FACTA UNIVERSITATIS (NIŠ)
SER. MATH. INFORM. 24 (2009), 29–37

FINITE TERMINATION AND DECREASING DIMENSIONS IN MEHROTRA'S PRIMAL-DUAL ALGORITHM*

Nebojša V. Stojković and Predrag S. Stanimirović†

Abstract. In this paper we propose a modification of the Mehrotra's primal dual algorithm. This modification reduces dimensions of the problem, improves stability of the method and eliminates the need for the finite termination algorithm. A few illustrative numerical examples which compare the modification with respect to the original method are reported.

1. Introduction

We are concerned with the linear programming problem given in the general form, which we write in the standard form as

$$\min\ c^T x \ \text{ subject to } \ Ax = b, \quad x \geqslant 0, \tag{1.1}$$

where $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and A is an $m \times n$ real matrix and c^T is transpose of the vector c. The dual problem for the problem (1.1) is

$$\max\ b^T \lambda \ \text{ subject to } \ A^T \lambda + s = c, \quad s \geq 0, \tag{1.2}$$

where $\lambda \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$. It is known that the vector $x^* \in \mathbb{R}^n$ is a solution of (1.1) if and only if there exist vectors $s^* \in \mathbb{R}^n$ and $\lambda^* \in \mathbb{R}^m$ such that the following conditions hold:

$$A^T \lambda^* + s^* = c, \tag{1.3}$$

$$Ax^* = b, \tag{1.4}$$

$$x_i^* s_i^* = 0, \quad i = 1, \ldots, n, \tag{1.5}$$

$$(x^*, s^*) \geq 0. \tag{1.6}$$

Received October 17, 2008.
2000 *Mathematics Subject Classification.* 90C05
*Supported by Ministry of Science and Technological Development, Republic of Serbia, Grant no. 144011.
†Corresponding author

All primal-dual methods generate iterates (x^t, λ^t, s^t) that satisfy the bounds (1.6) strictly and instead the condition (1.5) deal with the condition $x_i s_i = \tau$, $i = 1, \ldots, n$, where $\tau \to 0$.

For every solution (x^*, λ^*, s^*) we know that $x_j^* = 0$ and/or $s_j^* = 0$ for all $j = 1, \ldots, n$. Let

$$B = \{j \in \{1, \ldots, n\} | \, x_j^* > 0\}, \quad N = \{j \in \{1, \ldots, n\} | \, s_j^* > 0\}.$$

It is easy to see that $B \cap N = \emptyset$. Primal-dual strictly feasible set F^0 is

$$F^0 = \{(x, \lambda, s) | \, Ax = b, \, A^T\lambda + s = c, \, (x, s) > 0\}.$$

The motivation for the present paper is based on the fact that the primal-dual algorithm only generates a sequence converging towards the optimal solution and it needs the finite termination algorithm to find an exact primal-dual solution. Also, the finite termination algorithm is by itself a relatively complicated nonlinear optimization problem; see, for instance [1], [2], [3] and [10]. Our main goal is to develop an algorithm which avoid the application of the finite termination algorithm. Moreover, we intend to reduce dimensions of the problem as well as to improve the stability and centrality of the iterative sequence.

The paper is organized as follows. In the second section we provide the theoretical fundament for a modification of the Mehrotra's method. In the third section we select a small set of test problems from the public domain collections of linear programming solvers to illustrate the presented algorithm. Finally, in the last section we present a few concluding remarks and comparisons of the modified method with respect to the Mehrota's method and known linear programming solvers.

2. Modified Mehrotra's primal-dual algorithm

Note that there are several variants of the Mehrotra's algorithm, all quite similar. In this section we briefly describe a variant of the Mehrotra's algorithm which is similar to the variant used in the code of *PCx* (see for example [3, 6, 9]).

Step 1. Generate the starting iteration (x^t, λ^t, s^t), $t = 0$.

Step 2. Calculate the residues

$$r_b = Ax^t - b, \qquad r_c = A^T\lambda^t + s^t - c$$

and check the following stopping criteria

$$\frac{||r_b||}{1 + ||b||} \leqslant \epsilon, \quad \frac{||r_c||}{1 + ||c||} \leqslant \epsilon, \quad \frac{|c^T x - b^T \lambda|}{1 + |c^T x|} \leqslant \epsilon.$$

If the stopping criterion is satisfied, return the output x^t; otherwise, go to *Step* 3.

Step 3. Form the matrices S, X and the vector e, defined by

$$S = \operatorname{diag} s_1, \ldots, s_{k+q}, \; X = \operatorname{diag} x_1, \ldots, x_{k+q}, \; e = (1, \ldots, 1)^T \in \mathbb{R}^{k+q}.$$

Step 4. Set $D = S^{-1/2}X^{1/2}$, and $r_{xs} = XSe$ and solve the following system with respect to $(\Delta x^{aff}, \Delta\lambda^{aff}, \Delta s^{aff})$:

$$\begin{aligned} AD^2A^T\Delta\lambda^{aff} &= -r_b - A(S^{-1}Xr_c - S^{-1}r_{xs}), \\ \Delta s^{aff} &= -r_c - A^T\Delta\lambda^{aff}, \\ \Delta x^{aff} &= -S^{-1}(r_{xs} + X\Delta s^{aff}). \end{aligned}$$

It is easy to verify the following

$$S^{-1} = \operatorname{diag} 1/s_1, \ldots, 1/s_{k+q}, \quad D^2 = \operatorname{diag} x_1/s_1, \ldots, x_{k+q}/s_{k+q}.$$

Step 5. Calculate the measure of duality $\mu = \frac{1}{k+q}\sum_{i=1}^{k+q} x_i s_i$.

Step 6. Calculate the conditions for non-negativity of the iterative point

$$\begin{aligned} \alpha_{aff}^{pri} &= \max\{\alpha \in [0,1] : x^t + \alpha\Delta x^{aff} \geq 0\} \\ \alpha_{aff}^{dual} &= \max\{\alpha \in [0,1] : s^t + \alpha\Delta s^{aff} \geq 0\}. \end{aligned}$$

Step 7. Calculate

$$\mu_{aff} = \frac{1}{n}(x^t + \alpha_{aff}^{pri}\Delta x^{aff})(s^t + \alpha_{aff}^{dual}\Delta s^{aff}) \text{ and } \sigma = \left(\frac{\mu_a ff}{\mu}\right)^3.$$

Step 8. Compute $r_{xs} = -\sigma\mu e + \Delta X^{aff}\Delta S^{aff}e$, where

$$\Delta X^{aff} = \operatorname{diag}\Delta x_1^{aff}, \ldots, \Delta x_n^{aff}, \ \Delta S^{aff} = \operatorname{diag}\Delta s_1^{aff}, \ldots, \Delta s_n^{aff},$$

and solve the following system for $(\Delta x^{cor}, \Delta\lambda^{cor}, \Delta s^{cor})$:

$$\begin{aligned} AD^2A^T\Delta\lambda^{cor} &= AS^{-1}r_{xs}, \\ \Delta s^{cor} &= -A^T\Delta\lambda^{cor}, \\ \Delta x^{cor} &= -S^{-1}(r_{xs} + X\Delta s^{cor}). \end{aligned}$$

Step 9. Set

$$(\Delta x^t, \Delta\lambda^t, \Delta s^t) = (\Delta x^{aff}, \Delta\lambda^{aff}, \Delta s^{aff}) + (\Delta x^{cor}, \Delta\lambda^{cor}, \Delta s^{cor})$$

Step 10. Calculate the parameters

$$\begin{aligned} \alpha_{max}^{pri} &= \max\{\alpha \geq 0 : x^t + \alpha\Delta x^t \geq 0\} \\ \alpha_{max}^{dual} &= \max\{\alpha \geq 0 : s^t + \alpha\Delta s^t \geq 0\}. \end{aligned}$$

Step 11. Set

$$\begin{aligned} \alpha_t^{pri} &= \min\{0.99\alpha_{max}^{pri}, 1\} \\ \alpha_t^{dual} &= \min\{0.99\alpha_{max}^{dual}, 1\}. \end{aligned}$$

Step 12. Compute the next iteration

$$\begin{aligned} x^{t+1} &= x^t + \alpha_t^{pri}\Delta x^t, \\ (\lambda^{t+1}, s^{t+1}) &= (\lambda^t, s^t) + \alpha_t^{dual}(\Delta\lambda^t, \Delta s^t), \end{aligned}$$

put $(x^t, \lambda^t, s^t) = (x^{t+1}, \lambda^{t+1}, s^{t+1})$ and go to *Step 2*.

In this paper we propose the following modifications of *Step 12* and *Step 2*. The modification of *Step 12* is based on the elimination of variables x_i and s_i converging to zero. It is possible to use the following criterion which guarantees the convergence of some variables to zero. For any point $(x, \lambda, s) \in N_{-\infty}(\gamma)$, we can estimate the sets B and N when μ is sufficiently small [9][Theorem 7.2.1], as follows:

$$B(x,s) = \{i \in \{1,\dots,n\} | \, x_i \geq s_i\}, \quad N(x,s) = \{1,\dots,n\}\backslash B(x,s), \tag{2.1}$$

where

$$N_{-\infty}(\gamma) = \{(x,\lambda,s) \in F^0 | \, x_i s_i \geq \gamma\mu \text{ for all } i = 1,\dots,n\}$$

and $\gamma \in (0,1)$. Also, it is known that there is a threshold value $\overline{\mu}$ such that for all (x, λ, s) that satisfy

$$(x,\lambda,s) \in N_{-\infty}(\gamma) \subset F^0, \;\; 0 < \mu = x^T s/n \leq \overline{\mu},$$

we have $B(x,s) = B$ and $N(x,s) = N$, i.e. the actual index sets B and N are estimated correctly by (2.1) [9].

We need small real numbers *eps*1 and *eps*2 as criteria for the application of reductions. We use the vector *xzero* to remember the indices of deleted elements of the vecor x. The empty list is the initial value for the vector *xzero*.

Step 12M. Compute the next iteration

$$\begin{aligned} x = x^{t+1} &= x^t + \alpha_t^{pri}\Delta x^t, \\ (l,s) = (\lambda^{t+1}, s^{t+1}) &= (\lambda^t, s^t) + \alpha_t^{dual}(\Delta\lambda^t, \Delta s^t). \end{aligned}$$

Note that it is assumed that first q constraints are inequalities, while the constraints with the order numbers $q+1,\dots,q+m$ are equalities. If the condition $xs = x^T s < eps1$ is satisfied, perform two cycles, *Cycle 1* and *Cycle 2*, defined as in the following.

Cycle 1. For each $i = 1,\dots,k$, if the conditions

$$x_i = x_i^{t+1} < eps1, \quad x_i \leq s_i = s_i^{t+1} \tag{2.2}$$

are satisfied, run the following steps 1.1-1.4.

1.1. Delete i-th element from the vectors x, c and s, and delete i-th column from the matrix A.

1.2. Put $k = k - 1$.

1.3. Insert the index i at the beginning of the vector *xzero*.

1.4. Set $i = i - 1$.

Cycle 2. For each $i = k+1, \ldots, k+q$ execute steps 2.1-2.4.

2.1. Put *False* as the value of the logical variable *lg*. This variable is used as the indicator that any element from x is deleted at any time.

2.2. Under the conditions (2.2) run the following:

2.2.1. Put $lg = True$.

2.2.2. Delete i-th element from x, c, s and i-th column from A.

2.2.3. Put $q = q - 1$.

2.2.4. Replace $(i-k)$-th and the last elements in A, x and $l = \lambda$.

2.3. In the case $i \leq Length[s]$, if the conditions

$$lg = False, \quad s_i = s_i^{t+1} < eps2, \quad s_i \leq x_i = x_i^{t+1}$$

are satisfied, execute the following steps:

2.3.1. Delete i-th element from x, c and s, delete $(i-k)$-th element of b and $l = \lambda$, and delete $(i-k)$-th row and i-th column from A.

2.3.2. Put $q = q - 1$.

2.4. In the case $lg = True$ put $i = i - 1$.

Finally, as in the usual method, put $(x^t, \lambda^t, s^t) = (x^{t+1}, \lambda^{t+1}, s^{t+1})$ and go to *Step 2M*, a modification of *Step 2*.

Step 2M. Check the stopping criteria as in *Step 2*.

If these criteria are not satisfied, go to the next iteration. When these criteria are satisfied, perform the reconstruction of the vector x, inserting zero values at the corresponding positions of the vector x^t, where corresponding elements are deleted. For this purpose, we use the list *xzero* and apply the following *For* cycle:

For each $i = 1, \ldots, n = Length[xzero]$ insert the zero at the i-th position in the vector x^t.

Then return x as the output.

Remark 2.1. An important practical issue is a choice of an indicator $B(x,s)$ for the optimal partition B. Besides the criteria (2.2) we also used an alternative criterion $x^T s / x1^T s1 = r < eps2$, where $x1$ and $s1$ are values for x and s in the previous iterative step [4]. Our numerical experience shows that the most of tested problems in the last stages of iterative processes take the value $r \approx 0.01$. This means that the superlinear convergence is achieved for the values $eps2 = 0.01$. But, some of these problems possesses a quite satisfactory convergence even for the values $eps2 = 0.1$. After the application of this criterion we get almost the same results as the previous criterion is applied.

Remark 2.2. The settings $x_i = 0, i \in I, s_{k+j} = 0, j \in J$ in *Step 12M* are analogous to the settings $x_i^* = 0, i \in N(x,s), s_i^* = 0, i \in B(x,s)$ which are known from the finite termination algorithm. Our strategy differs from the classical finite termination algorithm

in the fact that we do not estimate all elements from the sets N and B. Moreover, we apply these assignments successively during the iterations, performing only simple verifications.

3. Numerical experiences

Example 3.1. In this example we consider a subset of known test problems in the literature. Several results obtained by the modified algorithm are arranged in the Table 1. Notice that the variable *eps* denotes the stopping criterion in the algorithm and the variables *eps*1 and *eps*2 are described in *Step 12M*. Also, the problem *KbM* is obtained by dropping upper bounds from the known problem *Kb2*. An additional column, denoted by $TMod/T$, contains the quotient of the processor time which is required for the modified and the original method Mehrotra's primal-dual method.

Problem	*eps*	*eps*1, 2	MPD opt. value	It.	MMPD opt. value	It.	TM/T
Afiro	10^{-13}	10^{-1}	-464.7531428571353	12	-464.75314285714694	12	0.73
Bandm	10^{-8}	10^{-4}	-158.6280184317434	19	-158.6280184499003	18	0.86
Blend	10^{-12}	10^{-2}	-30.81214984582682	13	-30.812149845827857	12	0.65
Israel	10^{-12}	10^{-4}	-896644.8218630098	25	-896644.8218630177	25	0.86
Kb2	10^{-10}	10^{-3}	-1749.9001299102652	17	-1749.900129938148	16	0.78
Adlittle	10^{-13}	10^{-2}	225494.96316236982	16	225494.9631623751	16	0.82
Sc50a	10^{-13}	10^{-3}	-64.57507705856638	13	-64.5750770585622	13	0.86
Sc50b	10^{-12}	10^{-3}	-69.99999999999217	11	-70.00000000000014	11	0.92
Sc105	10^{-12}	10^{-3}	-52.202061211707561	14	-52.202061211707324	14	0.82
Sc205	10^{-12}	10^{-4}	-52.20206121170699	18	-52.20206121170724	17	0.88
Scagr7*	10^{-13}	10^{-5}	$-2.331389824329545\times10^{6}$	15	$-2.3313898243310144\times10^{6}$	16	0.98
Sctap1	10^{-8}	10^{-6}	1412.2500007430447	15	1412.2499975456915	15	0.98
Share2b	10^{-11}	10^{-4}	-415.7322407432927	12	-415.73224074038535	12	0.83
Lotfi	10^{-8}	10^{-4}	-25.264706032192482	18	-25.264706046693806	18	0.88
Agg2	10^{-8}	10^{-4}	$-2.023925235593424\times10^{7}$	22	$-2.023925235068718\times10^{7}$	22	0.98
Agg3	10^{-10}	10^{-4}	$1.031211593510225\times10^{7}$	23	$1.031211593574227\times10^{7}$	21	0.92

Table 1. Comparison of the Mehrotra's algorithm and its modification.

We also illustrate the claim that the finite termination algorithm is unnecessary in our algorithm. Consider, for example, the test problem *Afiro*. The optimal point generated by the modified algorithm possesses zero coordinates at the positions which are determined by the following vector *xzero* {17, 17, 17, 12, 12, 12, 12, 12, 12, 11, 9, 2, 2, 2, 2, 1, 28}.

The optimal point *xm* is equal to

{0, 80.00000000000023,0,0,0,0,18.21428571428548,51.49692904472464,73.89388764455072,
25.49999999999969,500.0000000000107,475.9200000000041,24.08000000000006,0,
214.9999999999961,147.786872899196,0,54.50000000000029,0,0,0,0,0,0,
339.9428571428572,236.1559842436612,63.54835534665428,0,84.79999999999951,
69.71121475901005,0,0,0}.

On the other side, the usual Mehrotra's method gives the optimal point x, equal to the vector

{$2.215848062666221\times10^{-12}$,79.99999999999929,$1.312660664489544\times10^{-13}$,$4.394145729655261\times10^{-13}$,
$4.353491369592836\times10^{-13}$,$5.774260579869571\times10^{-13}$,18.21428571428262,51.50061507109443,
73.89779483249926,25.50000000000001,500.00000000001,475.9199999999925,24.08000000000016,
$7.65658212305429\times10^{-12}$,214.9999999999957,147.7683249031915,$2.425875538820393\times10^{-13}$,

54.49999999999952,2.415068530966149×10^{-13},2.4034606732123 15×10^{-13},1.409009221948509×10^{-13},
4.755857873948911×10^{-13},4.713557464415677×10^{-13},4.670668078581763×10^{-13},339.9428571428506,
236.1745322396589,63.54037970837263,3.686831036665461×10^{-14},84.79999999999963,
69.71490078537609,2.196141063804613×10^{-13},2.170131429848782×10^{-13},1.773714715268836×10^{-13}}.

The similar solution is generated by the solver *PCx* also. Small elements in x, which can generate badly conditioned matrix D^{-2}, are eliminated during the application of the modified method. Similarly, small elements in s which can generate badly conditioned matrix D^2, are also avoided in calculations of the modified algorithm. For this purpose, the modified algorithm produces the better solution with respect to the original one.

Example 3.2. Consider the following small and illustrative problem from [5]:

$$\begin{array}{lccccccccccc} \min & & & & & & 3x_3 & + & 2x_4 & + & x_5 & & \\ \text{subject to} & x_1 & + & 2x_2 & + & x_3 & & & & & & = & 1000.02, \\ & x_1 & + & x_2 & & & + & x_4 & & & & = & 1000.01, \\ & x_1 & - & x_2 & & & & & + & x_5 & = & 999.99. \end{array}$$

Applying the modified algorithm with $eps = 10^{-12}$, $eps1 = 10^{-4}$ and $eps2 = 10^{-3}$ we get the optimal value 0 and the extreme point {`999.9999999998636, 0.01000000039754838,0,0,0`}. On the other side, the usual algorithm is not able to make the result with the same precision 10^{-12}. It is interesting that *PCx* is unable to solve this problem, too. Without the iterative refinement, *PCx* stops after six iterations with the status `infeasible`. Using the iterative refinement, *PCx* stops after six iterations with `unknown` status.

4. Comparisons and concluding remarks

We describe an improvement of the primal-dual interior point method which is mainly based on the Mehrotra's algorithm, restated in [3] and [9]. We also develop a research code for the implementation of the Mehrotra's algorithm and the modification. For this purpose we select the package MATHEMATICA [7], [8]. At this moment, we do not develop a software with attractive performances relative to the processor time with respect to known robust *IPM* codes. Our main aim is to present a comparison between the Mehrotra's primal-dual interior point method and its modification, described in the third section.

After the application of the proposed modification of the Mehrotra's algorithm we have the following advantages with respect to original one, which are demonstrated in the above presented examples:

- The modified method allows a better precision in the final solution.
- The modification improves stability and centrality of the iterative sequence.

For example, during the solving the problem *Afiro* by the modified method, badly conditioned matrices which cause significant numerical errors are not arise. However, during the application of the usual method we observe badly conditioned matrices after the ninth step. Similar situation happens in the problem *Blend*, where the usual method causes badly conditioned matrix after the tenth step. Possible appearance of badly conditioned matrices can be avoided using more carefully selected values for some of parameters *eps*1 or *eps*2.

• The modified method is faster with respect to the original one, although it incorporates the additional elimination of some rows and columns and later reconstruction of the solution. This fact is justified by the last column in Table 1. Also, let us observe that the modification reduces the processor time to solve the problems *Blend* and *Kb2*, in spite of one additional step which is required in the modification.

• The modification decreases dimensions of the problem and the memory space requirements.

In the Table 2, by *Dim[AS]* we denote starting dimensions of the matrix A and *Dim[AF]* denotes dimensions of the final matrix A. Also, the maximum number of bytes of memory used to store the initial matrix A is denoted by *ByteCount[AS]*, while *ByteCount[AF]* denotes the maximum number of bytes of memory required to store the matrix A after the final iterative step.

Problem	Dim[AS]	Dim[AF]	ByteCount[AS]	ByteCount[AF]
Afiro	27×52	21×16	28864	7336
Blend	74×114	60×56	170816	68904
KbM	43×68	27×6	59712	3808
Kb2	52×77	33×33	81768	22864
Adlittle	56×138	46×61	156152	57616

Table 2. Further comparison of the Mehrotra's algorithm and its modification

Moreover, we observe the following additional properties of the modified method.

• In the modified algorithm the finite termination algorithm is unnecessary.

• It is possible to use the modification in any primal-dual algorithm, not only in the Mehrotra's.

• The described modification is applicable in already existing computer primal-dual codes.

• Since the modification reduces dimensions of the problem during the iterative steps, it is really to expect that it is especially efficient for large scale linear programming problems.

Because of high numerical characteristics of the package MATHEMATICA, even the first version of the implementation gives relatively satisfactory performances. For example, sometimes we get the result with better precision with respect to traditional implementations.

REFERENCES

1. E. D. Andersen, J. Gondzio, S. Meszaros, and X. Xu: *Implementation of interior point methods for large scale linear programming*. Hec-Geneve, Technical Report 96/3, (1996).

2. E. D. Andersen and Y. Ye: *Combining interior-point and pivoting algorithms for linear programming*. Technical report, Department of Menagement Sciences, The University of Iowa, (1994).
3. J. Czyzyk, S. Mehrotra, and S. J. Wright: *PCx User Guide*. Optimization Thechnology Center, Technical Report 96/01, 1996.
4. A. S. El-Bakry, R.A.Tapia, and Y. Zhang: *A study of indicators for identifying zero variables in interior-point methods*. SIAM Rev. **36(1)**(1994), 45–72.
5. V. V. Kovačević-Vujčić and M. D. Ašić: *Stabilization of interior-point methods for linear programming*. Computational Optimization and Applications **14** (1999) 331–346.
6. S. Mehrotra: *On the implementation of a primal-dual interior point method*. SIAM J. on Optimization **2** (1992) 575–601.
7. S. Wolfram: *The Mathematica Book*. 4th ed., Wolfram Media/Cambridge University Press, 1999.
8. S. Wolfram: *The Mathematica Book*. 5th ed., Wolfram Media, 2003.
9. S. J. Wright: *Primal-Dual Interior point methods*. SIAM Philadelphia, 1997.
10. Y. Ye: *On the finite convergence of interior-point algorithms for linear programming*. Mathematical Programming **57** (1992), 325–336.

Nebojša V. Stojković
Faculty of Economics
Trg Kralja Aleksandra 11
18000 Niš, Serbia
nebojsas@eknfak.ni.ac.rs

Predrag S. Stanimirović
Faculty of Science and Mathematics
Department of Mathematics and Informatics
Višegradska 33, 18000 Niš, Serbia
pecko@pmf.ni.ac.rs