

On Active-Set LP Algorithms Allowing Basis Deficiency

Pablo Guerrero-García[1(✉)] and Eligius M. T. Hendrix[2]

[1] Applied Mathematics, Universidad de Málaga, 29071 Málaga, Spain
pablito@ctima.uma.es
[2] Computer Architecture, Universidad de Málaga, 29071 Málaga, Spain
eligius@uma.es

Abstract. An interesting phenomenon in linear programming (LP) is how to deal with solutions in which the number of nonzero variables is less than the number of rows of the matrix in standard form. An interesting approach is that of basis-deficiency-allowing (BDA) simplex variations, which work with a subset of independent columns of the coefficient matrix in standard form, where the basis is not necessarily represented by a square matrix. By considering a different view on the usual dual-primal non-symmetric interaction, our aim is to show a relation between BDA and the non-simplex active-set methods. The whole is illustrated by several numerical examples. The ideas may aid the understanding of nowadays BDA approaches to sparse implementation and dealing with the Phase I.

Keywords: Linear programming · Phase I · Basis-deficiency-allowing simplex variations · Non-simplex active-set method · Farkas lemma · Sparse matrix

1 Introduction

It is intriguing that since the introduction of the simplex method for linear programming (LP) in 1947, there has been interest in viewing algorithms from a nonlinear optimisation point of view. Most well known and developed are the ideas of interior point methods since the first barrier methods of Dikin in 1967. A generalization of the well-known dual simplex method can be found in [9,35] relaxing operations towards non-simplex steps. One can find more recent considerations in this line in [25,31] with the incorporation of steepest-edge pivoting rules, see also [11,24]. In [9] Gill and Murray presented a non-simplex active-set algorithm for solving a problem (P) starting with a feasible point x_0. Descriptions and analyses can also be found in [6,10,24]. Our interest is to investigate the possibility to develop a non-simplex active-set (NSA) method for a dual linear program in standard form (D). A challenge in practical cases is to have a high degeneracy, which for a simplex-like method provides basis-deficiency, i.e. the dual variable set has less than n variables. Therefore, we aim

This paper has been supported by The Spanish Ministry (RTI2018-095993-B-I00) in part financed by the European Regional Development Fund (ERDF).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13378, pp. 174–187, 2022.
https://doi.org/10.1007/978-3-031-10562-3_13

for the development of a method which is also called a basis-deficiency-allowing (BDA) simplex variation, see [25,31,32].

Our first viewpoint is that of the method presented by Gill and Murray which uses a non-simplex active-set algorithm for solving an LP problem starting with a feasible point x_0. Our question is whether changing the primal vision by the dual vision may help us to be able to handle rank-deficiency instances.

To investigate the question, we introduce the notation in Sect. 2 and the main algorithm in Sect. 3. We sketch the convergence of the algorithm in Sect. 4 and discuss several ways to deal with Phase I in Sect. 5. Section 6 discusses implementation issues and numerical experiments. Section 7 summarises our findings.

2 Notation of the Primal-Dual Vision

Consider the usual non-symmetric primal-dual relation in linear programming. Notice that we deviate from the usual notation here. This means, we exchanged b and c, x and y, n and m, and (P) and (D) in the notation of e.g. [24]. Now consider

$$\begin{array}{lll} (P) & \min & \ell(x) := c^T x \ , \ x \in \mathbb{R}^n \\ & \text{s.t.} & A^T x \geq b \end{array} \qquad \begin{array}{lll} (D) & \max & \mathcal{L}(y) := b^T y \ , \ y \in \mathbb{R}^m \\ & \text{s.t.} & Ay = c \quad , \ y \geq 0 \end{array} \tag{1}$$

where $A \in \mathbb{R}^{n \times m}$ with $m \geq n$ and $\text{rank}(A) = n$. Let $\mathcal{F}$ and $\mathcal{G}$ denote the feasible region of (P) and (D), respectively.

The idea is the following. We start with a dual feasible point y_0. From there, we separate the index set $[1:m] := \{1, 2, \ldots, m\}$ into an ordered basic set $\mathcal{B}_k$ with $m_k := |\mathcal{B}_k|$ elements, with $m_k \leq n$ and its complement representing the columns in $\mathcal{N}_k := [1:m] \setminus \mathcal{B}_k$, which can have more than $m - n$ zero elements. Let $A_k \in \mathbb{R}^{n \times m_k}$ and $N_k \in \mathbb{R}^{n \times (m - m_k)}$ be sub-matrices of A formed by the columns corresponding to $\mathcal{B}_k$ and $\mathcal{N}_k$, respectively. Notice that $\text{rank}(A_k) = m_k$. We will use the notation $\mathcal{R}(A)$ to represent the range of the columns in A and $\text{null}(A)$ denotes its null space. Column j of A is denoted by a_j. Similarly, a_i^T is row i of matrix A^T. Moreover, $b_{\mathcal{B}_k}$ represents an m_k vector with the elements of $\mathcal{B}_k$ of b and $b_{\mathcal{N}_k}$ has the non-basic elements. The index k will be left out if it is clear from the context.

3 Algorithm

The algorithm works like the simplex method in determining in each iteration a non-basic variable that enters the basis and a basic variable(s) leaving the basis. The big difference is that it can handle basis-deficiency. This means that $m_k \leq n$ and m_k may reduce and grow. A pseudo-code is given in Algorithm 1.

- Notice in line 4, if we have a complementary pair of primal-dual feasible points, then x_k is optimal for (P).
- Line 15 determines step size and basis leaving variable(s), unless in line 12 we found that we have an unbounded dual objective function.

Algorithm 1. Pseudocode active-set algorithm

Set $k \leftarrow 0$, y_0 feasible point of (D) with corresponding $\mathcal{B}_0$ and A_0
$x_k \leftarrow$ solves $A_k^T x = b_{\mathcal{B}_k}$ with residuals r_k following from $r_{\mathcal{N}_k} = N_k^T x_k - b_{\mathcal{N}_k}$
if $r_k \geq 0$ **then**
    return x_k is optimal
select $p \in \mathcal{N}_k$ with $r_{kp} < 0$
if $a_p \notin \mathcal{R}(A_k)$ **then**
    $\mathcal{B}_{k+1} \leftarrow \mathcal{B}_k \cup \{p\}$, $\mathcal{N}_{k+1} \leftarrow [1:m] \setminus \mathcal{B}_{k+1}$, set A_{k+1}
    $y_{k+1} \leftarrow$ follows from $A_{k+1} y_{\mathcal{B}_{k+1}} = c$, $k \leftarrow k+1$, goto step 2
else
    solve $A_k \delta = a_p$
    if $\delta \leq 0$ **then**
        return Unbounded $\mathcal{L}$
    else
        Direction d_k is initially unit vector e_p and elements of $\mathcal{B}_k$ follow from $-\delta$
        Step-size $\tau \leftarrow \min_{q \in \mathcal{B}_k} \{\frac{y_{kq}}{-d_{kq}} | d_{kq} < 0\}$, $Q \leftarrow \operatorname{argmin}_{q \in \mathcal{B}_k} \{\frac{y_{kq}}{-d_{kq}} | d_{kq} < 0\}$
        $y_{k+1} \leftarrow y_k + \tau d_k, \mathcal{B}_{k+1} \leftarrow \mathcal{B}_k \setminus Q \cup \{p\}, \mathcal{N}_{k+1} \leftarrow \mathcal{N}_k \cup Q$
        Set A_{k+1}, $k \leftarrow k+1$, goto step 2

The algorithm does not require a primal feasible starting point. In that sense, it is an exterior method as sketched by the bottom-left part of Fig. 1. It has been studied by several researchers.

In earlier studies we have shown [40] that the algorithm is equivalent to the primal BDA simplex variation given by Pan in [25]. However, we put emphasis on a description which is easy to understand due to a geometrical interpretation and its independence of implementation details. The algorithm also fits in the primal-feasibility search loop of the sagitta method described by Santos-Palomo [37], which is related to a loop used in dual active-set methods for quadratic programming. Moreover, Li [20] extended the algorithm to deal with upper and lower bounds on y.

Practically, one obtains the primal-feasibility Phase I of Dax 1978 [5] idea for Rosen 1960 [35] when one replaces the min-ratio test by a most-obtuse-angle row rule to determine the leaving variable and removing an objective function consideration. In this context, we focused earlier [16] on the classical example of Powell [33] which illustrates the cycling behaviour of a most-obtuse-angle simplex pivoting rule described in [21,31].

The algorithm ends with an optimal solution for (P) and (D). However, the solution does not necessarily correspond to a square basis. Notice that the algorithm maintains dual feasibility and complementary slackness.

Example 1. Consider the following instance of an LP problem (adapted from [30]).

$$A = \begin{pmatrix} -1 & 1 & 1 & -1 & 1 & 0 & 1 & -2 \\ 2 & -1 & 1 & 1 & 0 & 1 & -1 & 1 \\ 1 & -2 & 0 & -1 & 0 & 0 & 0 & -3 \\ -1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 2 & -1 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix}, c = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, b^T = (2, -3, 4, -1, 0, 0, -2, 3)$$

We take as starting point $y_0 = (0, 0, 0, 0, 2, 1, 0, 0)^T$ with $\mathcal{L}_0 = 0$ and $\mathcal{B}_0 = \{5, 6\}$. Using

$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, we have $x_0 = 0$ and $r_0 = (-2, 3, -4, 1, 0, 0, 2, -3)^T$. Selecting y_3 to enter the deficient basis, the step size in step 15 will be $\tau = 1$, having $Q = \{6\}$ leaving the basis. This means that $\mathcal{B}_1 = \{3, 5\}$ corresponding to

$A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, y_1 = (0, 0, 1, 0, 1, 0, 0, 0)^T, \mathcal{L}_1 = 4$. Solving $A_1^T x = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$, provides primal solution $x_1 = (0, 4, 0, 0, 0)^T$ with residuals $r_1 = (6, -1, 0, 5, 0, 4, -2, 1)^T$. This means the solution is not optimal. We select y_7 for entering the basis. We have as direction of improvement $d_1 = (0, 0, 1, 0, -2, 0, 1, 0)^T$ providing a step size of $\tau = \frac{1}{2}$, $Q = \{5\}$ leaves the basis and $y_2 = y_1 + \frac{1}{2}d_1 = (0, 0, \frac{3}{2}, 0, 0, 0, \frac{1}{2}, 0)^T$. Having

$A_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, with $\mathcal{B}_2 = \{3, 7\}$ as deficient basis and objective function value $\mathcal{L}_2 = 5$. Now $x_2 = (1, 3, 0, 0, 0)^T$ is the new primal point with residuals $r_2 = (3, 1, 0, 3, 1, 3, 0, -2)^T$. This means we are interested to have $p = 8$ in the basis. However, from matrix A it is easy to see that $a_8 \notin \mathcal{R}(A_2)$. This means that we extend the basis to capture also a_8. The new matrix for $\mathcal{B}_3 = \{3, 7, 8\}$ becomes

$A_3 = \begin{pmatrix} 1 & 1 & -2 \\ 1 & -1 & 1 \\ 0 & 0 & -3 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix}$. It is interesting that one obtains a degenerate solution in the extension case, as variable y_8 cannot take a positive value by adapting

the other positive values with a direction δ, given that $a_8 \notin \mathcal{R}(A_2)$. So in this example, we have that solution $y_3 = y_2$. As the solution is degenerate, the primal has alternative solutions of $A_3^T x = b_{\mathcal{B}_3}$, as there are two rows less. Some possible solutions x_3 to this under-determined system of linear equations are the so-called basic least-squares solution $(1, 3, -\frac{2}{3}, 0, 0)^T$ and the so-called minimum-norm least-squares solution $(1, 3, -\frac{6}{11}, \frac{2}{11}, -\frac{2}{11})^T$, where the corresponding residual vectors $r_3 = A^T x_3 - b$ are $(\frac{7}{3}, \frac{7}{3}, 0, \frac{11}{3}, 1, 3, 0, 0)^T$ and $(\frac{21}{11}, \frac{27}{11}, 0, \frac{39}{11}, 1, 3, 0, 0)^T$. This means, both the dual point and the residuals of the primal point are completely nonnegative, indicating that y_3 and x_3 are optimal solutions of the LP problem with $\ell_3 = \mathcal{L}_3 = 5$.

The numerical example illustrates how one can work with a deficient basis. The algorithm not only allows extending the basis, but it can also shrink, if $|Q| > 1$.

Example 2. To illustrate a potential reduction of $|B_k|$, consider the same values for A and b of Example 1 and we consider now $c^T = (1, 1, 0, 0, 0)$. We take as starting point $y_0 = (0, 0, 0, 0, 1, 1, 0, 0)^T$ with the same $\mathcal{L}_0$, $\mathcal{B}_0$, A_0, x_0 and r_0. After selecting y_3 to enter the deficient basis, this example leads to tie in the min-ratio test since $d_0 = (0, 0, 1, 0, -1, -1, 0, 0)^T$. Hence $\tau = 1$, but there a multiple deletion occurs with $Q = \{5, 6\}$. This means that $\mathcal{B}_1 = \{3\}$ corresponding to $A_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, y_1 = (0, 0, 1, 0, 0, 0, 0, 0)^T, \mathcal{L}_1 = 4$. Solving $A_1^T x = 4$ provides a set of primal points, which can be described by $x_1 = (a, b, c, d, e)^T$ where $a + b = 4$. For example, using $c = d = e = 0$ we obtain:

- $a = 4$ and $b = 0$ with residuals $r_1 = (-6, 7, 0, -3, 4, 0, 6, -11)^T$
- $a = 0$ and $b = 4$ with residuals $r_1 = (6, -1, 0, 5, 0, 4, -2, 1)^T$
- $a = 2$ and $b = 2$ with residuals $r_1 = (0, 3, 0, 1, 2, 2, 2, -5)^T$

This means that none of them is optimal in spite of having $\ell_1 = 4 = \mathcal{L}_1$, but note that the number of candidates to enter the deficient basis differs in the next iteration. For the illustration we choose the third possibility corresponding to $a = b = 2$, which is the minimum 2-norm solution of $A_1^T x = 4$ (with $\|x_1\|_2 = \sqrt{8} < 4$). In that case, only y_8 is a basis entering candidate variable, $a_8 \notin \mathcal{R}(A_1)$. Consequently, $\mathcal{B}_2 = \{3, 8\}$ corresponds to $A_2 = \begin{pmatrix} 1 & -2 \\ 1 & 1 \\ 0 & -3 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}, y_2 = y_1, \mathcal{L}_2 = \mathcal{L}_1$. Solving $A_2^T x = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$ provides a set of primal points, for example:

- $x_2 = (0, 4, 1/3, 0, 0)^T$, $r_2 = (19/3, -5/3, 0, 14/3, 0, 4, -2, 0)^T$;
- $x_2 = (47, 77, -30, 10, -10)/31^T$, $r_2 = (-15, 143, 0, 91, 47, 77, 32, 0)/31^T$.

This means that none of them is optimal in spite of having $\ell_2 = 4 = \mathcal{L}_2$. However, there are (again) multiple candidates to enter the deficient basis in the next iteration. Consider the second possibility i.e., the minimum 2-norm solution of $A_2^T x = (4,3)^T$. Then only y_1 can be selected to enter the basis, $a_1 \notin \mathcal{R}(A_2)$ and then $\mathcal{B}_3 = \{1,3,8\}$ corresponds to $A_3 = \begin{pmatrix} -1 & 1 & -2 \\ 2 & 1 & 1 \\ 1 & 0 & -3 \\ -1 & 0 & 1 \\ 2 & 0 & -1 \end{pmatrix}, y_3 = y_2, \mathcal{L}_3 = \mathcal{L}_2$. The solution of $A_3^T x = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix}$ provides a set of primal points among which:

- $x_3 = (19, 29, -15, 0, 0)/12^T$, $r_3 = (0, 56, 0, 37, 19, 29, 14, 0)/12^T$;
- $x_3 = (154, 274, -100, 30, -25)/107^T$, $r_3 = (0, 456, 0, 332, 154, 274, 94, 0)/107^T$.

This means that both are optimal with $\ell_3 = 4 = \mathcal{L}_3$. Notice that we have finished with a deficient basis and that there was no improvement in the objective functions for two iterations. This illustrates how the selection of the technique to solve the underdetermined system of equations giving the current primal point affects the overall path.

4 Proof of Convergence

Without going into a formal description, it can be shown that the algorithm converges under some circumstances. We will detail some observations.

One of the observations is that degeneracy of the dual when a basic variable is exchanged for a non-basic one does not occur. This is more or less included in the idea that A_k is not full rank leaving out all zero valued variables in y, i.e. $y_j > 0$ for $j \in \mathcal{B}_k$. In this way, convergence of $\mathcal{L}_k := b^T y_k$ is forced. Notice that degeneracy will occur when extending the basis with a new variable, as the old solution will be taken.

During the iterations, we have $y_{k+1} = y_k + \tau d_k$ with $\tau > 0$. In order to have $\mathcal{L}_{k+1} > \mathcal{L}_k$, so $b^T y_{k+1} > b^T y_k$ it must hold that $b^T d_k > 0$. To prove this, we can consider that $A d_k = 0$. As $A_k \delta = a_p$, by the construction of the search direction vector defined in line 14, we have that $A d_k = -A_k \delta + a_p = 0$. Now consider the residuals where for the basic variables we have $r_{kj} = 0, j \in \mathcal{B}_k$ and specifically $r_p < 0$. From $b = A^T x_k - r_k$, it follows that

$$b^T d_k = (A^T x_k - r_k)^T d_k = x_k^T A d_k - r_k^T d_k = 0 - r_p > 0 \tag{2}$$

We are not certain of an improvement, when adding a_p in case $a_p \notin \mathcal{R}(A_k)$. In that case y_k with $y_{kp} = 0$ is still a feasible solution, such that $\mathcal{L}_{k+1} \geq \mathcal{L}_k$, but no guarantee is given on improvement. This step is of course limited up to when $m_k = n$ is reached where $a_p \notin \mathcal{R}(A_k)$ cannot occur anymore. Due to the non-degeneracy, we are certain of an improvement when $m_k = n$.

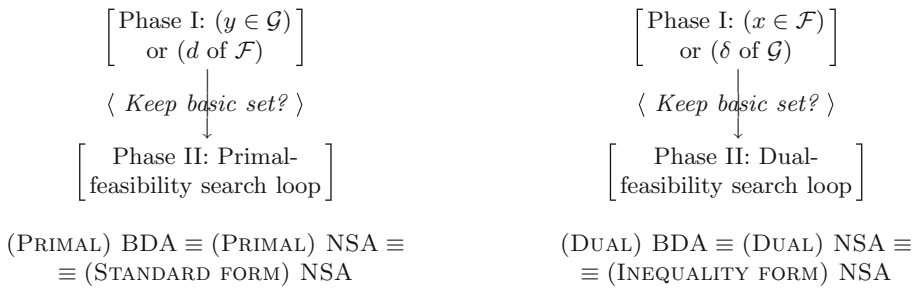

Fig. 1. Primal (left) and dual (right) two-phase approaches

Note that an anti-cycling rule like Bland's rule is required to deal with degenerate steps. Nevertheless, to prove finite termination (i.e., that no cycling occurs) we only need to have the dual non-degeneracy on iterations in which variables leave the basis.

5 Relation with Farkas Lemma and Phase I Approaches

The next question is to find a starting solution y_0 for the algorithm. For this, we can think of the Phase I approach suggested by Gass based on the single artificial variable Phase I, see [10,32,40]. Given an infeasible point $\hat{y} \geq 0$ such that $A\hat{y} \neq c$, a Phase I can be constructed using a single artificial variable z. Determine the corresponding values of the slack variables $\hat{s} = A\hat{y} - c$ and solve

$$\min z \quad \text{s.t. } Ay - \hat{s}z = c \ , \ y \geq 0 \ , \ z \geq 0 \tag{3}$$

Notice that the solution $(\hat{y}^T, 1)^T$ is feasible for this problem, where we have that $m_0 < n$. This means, one can solve this Phase I using the primal NSA method, e.g. with an infeasible solution $\hat{y} = 0$ and consequently start with the reduced basis $\mathcal{B}_0 = \{m+1\}$ using a suitable min-ratio tie-breaker. Using $\hat{y} = 0$ leads to a primal problem of Phase I

$$\min c^T d \quad \text{s.t. } \begin{pmatrix} A^T \\ c^T \end{pmatrix} d \geq -e_{m+1} \tag{4}$$

providing a primal null-space descent direction d, see [37,41] and the top leftmost part of Fig. 1.

Let us consider an alternative of what we have described so far, i.e. applying the same algorithmic scheme on two different problems, namely first on (3) and then on (P). We now consider two different algorithmic schemes on the same problem. First we use dual NSA directly on (P), applying instead of the min-ratio test a most-obtuse-angle rule, i.e., replacing $\min_{p \in \mathcal{N}} \frac{r_p}{-w_p}$ by $\min_{p \in \mathcal{N}} w_p$, where $w_p := a_p^T d < 0$. Then we apply the primal NSA to (P). In this "dual-then-primal" approach we start from an arbitrary dual infeasible point y_0, and do

not take the primal information of x into account. The nontrivial mathematical equivalence of the two proposals when having to deal with degenerate steps has been proven in [41]. A suitable normalization of w_p should be applied.

A completely different alternative to Phase I is to consider the Nonnegative Least Squares (NNLS) approach described in [4,40]. One can apply the NNLS algorithm [19] or Dax algorithm [7] to the positive semi-definite quadratic problem

$$\min_{y \geq 0} \frac{1}{2} \|Ay - c\|_2^2 \tag{5}$$

which implies solving a sequence of unconstrained least squares problems of the form $\min \|A_k z - c\|_2$ (NNLS) or of the form $\min \|A_k w - s_k\|_2$ (Dax). Such an approach, does not require artificial variables and obtains a feasible direction of the feasible region $\mathcal{F}$ (i.e., a so-called direction of $\mathcal{F}$) or a dual basic feasible solution with $m_k \leq n$. It appears that the computation can be described in terms of a primal null-space descent direction [38]. Instead of (5), one can focus on its Wolfe dual, which is a least distance strictly convex quadratic problem

$$\min \frac{1}{2} \|u\|_2^2 \quad \text{s.t.} \quad A^T u \geq v, \tag{6}$$

where $v := A^T c$. This also implies solving a sequence of least squares problems. To solve (6), one can use any of the primal, dual and primal-dual active-set methods introduced in [36]. Note that a primal direction $d := u - c$ is easy to generate.

We can also adopt the dual point of view (see top rightmost part of Fig. 1), namely using the dual NSA of [9,10] and the Gass Phase I with a single artificial variable like in (3)

$$\min z \quad \text{s.t. } A^T x - \hat{r} z \geq b \ , \ z \geq 0 \tag{7}$$

where $\hat{r} = A^T \hat{x} - b$. Consider again $\hat{x} = 0$ in $(\hat{x}^T, 1)^T$ representing a primal degenerate vertex and $\mathcal{B}_0 = \emptyset$ and a suitable min-ratio tie-breaker. Note that proving finite convergence also requires an anti-cycling rule and that in this case the dual problem corresponding to the Phase I is

$$\max(b^T, 0)^T \delta \quad \text{s.t.} \quad \begin{pmatrix} A & 0 \\ b^T & 1 \end{pmatrix} \delta = e_{n+1}, \ \delta \geq 0. \tag{8}$$

Notice that again the same algorithmic scheme can be applied to two different problems, namely to (7) and to (P).

An alternative is to use two different algorithmic schemes on the same problem. A first step can be to use primal NSA directly on (P). In this concept, we use another selection rule. Let $v_q := e_i^T (A_k^T A_k)^{-1} A_k^T a_p > 0$, where q corresponds to index i of $\mathcal{B}$. Now we replace $\min_{q \in \mathcal{B}} \frac{y_q}{v_q}$ by $\max_{q \in \mathcal{B}} v_q$. After that, one can apply the dual NSA on (P). This "primal-then-dual" approach can start from an arbitrary primal infeasible point x_0 and does not require any dual information on y. This implies a subtle variation of the Rosen-Dax method described

in Sect. 3, since it may take the objective function into account in both phases. Steepest-edge rules seem to have been recently incorporated by Pan et al. [31] into the Phase II of this alternative.

As pointed out in [40], also for the primal a least-squares-based Phase I can be used, solving

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x\|_2^2 \quad \text{s.t.} \quad A^T x \geq b, \tag{9}$$

with a quadratic algorithm when $b \notin \mathcal{R}(A^T)$. Alternatively, we can use Cline's method, see [19] implying to use NNLS to

$$\min_{\delta\geq 0} \frac{1}{2}\left\| \begin{pmatrix} b^T \\ A \end{pmatrix} \delta - e_1 \right\|_2^2. \tag{10}$$

This will generate a feasible direction δ of the dual $\mathcal{G}$ or a feasible primal point $x \in \mathcal{F}$ (by Farkas' dual lemma). Note that when $b \in \mathcal{R}(A^T)$ we can directly solve $A^T x = b$ to generate a feasible point $x \in \mathcal{F}$, but then $Ay = c$ would imply $b^T y = x^T Ay = x^T c$.

The presented Phase I approaches are mainly sparse-friendly, as we show in Sect. 6. Actually, they facilitate alternative proofs of Farkas lemma which are as elementary as the proof given by Dax in [7]. Note that the NNLS-based case does not require an anti-cycling rule in exact arithmetic. An application of this idea can be found in [13].

6 QR Factorization and Numerical Results

The efficiency of implementing the pseudo-code depends a lot on how to deal with the way to solve the set of linear equations. Mainly step 6 depends a lot on working with a QR factorization or not. We will first discuss this phenomenon in Sect. 6.1.

6.1 QR Factorization Aspects

The implementation of the algorithmic schemes given above for dense problems [26,39] can be realized using the QR factorization of A_k (adding and deleting columns). However, sparse problems are far more relevant. Therefore, we describe two sparse orthogonal approaches.

The first sparse orthogonal approach is to use a sparse QR factorization of A_k^T adding and deleting rows. The dense counterpart was studied by Powell [34]. Since A_k is formed by a subset of the columns of a fixed matrix A, we can adapt [42] Saunders technique for square matrices [43] to matrices with more rows than columns. A way to do so is to use the static data structure of George and Heath (e.g., see [17] and the references therein) but allowing row downdating on it. This facilitates taking advantage of the intermediate results obtained when dealing with the problem $\min \|A^T x - b\|_2$ by processing A^T row by row in relation to (D). The column order of A_k does not affect the density of a Cholesky factor R_k of $A_k A_k^T$. Computational experience is reported in [15].

A disadvantage of the least-squares-based Phase I approaches compared to single artificial variable approaches is that the necessity to obtain least squares solutions does not allow working with the QR factorization of A_k^T alone. In [6], Dax recommends to use an iterative relaxation technique. However, this does not facilitate making use of the relation between A_k and A_{k+1}. Another sparse orthogonal approach is to adapt the methodology of Björck [2] and Oreborn [23]. They apply a sparse NNLS algorithm via CSNE with the Cholesky factor R_k of $A_k^T A_k$ using a "short-and-fat" matrix A. Their idea is to apply an active set algorithm for the sparse least squares problem

$$\min \frac{1}{2} y^T C y + d^T y \quad \text{s.t.} \quad \underline{y} \leq y \leq \overline{y} \tag{11}$$

with a positive definite matrix C. In the problem under investigation $\overline{y} = \infty$ and $\underline{y} = 0$ and $d = -A^T c$, with $C = A^T A$ only positive semi-definite. Hence, to maintain a sparse QR factorization of A_k, we adapted [38] using a similar technique as in [3] and prevent creating C. Notice that the column ordering of A determines the order in A_k.

Table 1. NETLIB test problem instances; nnz: number of nonzeros

#	Name	Optimum value	m	n	nnz(A)	nnz(b)	nnz(c)	nnz(LP)
1	AFIRO	.46475314286E+3	51	27	102	5	7	114
2	SC50B	.70000000000E+2	78	50	148	1	5	154
3	SC50A	.64575077059E+2	78	50	160	1	10	171
4	SC105	.52202061212E+2	163	105	340	1	20	361
8	STOCFOR1	.41131976219E+5	165	117	501	27	8	536
6	ADLITTLE	−.22549496316E+6	138	56	424	82	37	543
9	BLEND	.30812149846E+2	114	74	522	30	8	560
7	SCAGR7	.23313898243E+7	185	129	465	133	53	651
10	SC205	.52202061212E+2	317	205	665	1	38	704
12	SHARE2B	.41573224074E+3	162	96	777	36	24	837
14	LOTFI	.25264706062E+2	366	153	1136	8	49	1193
15	SHARE1B	.76589318579E+5	253	117	1179	31	103	1313
17	SCORPION	−.18781248227E+4	466	388	1534	282	76	1892
22	BRANDY	−.15185098965E+4	303	220	2202	2	54	2258
19	SCAGR25	.14753433061E+8	671	471	1725	475	179	2379
20	SCTAP1	−.14122500000E+4	660	300	1872	360	154	2386
16	DUISRAEL	−.89664482186E+6	316	142	2411	171	89	2671
23	ISRAEL	.89664482186E+6	316	174	2443	89	171	2703
29	BANDM	.15862801845E+3	472	305	2494	165	118	2777
31	SCFXM1	−.18416759028E+5	600	330	2732	23	116	2871
30	E226	.18751929066E+2	472	223	2768	189	99	3056
26	SCSD1	−.86666666743E+1	760	77	2388	760	1	3149
28	AGG	.35991767287E+8	615	488	2862	131	432	3425

6.2 Numerical Experiments

We run computational experiments with slight modifications of two of the Phase I approaches comparing to TomLab LpSolve v3.0 [18] sparse implementation of the usual primal simplex method, see [12]. The chosen sparse technique depends on the used primal null-space descent direction. The details of the computational experience, both for sparse adaptations of classical test problems and for the first 31 smallest Netlib problems in Table 1[8], can be found in [12] and [14] respectively. The results reveal an advantage in terms of number of iterations, quality of solutions and execution time when suitable pivot strategies are used and with no special anti-cycling tools. Specifically, for the Netlib instance, 6812 (in 51 min) and 6950 (in 52 min) iterations versus 9586 (in 58 min) with Tomlab, as shown in Table 2. In the sparse adaptations of classical test problems, the difference is even more pronounced. A similar computational experience is reported in [21].

Table 2. A comparison of best options for each algorithm

	TomLab v3.0		SagDisp v1.1		SagFiab v1.1	
#	Iters	CtSecs	Iters	CtSecs	Iters	CtSecs
1	23	197	24	49	24	55
2	54	225	63	214	63	220
3	59	242	60	170	60	176
4	122	1109	131	775	131	780
6	151	626	144	1021	144	1038
7	196	2219	175	1648	218	2527
8	154	1334	141	1274	142	1307
9	150	741	132	1186	132	1186
10	258	6267	297	4163	297	4212
12	169	1307	196	1653	165	1329
14	345	4932	288	12073	288	11985
15	407	3323	248	2626	231	2351
16	579	6508	311	8145	304	7833
17	414	29967	375	7761	445	8689
19	952	86738	670	28688	696	29720
20	549	23695	384	9255	397	10172
22	588	12402	322	17768	326	18784
23	585	8272	431	13226	425	12781
26	583	4932	129	29017	132	29089
28	523	57743	554	10573	567	10397
29	1191	43754	568	82872	617	92550
30	864	18164	669	46258	608	35262
31	670	31594	500	24030	538	31324
	9586	58’	6812	51’	6950	52’

Sparse orthogonal approaches have in general the future that they can be parallelised and they fit well within a mixed interior-point simplex methodology. This is due to the fact that, in spite of using active-set methods, we work on top of the static structure of the Cholesky factor R of AA^T (our first approach, as in interior-point algorithms using the normal equations approach) or else on that of A^TA (our second approach, as in the interior-point method described in [1]), with $R_k \subset R$. Nonetheless, an implementation based on a dynamic revised LU factorization [27] has been shown by Pan to outperform the commercial primal simplex implementation included in MINOS [22,28,29]. However, in this case we face the same difficulties that arise when trying to parallelise the simplex method and we cannot deal with minimum-norm solutions of least-squares sub-problems easily.

7 Conclusions

In 1973 Gill and Murray presented a non-simplex active-set (NSA) method for the primal linear programming problem in inequality form $\min\{c^Tx : A^Tx \geq b\}$. We studied an NSA method for its dual linear program in standard form $\max\{b^Ty : Ay = c, y \geq 0\}$. This leads to an exterior method that facilitates working with a subset of independent columns of A which is not necessarily a square basis. We illustrate the relation with so-called basis-deficiency-allowing (BDA) simplex variations. Moreover, we provide a proof of finite termination of the method under the assumption of dual non-degeneracy.

A focus has been on two BDA features of interest related to the Phase I and its sparse implementation. The latter is carried out by solving either a sequence of sparse compatible systems or a sequence of sparse least squares problems. These "sparseness-friendly" Phase I approaches are related with Farkas' lemma and its proof. This implies an alternative proof to that by Dax in [7] which does not rely on anti-cycling pivoting rules. The computational behaviour of the tested Phase I approaches is highly encouraging.

References

1. Birge, J.R., Freund, R.M., Vanderbei, R.J.: Prior reduced fill-in solving equations in interior point algorithms. Oper. Res. Lett. **11**(4), 195–198 (1992)
2. Björck, Å.: A direct method for sparse least squares problems with lower and upper bounds. Numer. Math. **54**(1), 19–32 (1988)
3. Coleman, T.F., Hulbert, L.A.: A direct active set algorithm for large sparse quadratic programs with simple bounds. Math. Program. **45**(3), 373–406 (1989)
4. Dantzig, G.B., Leichner, S.A., Davis, J.W.: A strictly improving linear programming Phase I algorithm. Ann. Oper. Res. **46/47**(2), 409–430 (1993)
5. Dax, A.: The gradient projection method for quadratic programming. Technical report, Institute of Mathematics, Hebrew University, Jerusalem, Israel (1978)
6. Dax, A.: Linear programming via least squares. Linear Algebra Appl. **111**, 313–324 (1988)
7. Dax, A.: An elementary proof of Farkas' Lemma. SIAM Rev. **39**(3), 503–507 (1997)

8. Gay, D.M.: Electronic mail distribution of linear programming test problems. Committee Algorithms (COAL) Newslett. **13**, 10–12 (1985)
9. Gill, P.E., Murray, W.: A numerically stable form of the simplex algorithm. Linear Algebra Appl. **7**, 99–138 (1973)
10. Gill, P.E., Murray, W., Wright, M.H.: Numerical Linear Algebra and Optimization, vol. 1. Addison-Wesley, Redwood City (1991)
11. Gould, N.I.M.: The generalized steepest-edge for linear programming, Part I: theory. Technical report CORR 83-2, Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario, Canada, January 1983
12. Guerrero-García, P.: Range-space methods for sparse linear programs (in Spanish). Ph.D. thesis, Department of Applied Mathematics, University of Málaga, Spain, Málaga, Spain, July 2002
13. Guerrero-García, P., Santos-Palomo, A.: Gyula Farkas would also feel proud. Technical report MA-02-01, Department of Applied Mathematics, University of Málaga, Spain, Málaga, Spain, November 2002
14. Guerrero-García, P., Santos-Palomo, A.: A comparison of three sparse linear program solvers. Technical report MA-03-04, Department of Applied Mathematics, University of Málaga, Spain, Málaga, Spain, October 2003
15. Guerrero-García, P., Santos-Palomo, A.: Solving a sequence of sparse compatible systems. Technical report MA-02-03, Department of Applied Mathematics, University of Málaga, Spain, Málaga, Spain, October 2003
16. Guerrero-García, P., Santos-Palomo, A.: Phase-I cycling under the most-obtuse-angle pivot rule. Eur. J. Oper. Res. **167**(1), 20–27 (2005)
17. Heath, M.T.: Numerical methods for large sparse linear least squares problems. SIAM J. Sci. Stat. Comput. **5**(3), 497–513 (1984)
18. Holmström, K.: The TOMLAB optimization environment v3.0 user's guide. Technical report, Mälardalen University, Västerås, Sweden, April 2001
19. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems. Prentice-Hall, Englewood Cliffs (1974)
20. Li, W.: Bound constraints simplex method with deficient basis. Pure Appl. Math. (Xi'an) **20**(2), 171–176 (2004)
21. Li, W., Guerrero-García, P., Santos-Palomo, A.: A basis-deficiency-allowing primal Phase-I algorithm using the most-obtuse-angle column rule. Comput. Math. Appl. **51**(6/7), 903–914 (2006)
22. Murtagh, B.A., Saunders, M.A.: Minos 5.5 user's guide. Technical report SOL 83-20R, Systems Optimization Laboratory, Stanford University, Stanford, CA, July 1998
23. Oreborn, U.: A Direct Method for Sparse Nonnegative Least Squares Problems. Licentiat thesis, Department of Mathematics, Linköping University, Linköping, Sweden (1986)
24. Osborne, M.R.: Finite Algorithms in Optimization and Data Analysis. Wiley, Chichester (1985)
25. Pan, P.Q.: A basis-deficiency-allowing variation of the simplex method for linear programming. Comput. Math. Appl. **36**(3), 33–53 (1998)
26. Pan, P.Q.: A dual projective pivot algorithm for linear programming. Comput. Optim. Appl. **29**(3), 333–346 (2004)
27. Pan, P.Q.: Revised basis-deficiency-allowing simplex algorithm using the LU-factorization for linear programming. Presented at The Sixth International Conference on Optimization: Techniques and Applications, Ballarat, Australia, December 2004

28. Pan, P.Q.: A revised dual projective pivot algorithm for linear programming. SIAM J. Optim. **16**(1), 49–68 (2005)
29. Pan, P.Q.: A primal deficient-basis simplex algorithm for linear programming. Appl. Math. Comput. **196**(2), 898–912 (2008)
30. Pan, P.Q.: Linear Programming Computation, 1st edn. Springer, Cham (2014). https://doi.org/10.1007/978-3-642-40754-3
31. Pan, P.Q., Li, W., Wang, Y.: A Phase-I algorithm using the most-obtuse-angle rule for the basis-deficiency-allowing dual simplex method. OR Trans. (Chinese) **8**(2), 88–96 (2004)
32. Pan, P.Q., Pan, Y.P.: A Phase-1 approach for the generalized simplex algorithm. Comput. Math. Appl. **42**(10/11), 1455–1464 (2001)
33. Powell, M.J.D.: An example of cycling in a feasible point algorithm. Math. Program. **20**(1), 353–357 (1981)
34. Powell, M.J.D.: An upper triangular matrix method for quadratic programming. In: Mangasarian, O., Meyer, R., Robinson, S. (eds.) Nonlinear Programming 4, pp. 1–24. Academic Press, New York (1981)
35. Rosen, J.B.: The gradient projection method for nonlinear programming. Part I: linear constraints. J. Soc. Industr. Appl. Math. **8**(1), 181–217 (1960)
36. Santos-Palomo, A.: New range-space active-set methods for strictly convex quadratic programming. In: Olivares-Rieumont, P. (ed.) Proceedings of the III Conference on Operations Research, vol. 2, p. 27. Universidad de La Habana, Cuba, March 1997
37. Santos-Palomo, A.: The sagitta method for solving linear programs. Eur. J. Oper. Res. **157**(3), 527–539 (2004)
38. Santos-Palomo, A., Guerrero-García, P.: Solving a sequence of sparse least squares problems. Technical report MA-03-03, Department of Applied Mathematics, University of Málaga, Spain, Málaga, Spain, September 2003
39. Santos-Palomo, A., Guerrero-García, P.: Computational NETLIB experience with a dense projected gradient sagitta method. Technical report MA-05-05, Department of Applied Mathematics, University of Málaga, Spain, Málaga, Spain, September 2005
40. Santos-Palomo, A., Guerrero-García, P.: A non-simplex active-set method for linear programs in standard form. Stud. Inform. Control **14**(2), 79–84 (2005)
41. Santos-Palomo, A., Guerrero-García, P.: Sagitta method with guaranteed convergence. Technical report MA-05-02, Department of Applied Mathematics, University of Málaga, Spain, Málaga, Spain, February 2005
42. Santos-Palomo, A., Guerrero-García, P.: Updating and downdating an upper trapezoidal sparse orthogonal factorization. IMA J. Numer. Anal. **26**(1), 1–10 (2006)
43. Saunders, M.A.: Large-scale linear programming using the Cholesky factorization. Technical report CS-TR-72-252, Computer Science Department, Stanford University, Stanford, CA, January 1972