j = \begin{bmatrix} Y_j & Z_j \end{bmatrix} \begin{bmatrix} R_j \\ \mathrm{O} \end{bmatrix} = Y_j R_j,$$

storing and updating only the matrices Y_j and R_j. The column vectors of Y_j form an orthonormal basis for the range-space of A_j.

When the pth constraint is added to the foreactive set, $Y_{j+1} = \begin{bmatrix} Y_j & y \end{bmatrix}$ with

$$y = \frac{P_j a_p}{\sqrt{a_p^T P_j a_p}}$$

where $P_j = I_n - Y_j Y_j^T$ and, then, the equality

$$\begin{bmatrix} A_j & a_p \end{bmatrix} = \begin{bmatrix} Y_j & y \end{bmatrix} \begin{bmatrix} R_j & Y_j^T a_p \\ \mathbf{0}^T & \sigma \end{bmatrix}$$

holds with $\sigma = \sqrt{a_p^T P_j a_p}$. A numerically stable computation for σ is advised when the column a_p to be added is nearly linearly dependent of A_j (see [12, 13]). Also, update formulae are immediate for Y_j and R_j in the cases of exchange/drop of constraints.

Then, since $Z_j^T Z_j = I$ and $Z_j Z_j^T = I_n - Y_j Y_j^T$, the steepest descent null-space direction is

$$d^{(j)} = -Z_j (Z_j^T Z_j)^{-1} Z_j^T c = -Z_j Z_j^T c = -(I_n - Y_j Y_j^T) c$$

and the update formula is

$$d^{(j+1)} = d^{(j)} + (y^T c) y.$$

Furthermore, the first time that we compute an exterior point $\hat{x}^{(j)}$ we use

$$\hat{x}^{(j)} = Y_j R_j^{-T} b^{(j)}$$

and its associated multiplier vector $\mu^{(j)}$ is obtained by solving the triangular system

$$R_j \mu^{(j)} = Y_j^T c.$$

where their updating formulae in the primal-feasibility search loop were given in §3.

The Modified Gram-Schmidt (MGS) method with reorthogonalization or the Householder method can be used for a robust and numerically stable implementation [14].

6. Computational Results

Three MATLAB programs implementing the *sagitta* method and two (FP1 and FP2) feasible-point *sagitta* versions has been prepared and compared against. The FP1 program uses the local information only in the external loop of the algorithm, whereas the FP2 program uses it in the two loops. A personal computer was employed in this investigation.

No attempt has been made to do an extensive comparison of methods. Our objective has been to test only with some problems if a feasible-point approach makes possible an interesting reduction of the iteration number of the original *sagitta* method.

Table 1 gives the results obtained by solving randomly generated test problems for $n = 100$ and $m = 100,\ 200,\ 400,\ 800$. A block of 50 problems was solved for each couple $(n,\ m)$. The elements of A were generated as numbers at random, uniformly distributed in the range $[-1.,\ 1.]$. Also, the elements of an optimal solution x^* are random numbers uniformly distributed between $-5.$ and 5. and the elements of a generalized multiplier vector μ^* are random numbers uniformly distributed between 0. and 120., with zero multipliers corresponding to non-active constraints. Finally, the vectors c and b are computed according to the conditions of optimality and, for non-active constraints, randomly generated residues in the range $[0.,\ 1.]$. The initial feasible-point has been the nearest to the origin for all these problems.

n=100	*Sagitta Iterations*			*FP1 Sagitta Iterations*			*FP2 Sagitta Iterations*		
m	*mean*	*max*	*min*	*mean*	*max*	*min*	*mean*	*max*	*min*
100	**100.0**	100	100	**50.6**	52	50	**50.6**	52	50
200	**253.1**	325	210	**185.4**	275	113	**304.9**	458	175
400	**403.0**	471	327	**288.3**	379	220	**418.3**	632	249
800	**528.1**	611	417	**375.6**	462	294	**492.9**	644	343

Table 1: Average results for randomly generated linear programs

The results points out that, for this kind of problems, the FP1 program takes advantage over the *sagitta* method, but does not so the FP2 program. The FP1 program advantage is clearly interesting –more than $n/2$ iterations less of average– if m is greater. None of these problems had to drop a constraint from the foreactive set and to restart the external loop.

Table 2 gives the average results for 50 Kuhn-Quandt problems [15, p. 117]:

$$(KQ) \qquad \underset{x \in R^n}{Maximize} \qquad \ell(x) = \mathbf{1}^T x$$

$$\text{subject to} \qquad Nx \leq 10^4\mathbf{1}$$
$$x \geq \mathbf{0}$$

where $\mathbf{1}$ is a vector of all unit elements and N is a dense $n \times n$ matrix with integer elements chosen at random in the range 1 to 1000. The initial feasible-point has been the zero vector for all this problems.

	Sagitta Iterations			*FP1 Sagitta Iterations*			*FP2 Sagitta Iterations*		
n	*mean*	*max*	*min*	*mean*	*max*	*min*	*mean*	*max*	*min*
50	**132.7**	173	101	**109.3**	159	55	**99.7**	154	48
100	**409.0**	503	298	**324.0**	501	191	**279.3**	469	136
200	**1539.8**	2013	1255	**1314.8**	2012	776	**1063.7**	1811	433

Table 2: Average results for Kuhn-Quandt problems

The results points out that, for Kuhn-Quandt problems, both FP1 and FP2 programs take advantage over the *sagitta* method, and the FP2 program is slightly better. Nevertheless, for 5 over the 50 problems, the FP2 program restarts the external loop, after it drops one constraint from the foreactive set. We have seen that the objective function value at the first exterior iteration point is close to the objective optimal value, and this could be a reason for the good results of the FP2 program.

Finally, computational results are added for two problems with hard or pathological characteristics.

Table 3 gives results for parametric linear programs whose feasible region is a polytope combinatorially equivalent to the n-cube, and used by Goldfarb [11] on the complexity analysis of the simplex method:

$$\begin{array}{llrll} (PLP) & \underset{x \in R^n}{Maximize} & \ell(x) = c_{n-1} & x_{n-1} + & c_n \quad x_n \\ & \text{subject to} & 0 & \leq x_1 & \leq 1 \\ & & \beta x_1 & \leq x_2 & \leq \delta - \beta x_1 \\ & & \beta x_j - x_{j-1} & \leq x_{j+1} & \leq \delta^j - \beta x_j + x_{j-1} \end{array}$$

where $j = 2, \ldots, n-1$, $\beta \geq 2$ and $\delta > 2\beta$. The cost vector is selected in such a way that the optimal solution of the problem is the upper vertex of the polytope with all its elements, except the last, zeroed (see Example 3 in §4). The *simplex* method path proceeds through all the 2^n vertices if the initial vertex is the zero vector (using the *steepest-edge pivoting rule*), but the *Bland's least-index rule* makes possible to reduce the number of vertices visited.

Problem characteristics					*Iterations*			
n	β	δ	c_{n-1}	c_n	*Simplex*	*Sagitta*	*FP1 Sagitta*	*FP2 Sagitta*
6	2	9	7	6	25	31	13	36
6	3	9	377	144	25	33	13	36
6	4	9	2911	780	25	33	13	36
8	2	10	9	8	67	42	18	134
8	3	10	2584	987	67	54	17	134
10	2	8	11	10	177	52	27	520
10	2	10	11	10	177	52	27	520
12	2	8	13	12	465	61	39	2058
12	2	10	13	12	465	62	39	2058

Table 3: Results for Parametric Linear Programs

The results points out that these parametric linear programs are pathological for the FP2 program; however, the *sagitta* and FP1 programs take clear advantage. We have noted that numerical unstability turns up frequently when solving these problems.

Table 4 gives results for the hard `israel` test problem from *Netlib*. This problem has $n = 142$ nonnegative variables and $m = 174$ inequality constraints. The initial feasible point $x^{(1)}$ was the vector with objective value $\ell(x^{(1)}) = -81900$ and the following nonzero elements:

$$x^{(1)}_{[26,\ 37,\ 41,\ 42,\ 43,\ 44,\ 45,\ 46,\ 48]} = [100,\ 220,\ 300,\ 1100,\ 100,\ 200,\ 100,\ 400,\ 200].$$

Method	*Iterations* *Total*	*First Ext. Point*	*Optimal Objective Value*
Sagitta	310	137	-896644.821863070
FP1 Sagitta	210	61	-896644.821863064
FP2 Sagitta	385	62	-896644.821863693

Table 4: Results for the `israel` problem

The results are highly interesting if we compare against the total iterations facilitated by Bixby [16] for the simplex method using different Initial Bases —464 (Artificial), 172 (Feasible Slack), 204 (Slack) and 204 (CPLEX)—, because the *sagitta* methods do not use artificial nor slacks variables, and *Total Iterations* in Table 4 is the counter of the changes in the foreactive set. So, for example, the first *sagitta* computes only 173 iteration points.

7. Summary

We have considered a feasible-point approach for the *sagitta* method [1], a new non-simplex active-set method for solving linear programs in inequality form. The basic characteristics of the *sagitta* method are: the use of a global viewpoint of the problem to determine a foreactive set of constraints, starting without an iteration point; the constraint addition to the foreactive set to attempt initially to determine if the linear program is unbounded below, and the special primal-feasibility search loop (when an iteration point, usually exterior to the feasible region, is determined) to converge to an optimal solution the first time that the exterior iteration point arrives at the boundary.

The feasible-point *sagitta* method presented in this paper adds active constraints at the feasible point to the foreactive set, trying to take advantage of the "local" information upon the feasible region obtained by the feasible point. Convenient strategies are adopted pursuiting the method convergence.

The computational results obtained by a first range-space implementation are highly encouraging, but not concluding; however they clearly show that appropriate use of the "local" information allows us to solve different linear programs with an important reduction of the iteration number, perhaps using a suitable strategy for each different kind of linear programs.

References

1. Santos, A., *The* Sagitta *Method for Solving Linear Programs*, Submitted for publication, 1998.

2. Goldfarb, D., and Todd, M. J., *Linear Programming*, Optimization, Edited by G. L. Nemhauser et al., pp. 73-170, 1989.

3. Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D., *Linear Programming and Network Flows*, Second Edition, John Wiley & Sons, Nueva York, 1990.

4. Gill, P. E., Murray, W., and Wright, M. H., *Numerical Linear Algebra and Optimization, Volumen 1.* Addison-Wesley Publishing, Redwood City, California, 1991.

5. Saigal, R., *Linear Programming. A Modern Integrated Analysis*, Kluwer Academic Pub, 1995.

6. Dantzig, G. B., and Thapa, M. N., *Linear Programming. 1: Introduction*, Springer-Verlag, 1997.

7. Gill, P. E., and Murray, W., *A numerically stable form of the simplex algorithm*, Linear Algebra and its Applications, Vol. 7, pp. 99-138, 1973.

8. Peressini, A. L., Sullivan, F. E., and Uhl, J. J., *The Mathematics of Nonlinear Programming*, Springer-Verlag, 1988.

9. Goldfarb, D., and Idnani, A., *A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs*, Mathematical Programming, Vol. 22, pp. 1-33, 1983.

10. Caballero, R., and Santos, A., *A new dual method for solving strictly convex quadratic programming*, Submitted for publication, 1998.

11. Goldfarb, D., *On the complexity of the simplex method*, Advances in Optimization and Numerical Analysis, Edited by S. Gómez and J. P. Hennart, Kluwer Academic Pubs, pp. 25-38, 1994.

12. Gill, P. E., Golub, G. H., Murray, W., and Saunders, M. A., *Methods for Modifying Matrix Factorizations*, Mathematics of Computation, Vol. 28, pp. 505-535, 1974.

13. Bjrck, A., *A Direct Method for Sparse Least Squares Problems with Lower and Upper Bounds*, Numerische Mathematik, Vol. 54, pp. 19-32, 1988.

14. Golub, G. H. and Van Loan, C. L., *Matrix Computations*, North Oxford Academic Publishing, Oxford, 1996.

15. Megiddo, N., *Progress in Mathematical Programming*, Springer-Verlag, 1989.

16. Bixby, R. E., *Implementing the Simplex Method: The Initial Basis*, ORSA Journal on Computing, Vol. 4, pp. 267-284, 1992.