INTERIOR POINT METHODS FOR MULTICOMMODITY NETWORK FLOWS

***Luis E. Torres Guardia*[1] *and Gilson Alves Lima*[1]**

May, 2010

Abstract

This article studies the linear multicommodity network flow problem. This kind of problem arises in a wide variety of contexts. A numerical implementation of the primal-dual interior-point method is designed to solve the problem. In the interior-point method, at each iteration, the corresponding linear system, expressed as a normal equations system, is solved by using the AINV algorithm combined with a preconditioned conjugate gradient algorithm or by the AINV algorithm for the whole normal equations. Numerical experiments are conducted for networks of different dimensions and numbers of products for the distribution problem. The computational results show the effectiveness of the interior-point method for this class of network problems.

2010 Mathematics Subject Classification: 90C05, 90C51, 90B06.

Keywords: *Linear programming, Interior-point methods, Network optimization, Multicommodity flows.*

1. *Universidade Federal Fluminense, RJ, Brasil.*

1 Introduction

This work seeks a numerical solution for the problem of optimizing a linear network problem for the multicommodity case where each single item (commodity) is governed by their own network flow constraints, but also shares common facilities, so the individual single commodity are not independent. This model is known as the multicommodity flow problem, in which the individual commodity shares common arcs in a capacitated network. This kind of problem does not satisfy the integrality property as the case of single - commodity network flow problems that they always have integer solutions whenever the supply/demand and capacity data are integer valued.

Multicommodity flow problems arise in a wide variety of application contexts, for example, in telecommunications applications, telephone calls between specific node pairs in an underlying telephone network. Some of these applications and others ones can be seen in [1], [24] and [29].

Network specializations based on the simplex method have been studied by many authors. Some of these works are the comprehensive survey found in [5], which includes decomposition, partitioning, compact inverse methods, and primal-dual algorithms. The paper [23] also presents a state-of-the-art survey of algorithms and some results for the mentioned problem. The paper [2] presents the computational experience for solving multicommodity network flow problems using specialized techniques that include a price-directive decomposition procedure, a resource-directive decomposition procedure using sub-gradient optimization, and a primal partitioning procedure, see also the paper [3]. Also, the simplex algorithm is described in [16] for special multicommodity network flow problems. A new solution approach for the above problem can be found in [18] based upon both primal partitioning and decomposition techniques, which simplifies the computations required by the simplex method. These ideas are also presented in [20]. The paper [30] presents a penalty-based algorithm that solves the multicommodity flow problem as a sequence of a finite number of scaling phases.

As it is known, the simplex method solves linear programming problems by visiting extreme points, on the boundary of the feasible set, each time improving the cost. In the mid 1980's new algorithms for linear programming were devised that find an optimal solution while moving in the interior of the feasible set, for this reason, they are generally called interior point methods. The field of these methods has its origins in the work described in [21]. This is the paper that introduced the first interior point algorithm with polynomial time complexity. In practice, the interior point methods are competitive with the simplex method, especially for large and sparse problem, they often outperform the simplex method. Details of these interior point methods can be seen in the books [33] and [34].

The most computationally expensive step of an interior point method is to find a solution of a linear system of equation, the so-called Newton equation system. All general purpose interior point method codes use a direct approach or iterative methods to solve the Newton equation system. There are two competitive direct approaches for solving the Newton equations: the augmented system approach and the normal equations approach. The former requires factorization of a symmetric indefinite matrix, the latter works with a smaller positive definite matrix.

The most efficient interior point method is the infeasible-primal-dual algorithm. The algorithm generates iterates which are positive, i.e. are interior with respect to the inequality constraints but do not necessarily satisfy the equality constraints. Other difficulty is the choice of a good initial solution.

Most implementations of primal-dual methods are based on the system of normal equations. They use direct Cholesky decomposition of the associated matrix. Iterative methods also could be used to solve the normal equations, but a good and computationally cheap preconditioned matrix could be chosen in order to accelerate the method to obtain the solution of the mentioned system. The papers [9], [10] and [11] for example, use a pre-conditioned conjugate gradient solver and a sparse Cholesky factorization, to solve the normal equations for multicommodity network flows. The paper [25] uses this procedure for block-structured linear programs, see also [15]. The other paper [13] presents various approaches to solve nonoriented multicommodity flow problems. It focuses on the specialization for the node-arc formulation of the problem and uses the dual affine scaling algorithm. This algorithm requires the Cholesky factorization of the respective matrix.

The present work applies a different method for solving the normal equations. Instead of the Cholesky method, this work uses a factorized sparse approximate inverse of the corresponding matrix, named AINV method, found in [7], with a combined conjugate gradient method. The AINV algorithm is a robust one, although taking a sometimes large computational time.

The remainder of the paper is organized as follows. Section 2 briefly describes the primal-dual interior-point method. Section 3 presents the mathematical formulation of the linear multicommodity network flow with capacitated arcs, and section 4 develops a specialization of the interior-point method for multicommodity problems, considering the normal equations approach and the AINV algorithm. Section 5 presents the computational results for networks of various dimensions and variable number of commodities for the distribution problem. Finally, concluding remarks are made in section 6.

2 The Primal-Dual Interior-Point Method

This section presents a brief description of the primal-dual interior-point method to solve the linear programming problem (LP) in the primal form. This problem is given by:

$$\begin{array}{rrcl} \text{Minimize} & c^T x & & \\ \text{subject to :} & Ex & = & b, \\ & x & \geq & 0, \end{array} \tag{1}$$

being $x \in R^q$ is the decision vector, $c \in R^q$, $b \in R^p$ and E is a matrix, $E \in R^{pxq}$, $p < q$ of full rank. The dual of the linear problem (1) has the form:

$$\begin{array}{rrcl} \text{maximize} & b^T y & & \\ \text{subject to :} & E^t y + z & = & c, \\ & z & \geq & 0, \end{array} \tag{2}$$

being $y \in R^p$ is the dual variables and $z \in R^q$ is the vector of dual slack variable.

The first order optimality conditions, also known as the Karush-Kuhn-Tucker (KKT) conditions, for the problems (1) and (2) are:

$$\begin{array}{rcll} Ex & = & b, & x \geq 0, \\ E^T y + z & = & c, & z \geq 0, \\ XZe & = & 0, & \end{array} \tag{3}$$

where X and Z are diagonal matrices defined as $X = diag(x_1, \ldots, x_q)$, $Z = diag(z_1, \ldots, z_q)$, and e is the q-vector of all ones, that is: $e = (1, \ldots, 1, \ldots, 1) \in R^q$.
To apply the primal-dual interior-point method to solve the LP problem, it is solved the following perturbed KKT conditions :

$$\begin{array}{rcll} Ex & = & b, & x \geq 0, \\ E^T y + z & = & c, & z \geq 0, \\ XZe & = & \mu e, & \end{array} \tag{4}$$

where $\mu > 0$ is called the barrier parameter. These modifications (4) are equivalent to the first order KKT conditions (3), except that the third condition is perturbed by μ .

Let us notice that if $\mu = 0$ and $x \geq 0$, $z \geq 0$, the KKT conditions (4) coincide with the KKT conditions (3). For this reason, the choice of the parameter μ plays an important role in the interior-point method. In the interior point method, at each iteration, the parameter $\mu > 0$ is reduced by a certain factor. As the sequence of barrier parameters μ converging to zero, the solution $(x(\mu), y(\mu), z(\mu))$ converges to an optimal solution of the LP problem. The system (4) is solved using Newton's method. Let $dw = (dx, dy, dw)^T$ denote the Newton's direction, obtained by the linearization of system (4) and determined by the solution of the system of linear equations:

$$\begin{pmatrix} E & 0 & 0 \\ 0 & E^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} \xi_b \\ \xi_c \\ \xi_\mu \end{pmatrix}, \tag{5}$$

where

$$\begin{aligned} \xi_b &= b - Ex, \\ \xi_c &= c - E^T y - z, \\ \xi_\mu &= \mu e - XZe \end{aligned}$$

If the third equation of the system (5) is eliminated, that is, $dz = X^{-1}(\xi_\mu - Zdx)$, it is obtained the following indefinite symmetric system, also called an augmented system:

$$\begin{pmatrix} -X^{-1}Z & E^T \\ E & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \xi_c - X^{-1}\xi_\mu \\ \xi_b \end{pmatrix}, \tag{6}$$

$$dz = \xi_c - E^T dy$$

and making a further substitution, if dx is eliminated from system (6), the following linear system, named normal equations, is obtained:

$$(EZ^{-1}XE^T)dy = EZ^{-1}X(\xi_c - X^{-1}\xi_\mu) + \xi_b \tag{7}$$

and the others variables dz and dx can be determined as following:

$$\begin{aligned} dz &= \xi_c - E^T dy \\ dx &= Z^{-1}(\xi_\mu - Xdz) \end{aligned} \tag{8}$$

To summarize an iteration of the infeasible primal-dual interior-point method, let at the j-th iteration, $dw_j = (dx_j, dy_j, dz_j)^T$ denote the solution obtained from the system (5). In the next iteration, a new interior point $w_{j+1} = (x_{j+1}, y_{j+1}, z_{j+1})^T$ is determined using the following rules:

$$\begin{aligned} x_{j+1} &= x_j + \beta\alpha_j dx_j, \\ y_{j+1} &= y_j + \beta\alpha_j dy_j, \\ z_{j+1} &= z_j + \beta\alpha_j dz_j, \end{aligned}$$

α_j being the step length, determined by a suitable line search procedure and $\beta \in (0,1)$ and near 1.

With this new point w_{j+1}, the barrier parameter μ is updated according to certain rules and a new linear system (5) is formed and solved by any solution method and the iterative procedure follows until a stopping rule is satisfied. Implementation of this interior point method can be found in [4].

3 Problem Formulation of the Multicommodity Network Flow

Let us consider a directed graph $G = (N, E)$, with N the set of nodes and E the set of edges. The graph represents a network where K different commodities are sent from given origins to given destinations represented by initial and terminal nodes. Let b_k denote the supply/demand for each commodity k. The coordinate b_{ki} denotes, if positive, the supply of commodity k in node i and, if negative, the demand for commodity k in node i. The flow through the network arcs, represented by the graph edges, is capacitated if there is a maximum b_{mc} for the sum of the mass of all commodities passing by each arc.

The linear formulation for this network flow problem, in the format node-edge, is as follows:

$$\text{minimize} \sum_{k=1}^{K} c_k^T x_k \tag{9}$$

$$\text{subject to :} \quad Ax_k = b_k, \quad k = 1, \ldots, K \tag{10}$$

$$\sum_{k=1}^{K} x_k + x_\nu = b_{mc} \tag{11}$$

$$x_\nu \geq 0, x_k \geq 0, k = 1, \ldots, K. \tag{12}$$

Here, $A \in R^{mxn}$ is the node-edge incidence matrix of graph G. For each $a \in E$, the coordinate x_{ka} of vector x_k denotes the flow of commodity k through the network arc a and $x_\nu \in R^n$ is the vector of slack variables, and $c_k \in R^n$ is the cost for each commodity. It will be assumed that the incidence matrix A is full rank. Otherwise, rows can be removed.

The Equation (9) presents the objective function to be optimized. The Equation (10) establishes the flow conservation constraint and (11) is known as the capacity constraint and establishes the maximum total flow of all commodities at each arc. The Equation (12) expresses the fact that the flows must be nonnegative. The case for the nonlinear model was studied in [32].

4 The Multicommodity Network Flow and the Primal-dual Method

In this section, the primal-dual interior-point method presented in section 2 is applied to the multicommodity network flow problem defined by (9)-(12). The process starts by building matrices X and Z for the case of multiple commodities. These are block diagonal matrices: The matrix X is diagonal given by:

$$X = \begin{bmatrix} X_1 & & & \\ & \cdot & & \\ & & X_K & \\ & & & X_\nu \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & & & \\ & \cdot & & \\ & & Z_K & \\ & & & Z_\nu \end{bmatrix}.$$

Each sub-matrix $X_k, k = 1, \ldots, K$ is a diagonal matrix with components $x_{ki}, i = 1, \ldots, n$, for each commodity k, and X_ν is a diagonal matrix with components given by the slack variable x_ν. The matrix Z has the same structure.

On the other side, the matrix of constraints of the multicommodity network problem may be visualized as:

$$E = \begin{bmatrix} A & & \\ & \cdot & \\ & & A \\ I & I & . & I \end{bmatrix},$$

with each block matrix A corresponding to the node-edge matrix of incidence. The identity matrices in the last line correspond to the capacity constraints (11).

Finally, let us build a block diagonal matrix D as follows:

$$D = \begin{bmatrix} D_1 & & & \\ & \cdot & & \\ & & D_K & \\ & . & & D_\nu \end{bmatrix},$$

with each sub-matrix $D_k, k = 1, \ldots, K$, being a diagonal matrix given by $D_k = Z_k^{-1} X_k$, and with $D_\nu = Z_\nu^{-1} X_\nu$ a diagonal matrix related to the slack variables.

The hardest computational effort required by the primal-dual method consists of solving the linear system (7). The next steps of this work will involve describing different methods to solve these normal equations.

To solve the linear system (7), it is performed block multiplications to determine the matrix (EDE^T), which has the following structure:

$$\mathbf{EDE^T} = \left[\begin{array}{cccc|c} AD_1A^T & 0 & . & 0 & AD_1 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & AD_KA^T & AD_K \\ \hline D_1A^T & . & . & D_KA^T & D_\nu + \sum_{k=1}^K D_k \end{array}\right].$$

Let B denote the block diagonal matrix of blocks $B_k = AD_kA^T, k = 1,\ldots,K$, and C denote the matrix given by $C^T = [D_1A^T,\ldots,D_KA^T]$ and F the matrix given by $F = D_\nu + \sum_{k=1}^K D_k$. The matrix F is diagonal since D_ν and D_k, $k = 1,\ldots,K$, are diagonal matrices.
From the previous notation, it follows:

$$\mathbf{EDE^T} = \begin{bmatrix} B & C \\ C^T & F \end{bmatrix}.$$

Thus, the system (7) may be written as:

$$\begin{bmatrix} B & C \\ C^T & F \end{bmatrix} \begin{pmatrix} dy_1 \\ dy_2 \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix},$$

where $dy = (dy_1, dy_2)^T$ and $h = (h_1, h_2)^T = ED(\xi_c - X^{-1}\xi_\mu) + \xi_b$.
The above linear system may be written as follows:

$$[F - C^TB^{-1}C]dy_2 = h_2 - C^TB^{-1}h_1 \tag{13}$$

$$Bdy_1 = h_1 - Cdy_2 \tag{14}$$

where the matrix $(F - C^TB^{-1}C)$ is known as Schur complement.
Let $h_1 = [h_{11},\ldots,h_{1K}]^T$. From Equation (13), it follows:

$$[F - C^TB^{-1}C]dy_2 = h_2 - \sum_{k=1}^{K}(D_kA^TB_k^{-1})h_{1k}, \tag{15}$$

where

$$F - C^T B^{-1} C = D_\nu + \sum_{k=1}^{K} D_k - \sum_{k=1}^{K} (D_k A^T B_k^{-1} A D_k).$$

To use this result it must be computed the inverse of matrix $B_k = AD_kA^T$, $k = 1, \dots, K$. This can be determined using, for instance, Cholesky decomposition. In this work, approximation of matrices B_k^{-1}, $k = 1, \dots, K$, will be used by a version of AINV method (see [7]), in such a way that:

$$B_k^{-1} \cong Z_k P_k^{-1} Z_k^T, \quad k = 1, \dots, K,$$

where Z_k an upper triangular matrix with diagonal of $1's$ and P_k is a diagonal matrix.

The AINV algorithm given in [7] to build an inverse of the matrix $B_k = AD_kA^T$ is presented below. In this development, e_i denotes the $i - th$ axis unitary vector and, to simplify notation, the sub-index k is omitted.

1. Let $z_i^{(0)} = e_i$, $1 \leq i \leq m$
2. for $i = 1, 2, \dots, m$ do
3. $v_i = (ADA^T) z_i^{(i-1)}$
4. for $j = i, i+1, \dots, m$ *do*
5. $p_j^{i-1)} = v_i^t z_j^{(i-1)}$
6. end
7. if $i = m$ end;
8. for $j = i+1, \dots, m$ do
9. $z_j^{(i)} = z_j^{(i-1)} - \left(\frac{p_j^{(i-1)}}{p_i^{(i-1)}} \right) z_i^{(i-1)}$

10. end

11. End

Let $z_i = z_i^{(i-1)}$ and $p_i = p_i^{(i-1)}$, for $1 \leq i \leq m$. Return.

$$Z = [z_1, z_2, \ldots, z_m] \quad and \quad P = diag(p_1, p_2, \ldots, p_m).$$

In step (3) of the AINV algorithm above, it can be seen that the procedure performs the matrix-vector multiplication $(ADA^T)z_i^{(i-1)}$. For our network problem it is not needed to store neither the matrix A nor the matrix ADA^T, only the results of the multiplication of A or A^T by vectors are needed. Let us recall that the matrix D is a diagonal matrix, so that from its structure it is possible to apply the method to large-scale problems.

Thus, the system (15) may be written as:

$$[F - C^T B^{-1} C] dy_2 = h_2 - \sum_{k=1}^{K} (D_k A^T Z_k P_k^{-1} Z_k^T) h_{1k}.$$

This last system may be solved by many methods. Among these, the preconditioned conjugate gradient algorithm can be applied in such a way as to reduce the number of iterations needed to obtain the approximate solution of the respective system. Details of this algorithm may be obtained, for instance, in [26], [27], [28]. Now, to obtain a solution for the above linear system, it may be used as conditioning matrix simply the diagonal matrix F.

Once dy_2 is determined, it is possible to determine dy_1 by using the system (14). In fact, from $Bdy_1 = h_1 - Cdy_2$, it follows:

$$dy_1 = B^{-1}(h_1 - Cdy_2),$$

and using the fact that the matrix B is block diagonal and the AINV decomposition, and from the construction of C, it follows:

$$dy_{1k} = (Z_k P_k^{-1} Z_k^T)[h_{1k} - (AD_k) dy_2], \quad k = 1, \ldots, K,$$

being $dy_1 = (dy_{11}, \ldots, dy_{1K})^T$.

By using the above procedure, o vector $dy = (dy_1, dy_2)^T$ is determined. Then it follows to compute the remaining variables dz and dx by employing the relations (8), and a new interior point $w = (x, y, z)^T$ is determined.

A second way to solve (15) consists of, instead of applying the conjugate gradient algorithm to determine dy_2, applying, once more, the AINV algorithm to compute the inverse of the matrix $[F - C^T B^{-1} C]$ by:

$$[F - C^T B^{-1} C]^T \cong Z P^{-1} Z^T,$$

where, again, Z is an upper triangular matrix of $1'$s in the diagonal and P is a diagonal matrix.

By this way, dy_2 may be determined using again the AINV algorithm, to determine the inverse of the matrix B_k by $B_k^{-1} = Z_k P_k^{-1} Z_k^T, k = 1, \ldots, K$, and again to determine the inverse of the matrix $[F - C^T B^{-1} C]$. Thus, from (15), it follows:

$$dy_2 = (Z P^{-1} Z^T)(h_2 - \sum_{k=1}^{K} (D_k A^T Z_k P_k^{-1} Z_k^t) h_{1k}).$$

By this last approach, the vector dy_2 is computed. After that, using (14), dy_1 is obtained and dx and dz are determined as above.

A third way of solving the linear system (6) consists of reducing to the following indefinite symmetric system, also called augmented system of equations:

$$\begin{bmatrix} -X^{-1}Z & E^T \\ E & 0 \end{bmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \xi_c - X^{-1}\xi_\mu \\ \xi_b \end{pmatrix}. \tag{16}$$

There are several methods to solve the indefinite symmetric system. They may be found in [6], [8], [12], [17], [19], [22] among others. In the present work, as it say before, it is not used this indefinite symmetric system to solve the corresponding Newton system. It could be done in any other paper.

5 Computational Results

This section presents the results of numerical computations performed to solve the linear multicommodity network flow problem, of different dimensions, for variable number of commodities. The experiments were performed using the primal-dual interior-point method and using the normal equations approach.

As an application, the network considers the problem of distributing goods from one or more plants through a set of warehouses which serves customer demands, see [14]. This work considers the demands are deterministic. This distribution problem was studied in [31] for one commodity, using the homogeneous and self-dual linear programming algorithm, implemented in MATLAB code. This work treats the distribution model for multicommodity case. The network is extended to generate large-scale networks. To do that, a specific FORTRAN program was implemented to determine the dimension of the new network, that is, the number of arcs and nodes and the initial and final nodes that define each arc in the network. All computational tests were conducted in an AMD Athlon PC with a Windows XP platform, of 1.0 GB of RAM and 2.4 GHz of frequency. The computational codes were entirely written in FORTRAN. Double precision was used in all computations.

A starting point is given in the primal-dual method and it may not be viable. The step length α is determined by a line search procedure, see [4]. It was used in all computational tests the value $\beta = 0.99995$.

Sometimes, especially for large-scale systems, it was not required great precision. By its turn, the stopping rule for the interior-point method is determined in terms of proximity of the values of the objective function in successive steps. It is required an absolute difference between these values of the objective function smaller or equal to 10^{-8} and that the value of the parameter μ be close to zero. Also, it is required the proximity of the values of the objectives functions of the linear problem and its corresponding dual problem.

Several tests are performed for a different number of commodities varying from 1000 to 1500. As each variable x_k, for $k = 1, \ldots, K$, has dimension equal to the number of arcs, the total dimension of the decision variables $x = (x_k)$, $k = 1, \ldots, K$ is equal to the number of arcs multiplied by the number of commodities.

Tables 1 and 2 display the results for some distribution problems, obtained from the interior point method, using the corresponding normal equations. These equations are solved using the preconditioned conjugate gradient algorithm combined with the AINV algorithm or using the AINV algorithm alone. These tables show the computational time in seconds, the number of the interior point method iterations required to converge to an optimal solution, the optimal objectives values of the primal and dual problems, the total number of variables, without considering the dimension of the slack vector, as well as the number of links and nodes. The value of the barrier parameter μ obtained, in all the cases studied, is less than 10^{-12}.

These tables also show others variables as follows:

- m: representing number of plants;
- n: representing number of warehouses;
- p: representing number of customers;
- h: representing number os stages.

It can be seen from the tables 1 and 2 that, the objective values of the primal and dual problems are very close, but a little more precision is obtained by using the AINV algorithm with a large computational time.

Table 1:Computational results for multicommodity flow problem
Preconditioned conjugate gradient algorithm

number commodities	1000	1200	1500
m	4	5	4
n	6	7	6
p	8	9	8
h	12	6	6
links	1057	759	565
nodes	234	147	126
number variables	1057000	910800	847500
value of primal	11267999.99999997	10139999.99999575	8117999.99996751
value of dual	11268000.14211712	10140005.02266686	8118002.65844308
iterations	13	12	13
time(seconds)	1561.156	416.437	352.844

Table 2: Computational results for multicommodity flow problem
AINV algorithm

number commodities	1000	1200	1500
m	4	5	4
n	6	7	6
p	8	9	8
h	12	6	6
links	1057	759	565
nodes	234	147	126
number variables	1057000	910800	847500
value of primal	11268000.00000625	10139999.99999627	8117999.99999757
value of dual	11268000.00001867	10140000.00016499	8118000.00001930
iterations	13	12	13
time(seconds)	10562.703	3629.593	2482.969

6 Conclusions

This article discussed the application of the primal-dual interior-point method to solve multicommodity network flow problems for the distribution model. Some alternatives algorithms were developed to deal with the systems of linear equations involved in the flow optimization. The corresponding linear problems, expressed as normal equations, is solved by using the AINV algorithm for the whole system or combined with the preconditioned conjugate gradient algorithm. In these cases, the algorithm explores the structure of the linear constraints in such a way as to avoid storing the constraints matrices as well as products of the matrices that appear in the iterative procedures applied allows dealing efficiently with large-scale problems. The experimental results, to solve the dynamic distribution model, have demonstrated the efficiency of the method.

References

[1] Ahuja, A., Magnanti, T. and Orlin, J., (1993). *Network Flows: Theory, Algorithms and Applications.* USA: Prentice-Hall, Inc. New Jersey.

[2] Ali, A., Helgason, R., Kennington, J. and Lall, H., (1980). *Computational comparison among three multicommodity network flow algorithms.* Operations Research, 28, 995 - 1000.

[3] Ali, A., Barnett, D., Farhangian, K., Kennington, J., Patty, B., Shetty, B., McCarl, B. and Wong, P., (1984). *Multicommodity network problems: Applications and computations.* IIE Transactions, 16, 127 - 134.

[4] Andersen, E., Gondzio, J., Mészáros, C. and Xu, X., (1996). *Implementation of interior - point methods for large scale linear programs.*

In Interior Point Methods of Mathematical Programming, Terlaky, T., ed. Kluwer Academic Publishers, Netherlands, 189-252, 1996.

[5] Assad, A., (1978). *Multicommodity network flows: A survey.* Networks, 8, 37 - 91.

[6] Bai, Z. and Wang, Z., (2006). *Restrictive preconditioners for conjugate gradient methods for symmetric positive definite linear systems.* Journal of Computational and Applied Mathematics, 187, 202-226.

[7] Benzi, M., Cullum, J. and Tuma, M., (2000). *Robust approximate inverse preconditioning for the conjugate gradient method.* SIAM Journal on Scientific Computing, 22, 4, 1318 - 1332.

[8] Bergamaschi, L., Gondzio, J. and Zilli, G., (2004). *Preconditioning indefinite systems in interior point methods for optimization.* Computational Optimization and Applications, 28, 149 - 171.

[9] Castro, J., (2000). *A specialized interior point algorithm for multicommodity network flows.* SIAM Journal on Optimization, 10, 3, 852 - 877.

[10] Castro, J., (2003). *Solving difficult multicommodity problems with a specialized interior-point algorithm.* Annals of Operations Research, 124, 35 - 48.

[11] Castro, J., (2007). *An interior-point approach for primal block-angular problems.* Computational Optimization and Applications, 36, 195 - 219.

[12] Chai, J. and Toh, K., (2007). *Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming.* Computational Optimization and Applications, 36, 221-247.

[13] Chardaire, P. and Lisser, A., (2002). *Simplex and interior point specialized algorithms for solving nonoriented multicommodity flow problems.* Operations Research, 50, 2, 260 - 276.

[14] Cheung, R. and Powell, W., (1996). *Models and algorithms for distribution problems with uncertain demands.* Transportation Science, 30, 43- 59.

[15] Choi, I. and Goldfarb, D., (1990). *Solving multicommodity network flow problems by an interior point method.* In Large-Scale Numerical Optimization, T. Coleman e Y. Li, Eds, SIAM, Philadelphia, PA., 58-69, 1990.

[16] Detlefsen, N. and Wallace, S., (2002). *The simplex algorithm for multicommodity networks.* Networks, 39, 1, 15-28.

[17] Durazzi, C. and Ruggiero, V., (2003). *Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems.* Numerical Linear Algebra with Applications, 10, 673- 688.

[18] Farvolden, J., Powell, W. and Lustig, I., (1993). *A primal partitioning solution for the arc-chain formulation of a multicommodity network flow problem.* Operations Research, 41, 669 - 693.

[19] Golub, G., Wu, X. and Yuan, J., (2001). *SOR-like methods for augmented systems.* BIT, 41, 1, 71-85.

[20] Jones, K., Lustig, I., Farvolden, J. and Powell, W., (1993). *Network flows: The impact of formulation on decomposition.* Mathematical Programming, 62, 95 - 117.

[21] Karmarkar, N., (1984). *A new polynomial time algorithm for linear programming.* Combinatorica, 4, 373 - 395.

[22] Keller, C., Gould, N. and Wathen, A., (2000). *Constraint preconditioning for indefinite linear systems.* SIAM Journal Matrix Analysis and Applications, 21, 4, 1300 - 1317.

[23] Kennington, J., (1978). *A survey of linear cost multicommodity network flows.* Operations Research, 26, 209 - 236.

[24] Kennington, J. and Helgason, R., (1980). *Algorithms for Network Programming.* John Wiley and Sons, Inc., New York.

[25] Lustig, I. and Li, G., (1992). *An implementation of a parallel primal-dual interior-point method for block - structured linear programs.* Computational Optimization and Applications, 1, 141 - 161.

[26] Nocedal, J. and Wright, S., (1999). *Numerical Optimization.* Springer.

[27] Portugal, L., Resende, M., Veiga, G. and Júdice, J., (2000). *A truncated primal-infeasible dual-feasible network interior point method.* Networks, 35, 2, 91 - 108.

[28] Portugal, L., Resende, M., Veiga, G., Patrício, J. and Júdice, J., (2008). *FORTRAN subroutines for network flow optimization using an interior point algorithm.* Pesquisa Operacional, 28, 2, 243 - 261.

[29] Resende, M. and Pardalos, P., (2006). *Handbook of optimization in telecommunications*, Springer.

[30] Schneur, R. and Orlin, J., (1998). *A scaling algorithm for multicommodity flow problems.* Operations Research, 46, 2, 231 - 246.

[31] Torres, L., (2009). *El método homogéneo y auto-dual para el problema dinâmico lineal de distribución/inventario - Ambiente MATLAB.* Pro Mathematica, 23/45-46, 175 - 192.

[32] Torres, L. and Sant'Anna, A., (2009). *A separable nonlinear model for the multicommodity flow problem.* Optimization Methods and Software, 24, 6, 973-987.

[33] Vanderbei, R., (2008). *Linear Programming.* 3rd. Ed. Springer, Boston.

[34] Wright, S., (1997). *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia.

Resumen

Este artículo trata el problema de flujo en red para múltiplos productos. Ese tipo de problema aparece en una variedad de aplicaciones. La implementación numérica del método de puntos interiores primal - dual es realizada para resolver ese problema. En cada iteración del método de puntos interiores, el correspondiente sistema linear, formulado como un sistema de ecuaciones normales, es resuelto usando el algoritmo de factorización AINV combinado con el algoritmo pre condicionado de la gradiente conjugada o usando ese algoritmo de decomposición AINV para el sistema normal total. Resultados numéricos son realizados para redes de diferentes dimensiones y números de productos para el problema de distribución. Resultados computacionales muestran la eficiencia del método de puntos interiores para esta clase de problemas de red.

Palabras Clave: Programación lineal, Métodos de puntos interiores, Optimización en red, Flujo para múltiplos productos.

Luis E. Torres Guardia
Universidade Federal Fluminense, RJ, Brasil.
`tepletg@vm.uff.br`

Gilson Alves Lima
Universidade Federal Fluminense, RJ, Brasil.
`gilson@latec.uff.br`