数理解析研究所講究録
第 835 巻 1993 年 181-193

Primal-Dual Proximal Point Algorithm for Multicommodity Network Flow Problems

茨木 智（京大工）福島雅夫（京大工）茨木俊秀（京大工）

1. Introduction

The multicommodity network flow problem is an important class of network optimization problems, in which arcs are shared by several commodities and the flow of each commodity must be conserved at every node. The applications can be found in such diverse areas as data communication systems, transportation systems of crops, goods or vehicles and production lines of resources and products.

Let $G = (\mathcal{N}, \mathcal{A})$ be a directed graph, where $\mathcal{N} = \{1, 2, \ldots, m\}$ is the node set and $\mathcal{A} = \{1, 2, \ldots, n\}$ is the arc set. We consider a multicommodity network flow problem on G having K distinct commodities. To formulate the problem, we introduce some notations.

$x_{kj} \in R$: the flow of commodity k on arc j,

$x_k = (x_{k1}, x_{k2}, \ldots, x_{kn})^T \in R^n$: the vector of flows of commodity k,

$x = (x_1, x_2, \ldots, x_n) \in R^{nK}$: the vector of flows for all commodities,

$y_j \in R$: the total flow on arc j, i.e. $y_j = \sum_{k=1}^{K} x_{kj}$,

$y = (y_1, y_2, \ldots, y_n)^T \in R^n$: the vector of total flow,

$f_k : R^n \to (-\infty, +\infty]$: the cost function associated with flow of commodity k,

$g : R^n \to (-\infty, +\infty]$: the cost function associated with total flow,

$E \in R^{m \times n}$: the node-arc incidence matrix of graph G,

$b_{ki} \in R$: the requirement for commodity k at node i,

$b_k = (b_{k1}, b_{k2}, \ldots, b_{km})^T \in R^m$: the vector of requirements for commodity k.

Node i is called a supply point for commodity k if $b_{ki} > 0$, a demand point for commodity k if $b_{ki} < 0$, and a transshipment point for commodity k if $b_{ki} = 0$. We assume that, for each commodity k, the total demand equals the total supply, i.e. $\sum_{i=1}^{m} b_{ki} = 0$. We also assume that the cost functions f_k and g are closed proper convex.

Now we formulate the multicommodity network flow problem as follows:

$$\begin{aligned} \text{P: minimize} \quad & \sum_{k=1}^{K} f_k(x_k) + g(y) \\ \text{subject to} \quad & Ex_k = b_k, \ \ k = 1, 2, \ldots, K, && (1.1) \\ & y = \sum_{k=1}^{K} x_k. && (1.2) \end{aligned}$$

Constraints (1.1) are the flow conservation equations for individual commodities, whereas (1.2) are coupling constraints that link together the flows of all commodities. In particular, the latter constraints hamper straightforward decomposition of problem P into single commodity subproblems.

It is important to note also that problem P explicitly contains equality constraints only. Inequality constraints such as arc capacity constraints may be regarded as a part of the cost functions f_k and g, as shown by the following two important examples of multicommodity network flow problems. In these examples, we assume that the functions f_k and g are separable, i.e.

$$f_k(x_k) = \sum_{j=1}^{n} f_{kj}(x_{kj}), \quad \text{for all } k = 1, 2, \ldots, K \tag{1.3}$$

and

$$g(y) = \sum_{j=1}^{n} g_j(y_j). \tag{1.4}$$

This situation is often seen in practical applications and belongs to the important class of problems. In particular, when problem P has this assumption, the proposed algorithmcan be applied effectively. We consider the case that the cost function of P is separable in Section 3.

Example 1 (Linear multicommodity network flow problem [1, 2, 18, 19, 23]):

Let

$$f_{kj}(x_{kj}) = \begin{cases} a_{kj}x_{kj} & \text{if } 0 \le x_{kj} \le c_{kj}, \\ +\infty & \text{otherwise,} \end{cases}$$

for all $k = 1, 2, \ldots, K, j = 1, 2, \ldots, n$, and

$$g_j(y_j) = \begin{cases} 0 & \text{if } 0 \le y_j \le d_j, \\ +\infty & \text{otherwise,} \end{cases}$$

for all $j = 1, 2, \ldots, n$. Then problem P is rewritten as

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{K}\sum_{j=1}^{n} a_{kj}x_{kj} \\ \text{subject to} & Ex_k = b_k, \;\; k = 1, 2, \ldots, K, \\ & \sum_{k=1}^{K} x_{kj} \le d_j, \;\; j = 1, 2, \ldots, n, \\ & 0 \le x_{kj} \le c_{kj}, \;\; k = 1, 2, \ldots, K, \;\; j = 1, 2, \ldots, n. \end{array}$$

In this problem, c_{kj} is the capacity for the flow of commodity k on arc j, while d_j is the capacity for the total flow on arc j.

Example 2 (Traffic assignment problem [10, 11, 14]):

Let

$$f_{kj}(x_{kj}) = \begin{cases} 0 & \text{if } 0 \le x_{kj} \le c_{kj}, \\ +\infty & \text{otherwise,} \end{cases}$$

for all $k = 1, 2, \ldots, K, j = 1, 2, \ldots, n$. Let the functions g_j be given by

$$g_j(y_j) = \int_0^{y_j} \tilde{g}_j(t)dt,$$

where $\tilde{g}_j : R \to [0, +\infty)$ is an arc travel cost function which is nonnegative, increasing and convex. Then the traffic assignment problem may be formulated as

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} g_j(y_j) \\ \text{subject to} & Ex_k = b_k, \;\; k = 1, 2, \ldots, K, \\ & y_j = \sum_{k=1}^{K} x_{kj}, \;\; j = 1, 2, \ldots, n, \\ & 0 \le x_{kj} \le c_{kj}, \;\; k = 1, 2, \ldots, K, \;\; j = 1, 2, \ldots, n. \end{array}$$

In this problem, it is often assumed that $c_{kj} = +\infty$ for all k and j, in which case the Kuhn-Tucker conditions for the problem represent the well-known user optimal principle in a congested traffic network [9].

There are a large number of references on the nonlinear multicommodity network flow problems, among others, the traffic assignment problem. For example, linear approximation methods [5, 8], Frank-Wolfe method [10, 22] and gradient projection method [3] belong to the class of algorithms which directly exploit the advantage of the network structure. On the other hand, algorithms based on the dual approach have also been studied extensively in conjunction with various optimization techniques, e.g. a subgradient method [11], descent methods [12, 14] and relaxation methods [4, 24, 30]. Note that all the above mentioned methods except [14] are concerned with the Lagrangian dual problem obtained by relaxing both coupling constraints and flow conservation equations, while the method of [14] utilizes another Lagrangian dual which is defined by relaxing coupling constraints only.

The purpose of this paper is to present a primal-dual proximal point algorithm for the convex multicommodity network flow problem P. The proximal point algorithm and its variants have been extensively studied in the literature [6, 7, 16, 26, 27, 29]. In particular, the primal-dual proximal point algorithm is considered in [16, 26, 29]. The algorithm proposed in this paper is closely related the one presented by the authors in [16], which is tailored to solve linearly constrained convex programming problems. The method of [16], however, is concerned with the ordinary Lagrangian dual problem obtained by relaxing all linear constraints, while the method to be proposed in this paper deals with the Lagrangian function formed by relaxing the coupling constraints (1.2) only. Note that the dual optimality is attained when the primal feasibility is satisfied, i.e. the relaxed constraints of the primal problem are satisfied. Note also that since we cannot proceed the iteration of the algorithm infinitely, a solution obtained in practice is usually an approximate one that satisfies a certain termination criterion. Concerning this point, the proposed method has a remarkable feature. Namely, an approximate solution obtained by the proposed method, which may not satisfy the coupling constraints (1.2), necessarily satisfies the flow conservation equations (1.1) for all commodities. This property turns out to be very useful in practical applications.

2. Primal-Dual Proximal Point Algorithm

Let $\hat{f}_k : R^n \to (-\infty, +\infty]$ be the function defined by

$$\hat{f}_k(x_k) = \begin{cases} f_k(x_k) & \text{if } Ex_k = b_k, \\ +\infty & \text{otherwise.} \end{cases} \tag{2.1}$$

Then problem P is reformulated as

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{K} \hat{f}_k(x_k) + g(y) \\ \text{subject to} & y = \sum_{k=1}^{K} x_k. \end{array}$$

For this problem, let p denote the vector of Lagrange multipliers and define the Lagrangian L : $R^{nK} \times R^n \times R^n \to (-\infty, +\infty]$ by

$$L(x, y, p) = \sum_{k=1}^{K} \hat{f}_k(x_k) + g(y) - \langle p, \sum_{k=1}^{K} x_k - y \rangle, \tag{2.2}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

Now we may define the dual of problem P as follows:

D: maximize $\psi(p)$,

where the dual objective function ψ is given by

$$\psi(p) = \inf_{x \in R^{nK}, y \in R^n} L(x, y, p).$$

For $(\bar{x}, \bar{y})$ to solve P and $\bar{p}$ to solve D, it is necessary and sufficient that the following Kuhn-Tucker conditions hold:

$$\bar{p} \in \partial \hat{f}_k(\bar{x}_k), \quad k = 1, 2, \ldots, K,$$

$$-\bar{p} \in \partial g(\bar{y}),$$

$$\bar{y} = \sum_{k=1}^{K} \bar{x}_k,$$

where $\partial \hat{f}_k$ and ∂g denote the subdifferential operators of $\hat{f}_k$ and g, respectively. These conditions imply that $(\bar{x}, \bar{y}, \bar{p})$ is a saddle point of the Lagrangian L. When problem P has an optimal solution $(\bar{x}, \bar{y})$, the existence of a multiplier vector $\bar{p}$ satisfying the above Kuhn-Tucker conditions is guaranteed, provided that P is strongly consistent, i.e., there is at least one feasible solution in the relative interior of the effective domain of the objective function [25, p. 300].

The primal-dual proximal point algorithm [16, 26, 29] is an iterative method that generates a sequence of points converging to a Kuhn-Tucker point of the problem. Each iteration consists of finding an (approximate) saddle point of a convex-concave function, which is obtained by augmenting the Lagrangian by quadratic terms of both primal and dual variables. For problem P, the augmented Lagrangian $L^{(\mu)} : R^{nK} \times R^n \times R^n \to (-\infty, +\infty]$ at the μ-th iteration is given by

$$L^{(\mu)}(x,y,p) = L(x,y,p) + \frac{1}{2\gamma^{(\mu)}}|x - x^{(\mu)}|^2 + \frac{1}{2\gamma^{(\mu)}}|y - y^{(\mu)}|^2 - \frac{1}{2\gamma^{(\mu)}}|p - p^{(\mu)}|^2, \tag{2.3}$$

where $\gamma^{(\mu)}$ is a positive constant and $|\cdot|$ denotes the Euclidean norm. Note that $L^{(\mu)}$ is strongly convex and strongly concave with modulus $\frac{1}{\gamma^{(\mu)}}$ in (x,y) and in p, respectively.

There are two strategies to find an approximate saddle point of $L^{(\mu)}$: One is that we first maximize $L^{(\mu)}$ in p and then approximately minimize the resulting function of (x,y) [26, 29], i.e.

$$(x^{(\mu+1)}, y^{(\mu+1)}, p^{(\mu+1)}) \approx \arg \min_{x \in R^{nK}, y \in R^n} \left\{ \max_{p \in R^n} L^{(\mu)}(x,y,p) \right\}, \tag{2.4}$$

and the other we first minimize $L^{(\mu)}$ in (x,y) and then approximately maximize the resulting function of p [16], i.e.

$$(x^{(\mu+1)}, y^{(\mu+1)}, p^{(\mu+1)}) \approx \arg \max_{p \in R^n} \left\{ \min_{x \in R^{nK}, y \in R^n} L^{(\mu)}(x,y,p) \right\}. \tag{2.5}$$

In the proposed algorithm, we adopt the latter strategy to find an approximate saddle point of $L^{(\mu)}$. The difference between the two strategies will be clarified in the following.

In order to explain (2.5) in more detail, we define

$$(x(p), y(p)) = \arg \min_{x \in R^{nK}, y \in R^n} L^{(\mu)}(x, y, p) \tag{2.6}$$

and

$$\begin{aligned} \psi^{(\mu)}(p) &= L^{(\mu)}(x(p), y(p), p) \\ &= \min_{x \in R^{nK}, y \in R^n} L^{(\mu)}(x, y, p). \end{aligned} \tag{2.7}$$

Note that $(x(p), y(p))$ is the exact minimizer of $L^{(\mu)}$ with p fixed, which uniquely exists because of the strong convexity of $L^{(\mu)}$ in (x, y). In this paper, we assume that such exact minimizer can actually be computed. Then, by the definition (2.1) of $\hat{f}_k$, the flow conservation equation for each commodity is always satisfied by $(x(p), y(p))$. Note also that $\psi^{(\mu)}$ is a closed concave function [25, Th. 12.1]. Since $(x(p), y(p))$ is uniquely obtained, so the function $\psi^{(\mu)}$ is continuously differentiable [21, §8.5 Cor. 1] and its gradient is given by

$$\nabla \psi^{(\mu)}(p) = y(p) - \sum_{k=1}^{K} x_k(p) - \frac{1}{\gamma^{(\mu)}}(p - p^{(\mu)}). \tag{2.8}$$

Using (2.6) and (2.7), the formula (2.5) can be written as

$$p^{(\mu+1)} \approx \arg \max_{p \in R^n} \psi^{(\mu)}(p) \tag{2.9}$$

and

$$\left(x^{(\mu+1)}, y^{(\mu+1)}\right) = \left(x(p^{(\mu+1)}), y(p^{(\mu+1)})\right). \tag{2.10}$$

The above maximization of $\psi^{(\mu)}$ in (2.9) is a nonlinear unconstrained smooth optimization problem, which may usually be solved iteratively. Since the function $\psi^{(\mu)}$ is continuously differentiable, we can use a gradient-based algorithm like a quasi-Newton method. The gradient $\nabla \psi^{(\mu)}(p)$ is obtained as a by-product when we compute the value of $\psi^{(\mu)}(p)$. This implies that $(x(p), y(p))$ of (2.6) is calculated for every p in the course of iterations to maximize $\psi^{(\mu)}$. For the iterations to maximize $\psi^{(\mu)}$, we may adopt one of the following two termination criteria:

$$|\nabla \psi^{(\mu)}(p^{(\mu+1)})| \leq \frac{\epsilon^{(\mu)}}{\gamma^{(\mu)}}, \tag{2.11}$$

where $\epsilon^{(\mu)}$ are positive constants such that $\sum_{\mu=0}^{\infty}\epsilon^{(\mu)} < \infty$, and

$$|\nabla\psi^{(\mu)}(p^{(\mu+1)})| \le \frac{\delta^{(\mu)}}{\gamma^{(\mu)}}|(x(p^{(\mu+1)}), y(p^{(\mu+1)}), p^{(\mu+1)}) - (x^{(\mu)}, y^{(\mu)}, p^{(\mu)})|, \tag{2.12}$$

where $\delta^{(\mu)}$ are positive constants such that $\sum_{\mu=0}^{\infty}\delta^{(\mu)} < \infty$. Note that checking these criteria requires evaluating $(x^{(\mu+1)}, y^{(\mu+1)})$, because we need to know the values of $x^{(\mu+1)}$ and $y^{(\mu+1)}$ when we evaluate $\nabla\psi^{(\mu)}(p^{(\mu+1)})$ (see (2.8)).

By transferring the results in [16, 26] to the context of the algorithm based on the formula (2.5), or equivalently (2.9)-(2.10), we may obtain the following convergence theorem. Recall that the mapping T^{-1} is said to be Lipschitz continuous at the origin with modulus $\alpha \ge 0$, if $T^{-1}(0)$ is single-valued and there exists $\tau > 0$ such that $|z - T^{-1}(0)| \le a|w|$ for all $z \in T^{-1}(w)$ and $|w| \le \tau$ [26].

Theorem. *Suppose that problem P is strongly consistent and has at least one optimal solution.*

(a) If the algorithm is executed under criterion (2.11) *with a sequence* $\{\gamma^{(\mu)}\}$ *bounded away from zero, then the sequence* $\{(x^{(\mu)}, y^{(\mu)}, p^{(\mu)})\}$ *generated by (2.5) is bounded and converges to* $(\bar{x}, \bar{y}, \bar{p})$, *where* $\bar{x}$ *and* $\bar{y}$ *are optimal for problem P and* $\bar{p}$ *is optimal for problem D.*

(b) Let $T_L : R^{nK} \times R^n \times R^n \to R^{nK} \times R^n \times R^n$ *be multifunction (point-to-set mapping) given by*

$$\begin{aligned} T_L(x,y,p) &= \{(u,v,w)\,|\,u_k \in \partial_{x_k}L(x,y,p), \forall k, v \in \partial_y L(x,y,p), w \in -\partial_p L(x,y,p)\} \\ &= \left\{(u,v,w)\,\middle|\, u_k \in \partial f_k(x_k) - p, \forall k, v \in \partial g(y) - p, w = \sum_{k=1}^{K} x_k - y\right\}. \end{aligned}$$

Assume that T_L^{-1} *is Lipschitz continuous at the origin with modulus* $\alpha \ge 0$. *If the algorithm is executed under criterion* (2.12) *with a sequence* $\{\gamma^{(\mu)}\}$ *nondecreasing, then the sequence* $\{(x^{(\mu)}, y^{(\mu)}, p^{(\mu)})\}$ *is bounded and converges to* $(\bar{x}, \bar{y}, \bar{p})$, *where* $(\bar{x}, \bar{y})$ *and* $\bar{p}$ *are the unique optimal solutions for P and D, respectively. In addition, there exists an integer* $\bar{\mu}$ *such that*

$$|(x^{(\mu+1)}, y^{(\mu+1)}, p^{(\mu+1)}) - (\bar{x}, \bar{y}, \bar{p})| \le \theta^{(\mu)}|(x^{(\mu)}, y^{(\mu)}, p^{(\mu)}) - (\bar{x}, \bar{y}, \bar{p})|, \quad \forall \mu \ge \bar{\mu},$$

where

$$\theta^{(\mu)} = \{\alpha(\alpha^2 + \gamma^{(\mu)})^{-1/2} + \delta^{(\mu)}\}(1 - \delta^{(\mu)})^{-1}.$$

3. Algorithm for Separable Problems

In this section, we assume that the cost function of problem P is separable, that is, the functions f_k and g in problem P are given by

$$f_k(x_k) = \sum_{j=1}^{n} f_{kj}(x_{kj}), \quad \text{for all } k = 1, 2, \ldots, K \tag{3.1}$$

and

$$g(y) = \sum_{j=1}^{n} g_j(y_j), \tag{3.2}$$

respectively. (See (1.3) and (1.4).) We assume that the functions f_{kj} and g_j are closed proper convex.

We specify the detail of the proposed algorithm, which exploits the special structure of the problem. From the separability of f_k and g, it follows that the function $L^{(\mu)}$ in (2.3) is separable in x_k and y_j when p is fixed. Therefore, the minimization of $L^{(\mu)}$ appearing in (2.6) is carried out separately with respect to x_k and y_j. Specifically, $(x(p), y(p))$ in (2.6) is evaluated as follows:

$$x_k(p) = \arg\min_{x_k \in R^n} \left\{ \sum_{j=1}^{n} \left\{ f_{kj}(x_{kj}) + \frac{1}{2\gamma^{(\mu)}}(x_{kj} - x_{kj}^{(\mu)})^2 - p_j x_{kj} \right\} \middle| \, E x_k = b_k \right\}, \tag{3.3}$$

for all $k = 1, 2, \ldots, K$, and

$$y_j(p_j) = \arg\min_{y_j \in R} \left\{ g_j(y_j) + \frac{1}{2\gamma^{(\mu)}}(y_j - y_j^{(\mu)})^2 + p_j y_j \right\}, \tag{3.4}$$

for all $j = 1, 2, \ldots, n$.

In (3.3), the computation of $x_k(p)$ amounts to solving a single commodity network flow problem whose objective function is separable and strongly convex. In practice, a variety of methods are available to solve such problems [13, 15, 20, 28]. In (3.4), the computation of $y_j(p_j)$ becomes a univariate minimization problem with a strongly convex objective function. The minimizer $y_j(p_j)$ may often be expressed in a closed form, or it can at least be computed accurately enough by using an appropriate one-dimensional optimization technique such as binary section method, golden section method and quasi-Newton method.

Recall that the maximization in (2.9) is a differentiable unconstrained optimization problem whose objective function $\psi^{(\mu)}(p)$ and its gradient $\nabla\psi^{(\mu)}(p)$ may be computed from $x(p)$ and $y(p)$, which are obtained by the minimization on the right-hand side of (3.3) and (3.4). Since the latter minimization problems are in general solved iteratively, the proposed algorithm may have a triple-loop structure. Namely, the outmost loop is the iteration of the primal-dual proximal point algorithm, the middle one is the iteration of maximizing $\psi^{(\mu)}$ to compute $p^{(\mu+1)}$ by (2.9), and the inmost one is the iteration to compute $x(p)$ and $y(p)$.

References

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

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

[3] Bertsekas, D.P.: "Algorithms for nonlinear multicommodity network flow problems," *International Symposium on Systems Optimization and Analysis* (eds. A. Bensoussan and J.L. Lions). Springer, New York, NY, 1979, 210-224.

[4] Censor Y., Chajakis E.D. and Zenios S.A.: "A primal-dual algorithm for large-scale nonlinear multicommodity network problems and its parallel implementations," Report 91-09-03 1991.

[5] Chen, R.J. and Meyer R.R.: "Parallel optimization for traffic assignment," *Mathematical Programming* Vol. 42 (1988), 327-345.

[6] Chen, G. and Teboulle, M.: "Convergence analysis of a proximal-like minimization algorithm using Bregman functions," TR-90-23, Department of Mathematics, University of Maryland, Baltimore, MD, 1990.

[7] Eckstein, J. and Bertsekas, D.P.: "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," Working Paper 90-033, Harvard Business School, Boston, MA, 1989.

[8] Florian, M.: "An improved linear approximation algorithm for the network equilibrium (packet switching) problem," *IEEE Proceedings, Decision and Control* (1977), 812-818.

[9] Florian, M.: "Mathematical programming applications in national, regional and urban planning," *Mathematical Programming, Recent Development and Applications* (eds. M. Iri and K. Tanabe), Kluwer Academic Publishers, Dordrecht, Holland, 1989, 57-81.

[10] Fukushima, M.: "A modified Frank-Wolfe algorithm for solving the traffic assignment problem, " *Transportation Research* Vol. 18B (1984), 169-177.

[11] Fukushima, M.: "On the dual approach to the traffic assignment problem," *Transportation Research* Vol. 18B (1984), 235-245.

[12] Fukushima, M.: "A nonsmooth optimization approach to nonlinear multicommodity network flow problems," *Journal of Operarions Research Society of Japan* Vol. 27 (1984), 151-177.

[13] Hager, W.W. and Hearn, D.W.: "The dual active set algorithm and quadratic networks," Research Report No. 90-7, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 1990.

[14] Hearn, D.W. and Lawphongpanich, S.: "A dual ascent algorithm for traffic assignment problems," *Transportation Research* Vol. 24B (1990), 423-430.

[15] Ibaraki, S., Fukushima, M. and Ibaraki, T.: "Dual-based Newton method for nonlinear minimum cost network flow problems," *Journal of the Operations Research Society of Japan* Vol. 34 (1991), 263-286.

[16] Ibaraki, S., Fukushima, M. and Ibaraki, T.: "Primal-dual proximal point algorithm for linearly constrained convex programming problems," *Computational Optimization and Applications* Vol. 1 (1992) ?-?.

[17] Ibaraki, T. and Fukushima, M.: *FORTRAN 77 Optimization Programming* (in Japanese), Iwanami-Shoten, Tokyo, 1991.

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

[19] Kennington, J.L. and Helgason, R.: *Algorithms for Network Programming.* John Wiley, New York, NY, 1980.

[20] Klincewicz, J.G.: "Implementing an "exact" Newton method for separable convex transportation problems," *Networks*, Vol. 19 (1989), 95-105.

[21] Lasdon, L.S.: *Optimization Theory for Large Systems.* Macmillan, New York, NY, 1970.

[22] LeBlanc L.J., Morlok E.K. and Pierskalla W.P.: "An efficient approach to solving the road network equilibrium traffic assignment problem," *Transportation Research.* Vol. 9 (1975), 309-318.

[23] Minoux, M.: *Mathematical Programming: Theory and Algorithms.* John Wiley, New York, NY, 1986.

[24] Nagamochi, H., Fukushima, M. and Ibaraki, T.: "Relaxation method for the strictly convex multicommodity flow problem with capacity constraints on individual commodities," *Networks* Vol. 20 (1990), 409-426.

[25] Rockafellar, R.T.: *Convex Analysis.* Princeton University Press. Princeton, NJ, 1970.

[26] Rockafellar, R.T.: "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research* Vol. 1 (1976), 97-116.

[27] Teboulle, M.: "Entropic proximal mappings with applications to nonlinear programming," TR-90-15, Department of Mathematics, University of Maryland, Baltimore, MD, 1990.

[28] Tseng, P. and Bertsekas, D.P.: "Relaxation methods for problems with strictly convex separable costs and linear constraints," *Mathematical Programming*, Vol. 38 (1987), 303-321.

[29] Wright, S.J.: "Implementing proximal point methods for linear programming," *Journal of Optimization Theory and Applications* Vol. 65 (1990), 531-554.

[30] Zenios S.A.: "On the fine-grain decomposition of multicommodity transportation problems," *SIAM Journal on Optimization* Vol. 1 (1991), 643-669.