FACTA UNIVERSITATIS
Series: **Mechanics, Automatic Control and Robotics** Vol.3, N° 11, 2001, pp. 219 - 222

INITIAL POINT IN PRIMAL-DUAL INTERIOR POINT METHOD

UDC 519.85 (045)

Nebojša V. Stojković[1], Predrag S. Stanimirović[2]

[1]Ekonomski fakultet, Niš, Trg VJ 11
[2]Prirodno matematički fakultet, Niš , Ćirila i Metodija 2

Abstract. *In this paper we propose two algorithms for the choise of the initial solution in a primal-dual algorithm. Several numerical examples are presented.*

Key words: *linear programming, primal-dual algorithm, initial solution*

1. INTRODUCTION

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

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

where $c, x \in R^n$, $b \in R^m$, and A is an $m \times n$ real matrix. The dual problem for (1.1) is

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

where $\lambda \in R^m$ and $s \in R^n$.

Interior points methods are much faster with respect to the simplex method. The most popular public domain implementations of primal-dual methods are *PCx* [2] and *HOPDM* [3]. But, there are problems which can not be solwed by means *PCx* and *HOPDM* [4],[5].

The main aim of the present paper is to provide an efficent initial solution in a primal-dual algorithm. For this purpose, we propose two algorithms. These algorithms are implemented in the package MATHEMATICA. Several numerical examples are given.

2. THE CHOISE OF STARTING POINT

The first difficulty arising in implementing the primal-dual algorithm is the choise of an initial solution. Points that are realtively close to the optimal solution, but are not well

Received September 20, 2000
Presented at 5th YUSNM Niš 2000, Nonlinear Sciences at the Threshold of the Third Millenium, October 2-5, 2000, Faculty of Mechanical Engineering University of Niš

centred, often lead to bad performance or numerical difficulties. Theorem 6.1 from [8] guarantees global convergence of primal-dual algorithm from any starting point (x^0,λ^0,s^0) with $(x^0,s^0) > 0$. Mehrotra [6] has proposed to solve a certain quadratic programming problem to obtain the initial solution. Heuristic for finding (x^0,λ^0,s^0) tarts by calculating (x^*,λ^*,s^*) as the solution of two least-squares problems

$$\min_x \| x \|^2 \text{ subject to } Ax = b,$$

$$\min_{(\lambda,s)} \| s \|^2 \text{ subject to } A^T\lambda + s = c\ .$$

That is, x^* and s^* are the vectors of least norm for which the residuals r_b and r_c are zero. We define the residuals for the two linear equations as

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

and $e \in R^n$, $e = (1,1,\ldots,1)$. The starting point is then defined as

$$(x^0,\lambda^0,s^0) = (x^* + \delta_x e, \lambda^*, s^* + \delta_s e),$$

where the scalars δ_x and δ_s are calculated such that $(x^0,s^0) > 0$. A solution of this problem can be computed at a cost comparable to a single interior point iteration [1].

We propose a simple algorithm and search an initial solution which will be well centered and satisfied at least one constraint if it is possible. Moreover, the proposed algorithms generate the initial solution about ten time faster than single interior pont iteration. These measures are obtained by means of the function *Timing* in MATHEMATICA [7].

Algorithm 1.

Step 1. Compute the quantity

$$pom = \max\{\frac{|b_i|}{|\sum_{j=1}^n a_{ij}|}, \quad \sum_{j=1}^n a_{ij} \neq 0.\}$$

Step 2. Generate starting point x and s whose all coordinates are equal to *pom,* and the starting point l with all coordinates equal to zero.

Corresponding MATHEMATICA routine is

```
inicp [a_,b_] :=
   Module [{ i, pom, ppom, maxpom=1, k,m},
   k =Last [ Dimensions [a] ];
   m=First [ Dimensions [a] ];
   For [ i=1, i<=m, i++,
      pom=N[Sum [a [[ i, j ]], {j, k} ], 20];
           If [pom!=0,
         ppom=N[Abs [b[[i]]/pom], 20];
         If [ppom > maxpom, maxpom =ppom];
] ] ;
Return [maxpom]
]
```

Then the starting points can be generated by means of the following code:

```
pi=inicp[a, b];
x=Table[pi,{k+q}];
s=x;
l=Table[0,{m}];
```

After the application of *Algorithm 1,* sometimes we get very small coordinates for the starting point, which causes numerical instability in the iterative process. For this reason, we propose the following improvement of this algorithm.

Algorithm 2.

Step 1. Compute the quantity

$$pom = \max\{\max\{\frac{|b_i|}{|\sum_{j=1}^{n} a_{ij}|},1\}, \quad \sum_{j=1}^{n} a_{ij} \neq 0.\}$$

Step 2. Generate starting points x, s and l as in Step 2 of *Algorithm 1.*

Corresponding routine in MATHEMATICA is as follows

```
inicp1[a_,b_] :=
   Module [{ i, pom, ppom, maxpom=1, k,m},
   k =Last [Dimensions [a]];
   m=First [Dimensions [a]];
   For [ i=1, i<=m, i++,
      pom=N[Sum [a [[ i, j]], {j, k}], 20];
           If [pom!=0,
         ppom=Max[N[Abs[b [[i]]/pom], 20],1];
         If [ppom > maxpom, maxpom =ppom];
   ] ] ;
   Return [maxpom]
   ]
```

Example 2.1. Maximize the objective function $2x1 + 0.5x2 + 4x3 + x4 + 3x5 + 5x6$ subject to constraints

$$\{20x1 + 6x2 + 32.3x3 - 6x4 + 24x5 + 60.5x6 <= 2,$$
$$32x1 - 6x2 - 32x3 - 4x4 + 48.3x5 + 160.6x6 <= 4,$$
$$20x1 + 7.5x2 + 100x3 + 5x4 + 90.7x5 + 50x6 <= 5,$$
$$24x1 + 15x2 + 72.7x3 + 12x4 + 54x5 + 120.6x6 <= 6\}.$$

Applying *Algorithm 1,* we get *pom* = 0.02004677581022386. The list of intermediate values for the duality measure $x \cdot s$ is:

$$\{0.0109818, 0.0285081, 0.0561806, 0.051085, \ 0.0437433, 0.138333, 1.85929, 0.561087,$$
$$0.52302, 0.0959923, 0.0038237, 0.0000384209, \ 3.84209 * 10^{-7}, 3.84209 * 10^{-9}, 3.84209 * 10^{-11}\}.$$

The result is obtained after 15 iterations:

Out [1]= {0.49999999987196,

{0.0950610037697902, 2.029240875683194 * 10^−12, 6.141145353976096 * 10^−13,

0.3098779924331999,1.836450143212205 * 10^−12,1.087047548266103 * 10^−12}}.

On the other hand, applying *Algorithm 2,* we obtain *pom* = 1and the result is derived in 9 iterations. The following intermediate values for the duality measure are generated:

$$\{2.46877, 0.797101, 0.213952, 0.0214338, \ 0.000318099, 3.18125 * 10^{-6}, 3.18125 * 10^{-8}$$
$$3.18125 * 10^{-10}, 3.18125 * 10^{-12}\}.$$

Example 2.2. Now we consider the problem which is known as *Afiro*. We get a large value *pom* = 500. for the initial coordinates of x and s But, this point is well centered, and the iterative process is very fast. This statemend is confirmed by the following values of the duality measure, generated during the iterations:

$$\{2,21823793226058985*10^{6},282483.,3941.31,$$
$$637.836,143.44,31.2389,1.48649,0.0149557,$$
$$0.000149557,1.4955671087526805*10^{-6}\}.$$

Let us mention that *Algorithm 2* requires 0.27 sec. for the generation of the starting point with respect to approximately 3 sec. required in a single iterative step. The code *PCx* requires 8 iteration to solve the problem.

Example 2.3. We also consider the problem known as *Blend.* The processor time for the initial solution is 1.92 sec., and a single iterative step requires approximately 14 sec. The result is generated in 11 iterations. On the other hand, *PCx* requires 10 iterations.

REFERENCES

1. E. D.Andersen, S.Meszaros, J.Gondzio and X. Xu, I*mplementation of interior point methods for large scale linear programming,* Hec-Geneve, Technical Report 96/3, 1996.
2. J. Czyzyk, S. Mehrotra and S. J. Wright, *PCx User Guide,* Optimization Thechonology Center, Technical Report 96/01, 1996.
3. J. Gondzio, *HOPDM (Version 2.12): A fast LP solver based on a primal-dual interior point method,* European Journal of Operation Research 85 (1995), 221-225.
4. V. Kovačević-Vujčić, *Techniacal Report 901-98,* Laboratory of Operations Research, Faculty of Organizational Sciences, November 1998.
5. V. Kovačević-Vujčić, *Some computational experience with the HOPDM code,* Proceedings of SYMOPIS 97 (1997), 383-385.
6. S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM J. on Optimization (1992), 575-601.
7. S. Wolfram, *Mathematica Book, Version 3.0,* Wolfram Media and Cambrige University Press, 1996.
8. S. J. Wright*, Primal-Dual Interior Point Method,* SIAM Philadelphia, 1997.

POČETNO REŠENJE U PRIMAL-DUAL METODI UNUTRAŠNJE TAČKE

Nebojša V. Stojković, Predrag S. Stanimirović

U radu su data dva algoritma za izbor početnog rešenja u primal-dual algoritmu. Navedeno je i nekoliko numeričkih primera.

Ključne reči: *linearno programiranje, primal-dual algoritam, početno rešenje*