Augmented and Normal Equations System in Mehrotra's Primal-dual Algorithm

Predrag S. Stanimirović, Nebojša V. Stojković, Branimir Momčilović and Zoran Jovanović

Abstract

In this paper[1] we compare two variants of Mehrotra's primal dual algorithm which are based on the augmented and normal equations system, respectively. An implementation of corresponding algorithms in the package MATHEMATICA, version 4.1, is used for the comparison. Numerical examples are reported applying the program on some *Netlib* test problems.

1 Introduction

We consider the linear programming problem in the general form, which comprehends both equality and inequality constraints. This problem can be transformed into the equivalent standard form

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

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

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

where $\lambda \in \mathbf{R}^m$ and $s \in \mathbf{R}^n$ and b^T, A^T denote transpose of the vector b and the matrix A, respectively. It is known that the vector $x^* \in \mathbf{R}^n$ is a solution of (1.1) if and only if there exist vectors $s^* \in \mathbf{R}^n$ and $\lambda^* \in \mathbf{R}^m$ such that the following conditions hold:

$$\begin{aligned} A^T \lambda^* + s^* &= c, \\ Ax^* &= b, \\ x_i^* s_i^* &= 0, \quad i = 1, \dots, n, && (1.3) \\ (x^*, s^*) &\geq 0. && (1.4) \end{aligned}$$

[1]Presented at the IMC "Filomat 2001", Niš, August 26–30, 2001

2000 Mathematics Subject Classification: 90C05

Keywords: Linear programming, primal-dual interior point method, augmented system, normal equations system, MATHEMATICA

All primal-dual methods generate iterates (x^t, λ^t, s^t) that satisfy the bounds (1.4) strictly, that is, $x^t > 0$ and $s^t > 0$, and instead the condition (1.3) deal with the condition $x_i s_i = \tau$, $i = 1, \ldots, n$, where $\tau \to 0$.

Known linear programming codes based on interior point methods have been developed mainly in programming languages FORTRAN, C and MATLAB. If performance is the ultimate goal, then the code should be implemented in C or FORTRAN. The implementation described in the present paper should be of interest to those wishing to become familiar with various variants of Mehrotra's interior point method for linear programming by experimenting with the implementation based on the convenience of a symbolic language rather than having to deal with the "nitty-gritty" details provided by use of procedural programming languages.

About the primal-dual interior point codes in MATHEMATICA see for example a recently published book [4]. But, these codes are not based on the Mehrotra's predictor-corrector algorithm, which is used as a fundament for most interior-point codes since 1990. The Mehrotra's predictor-corrector method incorporates a number of heuristics that have been developed during ten years of computational experience and also allows an adaptive choice of the centering parameter at each iteration.

The main numerical effort in Mehrotra's primal dual algorithm is to solve two systems of linear equations, presented in the so called augmented system form or the normal equations form. So far in the literature the normal equation system is more widely used. Starting from [2], several researches have decided to incorporate the augmented system approach [8]. The computational effort (flops) of the two competitive approaches is studied in [8]. We incorporate the capability of symbolic and numeric computation available in MATHEMATICA, version 4.1, and develop experimental codes for both variants of the algorithm. After that we will compare behavior of these algorithms on the set of test linear programming problems.

2 Description of algorithms

It is known that the linear systems to be solved at each primal-dual iteration can be formulated in three equivalent ways. The unreduced form for infeasible-interior-point algorithm is

$$\begin{bmatrix} 0 & A & 0 \\ A^T & 0 & I \\ 0 & S & X \end{bmatrix} \begin{bmatrix} \Delta\lambda \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_b \\ -r_c \\ -XSe + \sigma\mu e \end{bmatrix}, \tag{2.1}$$

where $\mu = x^T s/n$ and $r_b = Ax - b$, $r_c = A^T\lambda + s - c$.

Eliminating Δs from (2.1) and using the notation $D = S^{-1/2}X^{1/2}$, we obtain augmented system and after the elimination Δx from we obtain the normal equations system. The details will be given in the description of the algorithm.

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 [5], [12]).

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

$$\frac{||r_b||}{1+||b||} \le \epsilon, \qquad \frac{||r_c||}{1+||c||} \le \epsilon, \qquad \frac{|c^Tx - b^T\lambda|}{1+|c^Tx|} \le \epsilon.$$

In practice, it is rare that the third condition is satisfied and at the same time other conditions do not hold. Consequently, the most important and perhaps the only condition that really has to be checked is the third condition [1]. An 8-digits exact solution ($\epsilon = 10^{-8}$) is typically required in the literature.

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 as

$$S = \operatorname{diag}(s_1, \dots, s_n), \ X = \operatorname{diag}(x_1, \dots, x_n), \ e = (1, \dots, 1)^T \in \mathbf{R}^n.$$

Step 4. Compute $D = S^{-1/2}X^{1/2}$, $r_{xs} = XSe$ and solve one of the following two systems with respect to $(\Delta x^{aff}, \Delta\lambda^{aff}, \Delta s^{aff})$. In the normal equations system case solve the system

$$AD^2A^T\Delta\lambda^{aff} = -r_b - A(S^{-1}Xr_c - S^{-1}r_{xs}) \tag{2.2}$$

and compute the increments

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

where

$$S^{-1} = \operatorname{diag}(1/s_1, \dots, 1/s_n), \quad D^2 = \operatorname{diag}(x_1/s_1, \dots, x_n/s_n);$$

in the augmented system case solve

$$\begin{bmatrix} 0 & A \\ A^T & -D^{-2} \end{bmatrix} \begin{bmatrix} \Delta\lambda^{aff} \\ \Delta x^{aff} \end{bmatrix} = \begin{bmatrix} -r_b \\ -r_c + X^{-1}r_{xs} \end{bmatrix}, \tag{2.3}$$

and compute the increment $\Delta s^{aff} = -r_c - A^T\Delta\lambda^{aff}$.

Step 5. Calculate the measure of duality $\mu = \frac{1}{n}\sum_{i=1}^{n} x_is_i = \dfrac{x^Ts}{n}$.

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

$$\alpha_{aff}^{pri} = \max\{\alpha \in [0,1] \ : \ x^t + \alpha \Delta x^{aff} \geq 0\} = Min[1, \{x_i^t/\Delta x_i^{aff}, \ \Delta x_i^{aff} < 0\}],$$
$$\alpha_{aff}^{dual} = \max\{\alpha \in [0,1] \ : \ s^t + \alpha \Delta s^{aff} \geq 0\} = Min[1, \{s_i^t/\Delta s_i^{aff}, \ \Delta s_i^{aff} < 0\}].$$

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 one of the following two systems for $(\Delta x^{cor}, \Delta\lambda^{cor}, \Delta s^{cor})$. In the normal equations system case solve the system

$$AD^2 A^T \Delta\lambda^{cor} = AS^{-1} r_{xs}, \tag{2.4}$$

and compute

$$\Delta s^{cor} = -A^T \Delta\lambda^{cor},$$
$$\Delta x^{cor} = -S^{-1}(r_{xs} + X \Delta s^{cor});$$

in the augmented system case solve the system

$$\begin{bmatrix} 0 & A \\ A^T & -D^{-2} \end{bmatrix} \begin{bmatrix} \Delta\lambda^{cor} \\ \Delta x^{cor} \end{bmatrix} = \begin{bmatrix} 0 \\ X^{-1} r_{xs} \end{bmatrix}, \tag{2.5}$$

and compute $\Delta s^{cor} = -A^T \Delta\lambda^{cor}$.

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

$$\alpha_{max}^{pri} = \max\{\alpha \geq 0 \ : x^t + \alpha \Delta x^t \geq 0\} = Min[1, \{x_i^t/\Delta x_i^t, \ \Delta x_i^t < 0\}]$$
$$\alpha_{max}^{dual} = \max\{\alpha \geq 0 \ : s^t + \alpha \Delta s^t \geq 0\} = Min[1, \{s_i^t/\Delta s_i^t, \ \Delta s_i^t < 0\}].$$

Step 11. Set

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

Step 12. Compute the next iteration

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

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

We propose a simple algorithm for computation of an initial solution which will be well centered and satisfies at least one constraint, if it is possible. Moreover, the proposed algorithm generates the initial solution about ten times faster than a single interior point iteration.

Algorithm 1.

Step 1.1. Compute the quantity $aux = \max\{q_1, \ldots, q_m\}$, where

$$q_i = \begin{cases} \max\left\{\dfrac{b_i}{\sum_{j=1}^n a_{ij}}, 1\right\}, & \sum_{j=1}^n a_{ij} \neq 0, \\ 1, & \sum_{j=1}^n a_{ij} = 0,\ i = 1, \ldots, m. \end{cases} \tag{2.6}$$

Step 1.2. Generate starting points x, and s, whose all coordinates are equal to aux, and the starting point l, with all coordinates equal to zero.

Motivated by [9], we generate an alternative initial solution, which is generated in a similar way as in *Step 1.1* and *Step 1.2*, using the quantity $aux = \max\{q_1, \ldots, q_m\}$, where

$$q_i = \begin{cases} \max\left\{\dfrac{b_i}{\sqrt{\sum_{j=1}^n a_{ij}^2}}, 1\right\}, & \sum_{j=1}^n a_{ij}^2 \neq 0, \\ 1, & \sum_{j=1}^n a_{ij}^2 = 0,\ i = 1, \ldots, m. \end{cases} \tag{2.7}$$

3 Numerical experiences

Example 3.1 In this example we consider a subset of known test problems in the literature. Notice that the precision is *eps*$= 10^{-8}$ in all cases and the starting point is selected in accordance with (2.6). Values TA and TN represent the processor time required to solve the problem using the augmented and normal system, respectively.

Table 1.

Problem	Dimensions	TA/TN		Problem	Dimensions	TA/TN
Adlittle	56×138	0.76		Lotfi	153×366	0.64
Afiro	27×51	0.95		Sc105	105×163	0.51
Agg	488×615	0.52		Sc205	205×203	0.55
Agg2	516×758	0.46		Sc50b	50×78	0.72
Agg3	516×758	0.57		Sc50a	50×78	0.66
Bandm	305×472	0.63		Scagr7	129×185	0.61
Blend	74×114	0.75		Sctap1	300×660	0.78
Israel	174×316	0.69		Share2b	96×162	0.66
Kb2	43×68	0.69		Stocfor1	117×165	0.60

Note that the augmented system approach is faster in all cases in the Table 1. Observe that the difference in the processor time between the augmented and normal equations system decreases proportionally with the dimensions of the problem. Similar results are obtained using the starting point generated by (2.7).

Example 3.2 Algorithms for the construction of the starting point has no influence on numerical stability of the iterative process in all test problems from Table 1. But, in the case of the augmented system approach, there exist badly conditioned test problems whose numerical stability essentially depends of the choice of starting point. Note that in the normal system approach, these test problems are remain badly conditioned regardless of the choice of starting point. To illustrate this claim, regard the next small test problems. In [10] it is given an interesting example of dimensions 18×18 which can not be solved by known linear programming solvers PCx and $HOPDM$. Using the starting point in accordance with (2.6) and the augmented system approach we get the near-optimal value 0.0000231958320761213. On the other hand, using (2.7) for the starting point we get much better result 8.4656617528504510^{-6}.

Example 3.3 The second example is the following very simple test problem

$$\begin{array}{lccccccc} \text{maximize} & 30x1 & + & 60x2 & + & 50x3 & & \\ \text{subject to} & 3x1 & + & 4x2 & + & 2x3 & \leq & 60 \\ & x1 & + & 2x2 & + & 2x3 & \leq & 30 \\ & 2x1 & + & x2 & + & 2x3 & \leq & 40. \end{array}$$

Augmented system approach gives the optimal value `899.999999999` for each initial point. The normal system approach produce the same optimal value using (2.7) for the starting point, but it is unable to solve the problem using (2.6) for the starting point.

Example 3.4 In this example we show that the properties of the current problem can be crucial for choice of the approach. In the case of the problem *Degen2* even the initial matrix is ill-conditioned. The augmented system approach is unable to solve this problem. On the other hand, the normal system approach produce the optimal value `-1435.177999919273` with the precision 10^{-8}, in 12th iteration. In the case of the problem *Scfxm1* the iterative process based on the normal system approach is divergent, but the augmented approach produces the optimal value `18416.759034153343` with the precision 10^{-8}.

4 Conclusion

We compare two variants of Mehrotra's primal dual algorithm which are based on the augmented and normal equations system. As we see in Example 3.1 the augmented system approach is generally faster than the normal system. As the precision is similar in the both approaches, we conclude that augmented system approach has better performance in the most of cases. Also, we show that numerical performances of the augmented system approach can be improved by the appropriate selection of the staring point. It is worth of mention that this result depends on the specific methods for solving the augmented and normal system in the package MATHEMATICA. As it is reported in [1], [8], [12], the implementation of the augmented system approach in procedural languages has advantageous stability properties and ability of its easy extension to handling quadratic programming problems and free variables. But, the augmented system formulation has some disadvantages with respect to procedural languages [8], [12]:

- Algorithms and software for solving sparse symmetric indefinite systems are not as highly developed and widely available as sparse Cholesky codes.

- It takes more computer time (typically, 50% -100% more) to obtain the iterative step from the augmented system form than from the Cholesky algorithm.

The first objection is less significant, thanks to considerable recent work on software in this area [3], [6], [7]. The second objection - increased solution time - generally will probably remain, even if significant advances are made in software. In this paper we observed that the implementation of the augmented system in the package MATHEMATICA is even faster, especially for large scale problems. In any case, the advantages of the augmented system formulation give it an important role to play in future software for primal-dual methods. Note that *HOPDM* determines the implemented approach with respect to the properties of the current problem. This idea will be used in our future work.

References

[1] E.D. Andersen, J. Gondzio, C. Mészáros and X. Xu , *Implementation of interior point methods for large scale linear programming*, Technical report, HEC, Université de Genève (1996).

[2] M. Arioli, I.S. Duff and P.P.M., *On the augmented system approach to sparse least-squares problems*, Numer. Math. **55** (1989), 667–684.

[3] C. Ashcraft, R.L. Grimes, J.G. Lewis, *Accurate symmetric indefinite linear equation solvers*, SIAM Journal on Matrix Analysis **20** (1999), 513–561.

[4] M.A. Bhatti, *Practical optimization with MATHEMATICA applications*, Springer Verlag Telos (2000).

[5] J. Czyzyk, S. Mehrotra and S.J. Wright, *PCx User Guide*, Optimization Thechnology Center, Technical Report 96/01 (1996).

[6] I.S. Duff, *The solution of augmented systems*, in Numerical Analysis, D.F. Griffiths and G.A. Watson, eds., Longman Scientific and Technical, Essex,U.K. (1993), 40–45.

[7] R. Fourer, S. Mehrotra, *Solving symmetric indefinite systems in an interior-point method for linear programming*, Mathematical Programming bf 62 (1993), 15–39.

[8] J. Gondzio and T. Terlaky, *A computational view of interior-point methods for linear programming*, Cahiers de recherche, Technical Report 1994.22 (1994).

[9] N.K. Karmarkar, K.G. Ramakrishnan, *Computational result of an interior point algorithm for large scale linear programming*, Mathematical Programming **52** (1991), 555–586.

[10] V.V. Kovačević-Vujčić, and M.D. Ašić, *Stabilization of interior-point methods for linear programming*, Computational Optimization and Applications **14** (1999), 331–346.

[11] S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM J. Optim. **2** (1992), 575–601.

[12] S.J Wright, *Primal-dual interior-point methods*, SIAM, Philadelphia (1997).

Faculty of Science and Mathematics, Department of Mathematics
Višegradska 33, 18000 Niš
pecko@pmf.pmf.ni.ac.yu

Faculty of Economics
Trg Kralja Aleksandra 11, 18000 Niš
nebojsas@orion.eknfak.ni.ac.yu