Journal of the Operations Research
Society of Japan
Vol. 27, No. 2, June 1984

© 1984 The Operations Research Society of Japan

A NONSMOOTH OPTIMIZATION APPROACH TO NONLINEAR MULTICOMMODITY NETWORK FLOW PROBLEMS

Masao Fukushima
Kyoto University

(Received September 2, 1983: Revised February 10, 1984)

Abstract This paper is concerned with the convex cost multicommodity flow problem. First, it is shown that the Lagrangian dual of the problem can be formulated as a convex nonsmooth optimization problem. Then, an algorithm is presented for solving a problem which is obtained by slightly modifying the dual problem so as to be amenable to shortest chain algorithms. Since the proposed algorithm mainly consists of successively solving shortest chain problems and linearly constrained subproblems whose size is independent of the underlying network structure, we may expect that it can solve fairly large problems. Convergence of the algorithm is proved and several remarks on its implementation are given. Finally, limited computational experience with the proposed algorithm is reported.

1. Introduction

One of the most important network optimization problems is the class of multicommodity flow problems [1,18]. The multicommodity network flow problems may be classified into two categories, linear and nonlinear models, according to a type of the cost function to be optimized. The nonlinear models have, in particular, been extensively studied in connection with design and analysis of urban traffic systems and communication systems. Typically, the equilibrium traffic assignment problem in a transportation network [2,6,20] and the optimal routing problem in a packet switched computer network [5,10,14] lead to convex cost multicommodity flow problems. In those problems, the nonlinear dependence of costs on the flow is intrinsic because of congestion effects in the network.

In this paper, we present a new approach to the solution of convex cost multicommodity flow problems. We first show that the dual of the multicommodity flow problem can be formulated as a problem of maximizing a concave objective function subject to simple bounds on the variables. We also

point out that the dual objective function is nondifferentiable but its functional values and subgradients can conveniently be calculated by using an efficient shortest chain algorithm. We then develop a nonsmooth optimization algorithm of descent type for solving the dual problem. The algorithm is a modified and extended version of the one presented in [11], which takes special features of the problem into account. Once the dual problem is solved, it is straightforward to recover an optimal solution of the original multicommodity flow problem.

Our approach may thus be considered an adaptation of a nonsmooth optimization technique to the dual method for the nonlinear multicommodity flow problems and is significantly different from those which are available for problems of similar nature. Indeed, the majority of existing algorithms such as [3,8,13,20] fall into the category of feasible direction methods for linearly constrained optimization. Duality results concerning the traffic assignment problems may be found in [12,15], but it appears that only a few attempts have been made to solve those problems from a dual or primal-dual point of view [23,26]. On the other hand, the column generation technique, which is essentially a dual method, has been adapted to solve a variety of linear cost multicommodity flow problems [9,29]. It is also noted that the subgradient algorithm, which is a popular heuristic nonsmooth optimization technique, has been successfully applied in the context of the primal decomposition approach to the linear multicommodity flow problems [16,19].

This paper is organized as follows: In the next section, we explicitly formulate the convex cost multicommodity flow problem. In Section 3, we derive the dual problem and show that an optimal solution of the multicommodity flow problem can be obtained by solving it. In Sections 4 and 5, we describe an algorithm for solving the dual problem, prove its convergence, and make some comments on the implementation of the algorithm. In Section 6, we present results of numerical experiments to demonstrate the effectiveness of the proposed algorithm.

2. The Nonlinear Multicommodity Flow Problem

Consider a directed network with a finite set N of nodes and a finite set A of arcs. Let K denote the set of commodities to be transported through the network. We suppose without loss of generality that each commodity has a single origin-destination (O-D) pair which is different from those of other commodities.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

The minimum cost multicommodity flow problem may be formulated as

$$\text{(1)} \qquad \min \sum_{a \in A} f_a \Big(\sum_{k \in K} x_a^k \Big)$$

s.t.

$$\sum_{a \in O(i)} x_a^k - \sum_{a \in I(i)} x_a^k = \begin{cases} D_k & \text{if } i = s^k \\ -D_k & \text{if } i = t^k \\ 0 & \text{otherwise} \end{cases}$$

$$x_a^k \geq 0 \qquad \forall a \in A \,, \quad \forall k \in K \,,$$

where

x_a^k : flow of commodity k on arc a

D_k : known nonnegative flow requirement for commodity k

(s^k, t^k): O-D pair for commodity k

$O(i)$: set of arcs originating at node i

$I(i)$: set of arcs terminating at node i

f_a : cost function for arc a .

In words, the problem is to minimize the sum of arc costs which are functions of total arc flows

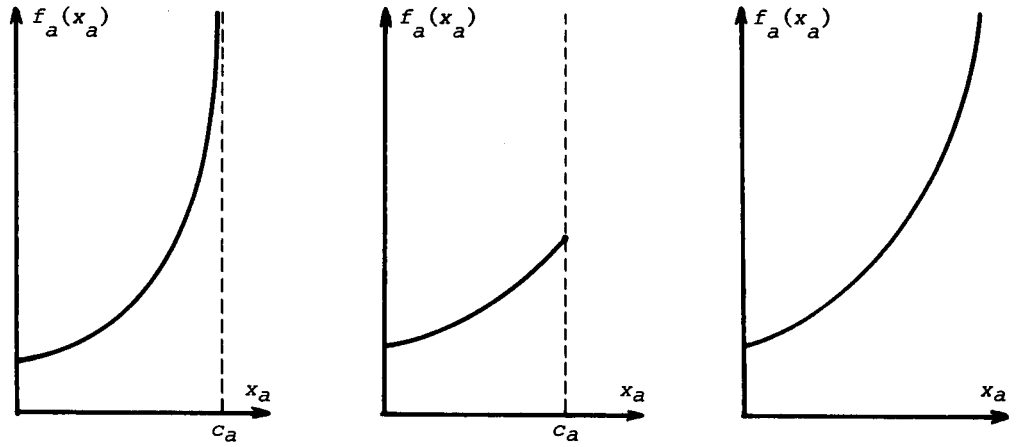
$$x_a = \sum_{k \in K} x_a^k \,,$$

subject to the flow conservation equations and the nonnegativity constraints.

The arc cost functions f_a are supposed to be nonlinear in order to take into account congestion effects in the network. Many authors assume each function f_a is everywhere finite-valued but $f_a(x_a)$ increases radically as x_a exceeds a substantial capacity of that arc, thereby forcing the solution of problem (1) to meet capacity requirements. This trick may sometimes be helpful, because then the problem turns out to be formally uncapacitated. In this paper, however, we allow the possibility of including arc capacities explicitly by assigning the value $+\infty$ to $f_a(x_a)$ for x_a greater than the capacity of arc a . It will be seen that such an extension does not cause any difficulty in dealing with problem (1) by way of its dual problem.

In the sequel, we assume the following: For each $a \in A$, f_a is a closed convex function with effective domain $\operatorname{dom} f_a = \{ x_a \mid f_a(x_a) < +\infty \}$ being an interval such that $[0, c_a)$ or $[0, c_a]$, where c_a is a (possibly infinite) positive capacity of arc a . Moreover, f_a is strictly convex and nondecreasing on $\operatorname{dom} f_a$, and $\lim_{t \to +\infty} f_a(t)/t = +\infty$ whenever $c_a = +\infty$.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.



(a) $c_a < +\infty$, $D_a = [0, c_a)$ (b) $c_a < +\infty$, $D_a = [0, c_a]$ (c) $c_a = +\infty$, $D_a = [0, +\infty)$

Figure 1. Arc cost functions f_a

These assumptions are natural enough to meet desirable properties of the cost functions. Typical examples of f_a are illustrated in Figure 1.

The minimum cost multicommodity flow problem can alternatively be stated in terms of flows through chains of the network. To be specific, let P_k denote the set of all chains of arcs between O-D pair (s^k, t^k) for commodity k. Since O-D pairs are distinct for all commodities, we have

$$P_k \cap P_{k'} = \emptyset \qquad \text{if} \quad k \neq k' .$$

Let us denote by y_p the flow through chain p. Then the total arc flows x_a and the chain flows y_p can be related by

$$x_a = \sum_{k \in K} \sum_{p \in P_k} \delta_{ap} y_p \qquad \forall a \in A , \tag{2}$$

where δ_{ap} are elements of the arc-chain incidence matrix of the network, i.e.,

$$\delta_{ap} = \begin{cases} 1 & \text{if arc } a \text{ is on chain } p \\ 0 & \text{otherwise .} \end{cases}$$

The arc-chain formulation of the minimum cost multicommodity flow problem is now given by

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$(3) \qquad \min \quad \sum_{a\in A} f_a\Big(\sum_{k\in K}\sum_{p\in P_k} \delta_{ap} y_p\Big)$$

$$\text{s.t.} \quad \sum_{p\in P_k} y_p = D_k , \qquad \forall k \in K$$

$$y_p \geq 0 , \qquad \forall p \in P_k , \ \forall k \in K .$$

The above formulation presupposes that we have an enumeration of chains for each O-D pair (s^k,t^k) . This appears to be unrealistic for networks one encountered in practice. In the development to follow, however, we shall be concerned with the arc-chain formulation (3) rather than the node-arc formulation (1). This is simply for convenience of exposition and, in fact, the algorithm proposed later will not require such chain enumeration.

3. Dual Problem and Its Properties

In this section, we develop duality theory for the multicommodity flow problem as formulated by (3). In what follows, we shall make extensive use of results from convex analysis [27]. In particular, for an arbitrary convex function f , we denote by f^* the convex conjugate function of f , i.e.,

$$f^*(u) = \sup_x \{ u^T x - f(x) \} , \qquad \forall u .$$

Also, we denote by $\partial f(x)$ the set of subgradients of f at x , i.e.,

$$\partial f(x) = \{ u \mid f(x') \geq f(x) + u^T(x' - x) , \ \forall x' \} .$$

Using the relation (2), we may rewrite problem (3) as the following linearly constrained convex programming problem

$$(4) \qquad \min \quad \sum_{a\in A} f_a(x_a)$$

$$\text{s.t.} \quad x_a = \sum_{k\in K}\sum_{p\in P_k} \delta_{ap} y_p , \qquad \forall a \in A$$

$$\sum_{p\in P_k} y_p = D_k , \qquad \forall k \in K$$

$$y_p \geq 0 , \qquad \forall p \in P_k , \ \forall k \in K .$$

The Lagrangian for problem (4) is then defined to be

$$L(x,y,u,v) = \sum_{a\in A} f_a(x_a) - \sum_{a\in A} u_a \{ x_a - \sum_{k\in K}\sum_{p\in P_k} \delta_{ap} y_p \}$$

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$- \sum_{k \in K} v_k \{ \sum_{p \in P_k} y_p - D_k \} ,$$

where u_a and v_k are Lagrange multipliers (dual variables), and x, y, u and v are the vectors with elements x_a, y_k, u_a and v_k, respectively.

Now we may state the Lagrangian dual to problem (4) as

$$\max \quad \ell(u,v) , \tag{5}$$

where the dual objective function ℓ is given by

$$\begin{aligned} \ell(u,v) &= \inf_{x,y} \{ L(x,y,u,v) \mid y \geq 0 \} \\ &= \inf_{x} \sum_{a \in A} \{ f_a(x_a) - u_a x_a \} + \sum_{k \in K} D_k v_k \\ &\quad + \inf_{y \geq 0} \sum_{k \in K} \sum_{p \in P_k} \{ \sum_{a \in A} u_a \delta_{ap} - v_k \} y_p \\ &= \begin{cases} -\sum_{a \in A} f_a^*(u_a) + \sum_{k \in K} D_k v_k & \text{if } \sum_{a \in A} u_a \delta_{ap} \geq v_k , \ \forall p \in P_k , \ \forall k \in K \\ -\infty & \text{otherwise .} \end{cases} \end{aligned} \tag{6}$$

In view of (6), problem (5) is explicitly represented as

$$\begin{aligned} &\max \quad - \sum_{a \in A} f_a^*(u_a) + \sum_{k \in K} D_k v_k \\ &\text{s.t.} \quad \sum_{a \in A} u_a \delta_{ap} \geq v_k , \qquad \forall p \in P_k , \ \forall k \in K . \end{aligned} \tag{7}$$

Noting that D_k are all nonnegative, we may further rewrite problem (7) as

$$\max \quad - \sum_{a \in A} f_a^*(u_a) + \sum_{k \in K} D_k \min_{p \in P_k} \sum_{a \in A} u_a \delta_{ap} . \tag{8}$$

Since the conjugates f_a^* are closed convex functions and the second term of (8) is piecewise concave in u, (8) is a problem of maximizing a concave function which involves only the variables u_a.

It is important to note that each f_a^* is a function of a single variable u_a. This fact enables us to obtain an explicit representation of f_a^* in most application situations. Furthermore, as a consequence of the assumptions on f_a, the functions f_a^* enjoy quite favorable properties as shown below.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Proposition 1. For each $a \in A$, the function f_a^* is everywhere finite and continuously differentiable.

Proof: The assumptions on f_a imply that each f_a is co-finite [27, p.116]. It then follows from [27,Cor.13.3.1] that the conjugate f_a^* is everywhere finite-valued. The continuous differentiability of f_a^* follows from [27,Thm.26.3 and Cor.25.5.1]. □

Now let us turn our attention to the second term of the objective function of problem (8). By inspection, it is easy to see that the minimization problem

$$\min_{p \in P_k} \sum_{a \in A} u_a \delta_{ap} \tag{9}$$

is nothing but a problem of finding the shortest chain between O-D pair (s^k, t^k) in the network with arc lengths given by u_a, $a \in A$. We may therefore use an efficient shortest chain (path) algorithm [7] to evaluate (9). However, since the shortest chain algorithm generally requires arc lengths to be nonnegative, it will be convenient to make a slight modification of problem (8).

Let α_a denote the right derivative of f_a at $x_a = 0$, i.e.,

$$\alpha_a = \lim_{t \downarrow 0} \frac{f_a(t) - f_a(0)}{t} .$$

Since f_a is assumed to be convex and nondecreasing on dom f_a, α_a exists and is nonnegative for every $a \in A$.

We consider the problem

$$\begin{aligned} \max \quad & -\sum_{a \in A} f_a^*(u_a) + \sum_{k \in K} D_k \min_{p \in P_k} \sum_{a \in A} u_a \delta_{ap} \\ \text{s.t.} \quad & u \geq \alpha , \end{aligned} \tag{10}$$

where α is the vector with elements α_a, $a \in A$. The only difference between problems (8) and (10) consists in the existence of the simple bounds on u_a in the latter problem. It is to be noted that the shortest chain problem (9) does not contain any arc with negative length, provided u is feasible to problem (10).

The next proposition assures that problem (8) can be replaced by problem (10) without loss of generality.

Proposition 2. (i) The optimal values of problems (8) and (10) are equal. (ii) Any optimal solution of problem (10) also solves problem (8).

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

(iii) If problem (8) has an optimal solution, then problem (10) also has an optimal solution.

Proof: Under our assumptions on f_a, it can be shown that

$$f_a^*(u_a) = -f_a(0) \quad \text{for any } u_a \leq \alpha_a . \tag{11}$$

To prove (i), suppose there exists a point $\tilde{u}$ such that the corresponding objective value is strictly greater than the maximum value of problem (10). Then, it is clear that $\tilde{u}$ violates the constraints $u \geq \alpha$ of problem (10). Let $\tilde{u}'$ be defined by

$$\tilde{u}'_a = \begin{cases} \tilde{u}_a & \text{if } \tilde{u}_a \geq \alpha_a \\ \alpha_a & \text{if } \tilde{u}_a < \alpha_a \end{cases} . \tag{12}$$

Obviously, $\tilde{u}'$ is feasible to problem (10). Moreover, the objective value at $\tilde{u}'$ is lo less than that at $\tilde{u}$, because of (10) and the fact that the second term of the objective function is nondecreasing with respect to u. This is a contradiction and proves (i).

Since problem (8) is a relaxation of problem (10), (ii) follows trivially from (i). To see (iii) holds, it suffices to note that, if $\tilde{u}$ is an optimal solution of problem (8), then the point $\tilde{u}'$ constructed by (12) turns out to be optimal to problem (10). □

In order to validate our dual approach to the multicommodity flow problem, it remains to show that the optimal solution of problem (4) can actually be obtained by solving problem (10). The following proposition asserts this.

Proposition 3. Suppose problem (4) has a feasible solution such that $x_a \in \text{int dom } f_a$ for all $a \in A$, where $\text{int dom } f_a$ stands for the interior of the interval $\text{dom } f_a$. Then, both problems (4) and (10) have optimal solutions and the optimal values of the two problems are equal to each other. Furthermore, if $\bar{u}$ solves problem (10), then the point $\bar{x}$ uniquely determined by the formula

$$\bar{x}_a = \nabla f_a^*(\bar{u}_a) , \qquad a \in A \tag{13}$$

constitutes a part of an optimal solution of problem (4).

Proof: Consider problem (4) and its dual (7). Clearly, problem (4) has an optimal solution, say $(\bar{x},\bar{y})$. Then, the hypothesis of the proposition guarantees [27,Cor.29.1.4] that there exists a vector $(\bar{u},\bar{v})$ satisfying the following Kuhn-Tucker conditions [27,Thm.28.3] for problem (4):

$$\bar{u}_a \in \partial f_a(\bar{x}_a) , \qquad \forall a \in A$$

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$\bar{x}_a = \sum_{k\in K} \sum_{p\in P_k} \delta_{ap} \bar{y}_p , \qquad \forall a \in A$$

$$\sum_{p\in P_k} \bar{y}_p = D_k , \qquad \forall k \in K$$

$$\left.\begin{aligned} &\sum_{a\in A} \bar{u}_a \delta_{ap} \geq \bar{v}_k , \quad \bar{y}_p \geq 0 \\ &\bar{y}_p > 0 \implies \sum_{a\in A} \bar{u}_a \delta_{ap} = \bar{v}_k \end{aligned}\right\} \qquad \forall p \in P_k , \ \forall k \in K .$$

Therefore, it follows from [27,Thms.30.3-5] that $(\bar{u},\bar{v})$ solves the dual problem (7) and the optimal values of problems (4) and (7) are equal. Moreover, since f_a^* is differentiable by Proposition 1 and ∇f_a^* is the inverse mapping of ∂f_a [27,Cor.23.5.1], the formula (13) follows from the first relation in the above Kuhn-Tucker conditions.

To complete the proof, it simply suffices to recall that problem (7) is a restatement of problem (8) and problem (8) is essentially equivalent to problem (10) by Proposition 2. □

To conclude this section, we shall briefly mention an economic interpretation of the dual variables u and v. To be specific, let us suppose that problem (4) represents an equilibrium (user-optimization) traffic assignment problem (see, e.g., [6]). Then, it is easy to see that the Kuhn-Tucker conditions appearing in the proof of Proposition 3 are nothing but a restatement of the Wardrop's principle [30] saying that the travel times on all the routes actually used are equal and no greater than those which would be experienced by a single vehicle on any unused route. In this context, it is quite reasonable to regard u_a and v_k as the travel time along arc a and the minimum travel time between O-D pair (s^k,t^k), respectively (see the first and the last relations in the Kuhn-Tucker conditions). Therefore, the dual problem (7),(8) or (10) may be considered an alternative formulation of the traffic assignment problem, in which the variables are travel times, rather than traffic flows, in the transportation network.

4. Algorithm for Solving the Dual Problem

As shown in the previous section, the dual of the multicommodity flow problem can be formulated as problem (10) which involves a concave but nondifferentiable objective function. We might therefore apply any of the existing general-purpose algorithms for nonsmooth optimization [22,25] to

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

solve problem (10). In this and the next sections, however, we shall develop a specialized algorithm for solving problem (10). The proposed algorithm is an extension of the nonsmooth optimization method presented in [11] and is designed to take special features of problem (10) into account. In particular, it may take advantage of second order information on the objective function, and hence is expected to work more efficiently than those which exploit only subgradients of the objective function.

In the development to follow, we shall be concerned with the problem equivalent to (10):

$$\begin{aligned} (14) \qquad & \min \quad F(u) \\ & \text{s.t.} \quad u \geq \alpha \ , \end{aligned}$$

where F is a convex function defined by

$$(15) \qquad F(u) = \sum_{a \in A} f_a^*(u_a) - \sum_{k \in K} D_k \min_{p \in P_k} \sum_{a \in A} u_a \delta_{ap} \ .$$

In this section, we present a basic framework of the algorithm for solving problem (14) and prove its convergence. Subsequently, in the next section, we shall discuss the implementation of the algorithm in some detail.

The algorithm is doubly iterative in itself and starts with a point u^1 arbitrarily chosen from the feasible region of problem (14), i.e., $u^1 \geq \alpha$. Let ε, β and m be prescribed real parameters such that $\varepsilon > 0$, $0 < \beta < 1$ and $0 < m < 1$.

Generally, on the kth major iteration, we are given a point u^k such that $u^k \geq \alpha$ and try to obtain an approximate solution of the problem

$$\begin{aligned} (16) \qquad & \min \quad \phi_k(z) \\ & \text{s.t.} \quad z \geq \alpha \ , \end{aligned}$$

where ϕ_k is a function defined by

$$(17) \qquad \phi_k(z) = \tfrac{1}{2}(z-u^k)^T B_k (z-u^k) + F(z)$$

and B_k is a positive definite matrix. Problem (16) is again a nonsmooth optimization problem, but involves a strongly convex objective function.

It is noted that, if B_k is chosen as the identity matrix I for each k, problem (16) reduces to the subproblem of the proximal point algorithm [28] and its solution provides the minimum norm ε-subgradient [27, p.219] for some $\varepsilon > 0$. (See [11] for detailed arguments.) Here we do not restrict ourselves to the special case $B_k \equiv I$, in order to better incorporate the second order information of the objective function.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

We now consider applying a cutting plane type procedure to problem (16). More specifically, we successively find an optimal solution z^i to the following quadratic programming subproblem

$$\text{(18)} \qquad \min \quad \frac{1}{2}(z-u^k)^T B_k (z-u^k) + \eta$$

$$\text{s.t.} \quad F(z^j) + (g^j)^T(z-z^j) \le \eta , \qquad j = 0,1,\ldots,i-1$$

$$z \ge \alpha ,$$

where $z^0 = u^k$ and each g^j is a subgradient of F at z^j. We shall call this process the inner iteration.

Let $\bar{\psi}_{k,i}$ denote the optimal value of problem (18). Since g^j are subgradients of F at z^j, it is not difficult to see that $\bar{\psi}_{k,i}$ always affords a lower bound of the optimal value of problem (16). Moreover, as will be shown shortly (Proposition 4), we must eventually have either

$$\text{(19)} \qquad F(u^k) \le \bar{\psi}_{k,i} + \varepsilon$$

or

$$\text{(20)} \qquad \phi_k(z^i) - \bar{\psi}_{k,i} \le \beta \{ F(u^k) - \bar{\psi}_{k,i} \}$$

for some i, where ε and β are the parameters specified previously. The conditions (19) and (20) serve as stopping criteria for the major and inner iterations, respectively.

Therefore, for each i, we first check the condition (19). If (19) is satisfied, we terminate the major iteration itself. When (19) is not met, we next check the condition (20). If (20) also fails to hold, we continue the inner iteration with i increased by one. On the other hand, if (20) is achieved, we define a search direction by

$$d^k = z^i - u^k$$

and then determine a step length $t_k \ge 1$ satisfying

$$\text{(21)} \qquad F(u^k + t_k d^k) \le F(u^k) + m t_k \sigma_k$$

and

$$\text{(22)} \qquad u^k + t_k d^k \ge \alpha ,$$

where σ_k is given by

$$\text{(23)} \qquad \sigma_k = -(1-\beta)\varepsilon - \frac{1}{2}(d^k)^T B_k d^k .$$

Once t_k is determined, we set

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$u^{k+1} = u^k + t_k d^k$$

and proceed to the next major iteration with k increased by one. It is to be noted that the sequence $\{u^k\}$ thus generated lies in the feasible region of problem (14) and $\{F(u^k)\}$ is a monotonically decreasing sequence.

The above procedure is summarized as follows:

Step 1: Choose $u^1 \geq \alpha$ and set $k = 1$.

Step 2: Select B_k, put $z^0 = u^k$, choose $g^0 \in \partial F(z^0)$ and set $i = 1$.

Step 3: Solve (18) to obtain z^i and $\bar{\psi}_{k,i}$.

Step 4: If (19) is satisfied, terminate. Otherwise, go to Step 5.

Step 5: If (20) is satisfied, then set $d^k = z^i - u^k$ and go to Step 6. Otherwise, choose $g^i \in \partial F(z^i)$, set $i = i+1$ and return to Step 3.

Step 6: Determine $t_k \geq 1$ satisfying (21) and (22). Set $u^{k+1} = u^k + t_k d^k$ and $k = k+1$, and return to Step 2.

The next two propositions demonstrate that each step of this algorithm is always consistent.

Proposition 4. At each major iteration k, either (19) or (20) is satisfied for some i.

Proof: Suppose to the contrary that, for some k, neither (19) nor (20) is achieved for each i. Then, we must have

$$F(u^k) > \bar{\psi}_{k,i} + \varepsilon$$

and

$$\phi_k(z^i) - \bar{\psi}_{k,i} > \beta \{ F(u^k) - \bar{\psi}_{k,i} \}$$

for all i. Combining these inequalities, we get

$$\phi_k(z^i) - \bar{\psi}_{k,i} > \beta\varepsilon \tag{24}$$

for all i. However, it can be shown in a way similar to [11,Prop.3] that

$$\lim_{i\to\infty} \phi_k(z^i) = \lim_{i\to\infty} \bar{\psi}_{k,i} = \phi_k^*,$$

where ϕ_k^* is the optimal value of problem (16), which is finite by the strong convexity of ϕ_k. This contradicts (24) and the proof is complete. □

Proposition 5. Suppose that, at major iteration k, (19) fails to hold but (20) is satisfied for some z^i. Then, the set of step lengths $t_k \geq 1$ satisfying (21) and (22) is nonempty.

Proof: It suffices to show that the step length of unity, $t_k = 1$, satisfies (21) and (22). Since it is clear that $u^k + d^k = z^i \geq \alpha$ by the con-

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

struction of z^i, we only prove that (21) is satisfied by $t_k = 1$. By our hypotheses, we have

$$F(u^k) > \bar{\psi}_{k,i} + \varepsilon$$

and

$$\phi_k(z^i) \leq \beta F(u^k) + (1-\beta)\bar{\psi}_{k,i} .$$

These inequalities together with (17) imply

$$F(z^i) < F(u^k) - (1-\beta)\varepsilon - \frac{1}{2}(z^i-u^k)^T B_k (z^i-u^k) .$$

Noting that $z^i = u^k + d^k$, we obtain from the last inequality and (23)

$$F(u^k+d^k) < F(u^k) + \sigma_k .$$

This implies that (21) is fulfilled by $t_k = 1$, because $0 < m < 1$ and $\sigma_k < 0$. □

The proof of the last proposition suggests that we may simply set $t_k = 1$ for all k. From the viewpoint of computational efficiency, however, it is tempting to obtain more satisfactory improvement in the objective value at each iteration by allowing t_k to be greater than unity. We shall provide a simple line search procedure suitable for the present problem in the next section.

Now we consider the convergence properties of the proposed algorithm. The following proposition states that the algorithm always terminates after generating a point which can be regarded as an approximate optimal solution to problem (14), whenever certain boundedness conditions are met.

Proposition 6. Suppose that the function F is bounded from below and the positive definite matrices B_k are uniformly bounded, i.e., there exists a $\lambda > 0$ such that

$$d^T B_k d \leq \lambda \|d\|^2 \qquad \forall d , \ \forall k . \tag{25}$$

Then, the algorithm terminates in a finite number of iterations and the last iterate u^k satisfies the following approximate optimality condition

$$F(u^k) \leq F(u) + \sqrt{2\lambda\varepsilon}\,\|u - u^k\| + \varepsilon \qquad \forall u . \tag{26}$$

Proof: Suppose that the algorithm lasts infinitely, i.e., the condition (19) is never achieved for any k and i. Then, since the inner iteration terminates finitely for each k by Proposition 4, we must have an infinite sequence $\{u^k\}$. Noting that B_k are positive definite and $t_k \geq 1$, however, we obtain from (21) and (23)

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$F(u^{k+1}) \leq F(u^k) - m\varepsilon(1-\beta) \qquad \forall k ,$$

which in turn yields

$$F(u^{k+1}) \leq F(u^1) - km\varepsilon(1-\beta) \qquad \forall k .$$

This implies $F(u^k)$ tends to $-\infty$ as $k \to \infty$ and contradicts the assumption that F is bounded from below. Thus the algorithm must terminate finitely.

Now let u^k be the last iterate generated by the algorithm. Define the function Φ_k by

$$\Phi_k(z) = \frac{\lambda}{2} \|z - u^k\|^2 + F(z) .$$

Obviously, we have

$$\Phi_k(u^k) = F(u^k) \tag{27}$$

and, by (25),

$$\phi_k(z) \leq \Phi_k(z) \qquad \forall z . \tag{28}$$

Moreover, since $\bar{\psi}_{k,i}$ is a lower bound of the optimal value of problem (16), we have

$$\bar{\psi}_{k,i} \leq \phi_k(z) \qquad \forall z . \tag{29}$$

It then follows from (27),(28),(29) and (19) that

$$\Phi_k(u^k) \leq \Phi_k(z) + \varepsilon \qquad \forall z . \tag{30}$$

Using the notion of ε-subdifferential [27,p.219], we have an equivalent representation of (30)

$$0 \in \partial_\varepsilon \Phi_k(u^k) \tag{31}$$

[4,Prop.1], where $\partial_\varepsilon \Phi_k(u^k)$ is the ε-subdifferential of Φ_k at u^k, i.e.,

$$\partial_\varepsilon \Phi_k(u^k) = \{ \xi \mid \Phi_k(u) \geq \Phi_k(u^k) + \xi^T(u - u^k) - \varepsilon , \ \forall u \} .$$

Furthermore, applying a formula for the ε-subdifferential of the sum of convex functions [17,Thm.2.1] to the function Φ_k, we have

$$\partial_\varepsilon \Phi_k(u^k) \subset \sqrt{2\lambda\varepsilon}\, B + \partial_\varepsilon F(u^k) , \tag{32}$$

where B is the closed unit sphere. Combining (31) and (32), we may conclude that there exists a $\xi \in \partial_\varepsilon F(u^k)$ such that $\|\xi\| \leq \sqrt{2\lambda\varepsilon}$. Consequently, by the definition of ε-subdifferential, we obtain for any u

$$\begin{aligned} F(u) &\geq F(u^k) + \xi^T(u - u^k) - \varepsilon \\ &\geq F(u^k) - \|\xi\| \, \|u - u^k\| - \varepsilon \end{aligned}$$

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$\geq F(u^k) - \sqrt{2\lambda\varepsilon}\,\|u - u^k\| - \varepsilon \ . \qquad \square$$

Note that the boundedness assumption on F is satisfied under the hypothesis of Proposition 3, which is normally fulfilled in practice. Also we mention that the approximate optimality condition (26) has been considered in [11,21].

To conclude this section, we should mention that the number of inner iterations required to achieve (20) usually increases as the major iteration proceeds. Unfortunately, we have been unable to establish any theoretical bound on the number of inner iterations with ε, β and k given. In our preliminary computational experiments reported in Section 6, however, the condition (20) was always attained before subproblem (18) became intractably large.

5. Remarks on the Algorithm

In this section, we provide some guidelines concerning the implementation of the proposed algorithm.

5.1. Evaluation of subgradients of F

In order to construct subproblems (18), it is required to supply a subroutine that calculates, for any feasible point of problem (14), a subgradient of the function F defined by (15). It is noted that the functions f_a^* are continuously differentiable (Proposition 1) and their gradients are in general available explicitly. Therefore, it would be sufficient to show how we can obtain a subgradient of the concave function h defined by

$$h(u) = \sum_{k\in K} D_k \min_{p\in P_k} \sum_{a\in A} u_a \delta_{ap} \ .$$

As was pointed out in Section 3, $h(u)$ can be evaluated at each u by solving the shortest chain problem (9) for each $k \in K$. Note that the arc lengths u_a are all nonnegative, because we are exclusively concerned with feasible points of problem (14). Denote

$$P_k(u) = \{\, p \in P_k \mid \sum_{a\in A} u_a \delta_{ap} \leq \sum_{a\in A} u_a \delta_{aq} \ , \ \forall q \in P_k \,\}$$

i.e., $P_k(u)$ is the set of all shortest chains for commodity k when arc lengths are given by u_a, $a \in A$. Then, it can be shown [12] that, for any $p_k \in P_k(u)$, the vector with elements

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$\sum_{k\in K} D_k \delta_{ap_k} , \qquad a \in A$$

affords a subgradient of h at u. Thus, a subgradient of h can be readily obtained as a byproduct of evaluating the functional value of h.

Consequently, a subgradient of the function F at u is given as the vector whose elements are

$$\nabla f_a^*(u_a) - \sum_{k\in K} D_k \delta_{ap_k} , \qquad a \in A ,$$

where $p_k \in P_k(u)$.

5.2. Choice of matrices B_k

In view of problem (18), it seems quite reasonable to choose matrices B_k so as to take account of second order information on F. Since the second term of (15) is piece-wise linear in u, we confine ourselves to information obtained from the first term $\Sigma f_a^*(u_a)$. By proposition 1, each f_a^* is continuously differentiable at any u_a. Moreover, it happens in most applications that f_a^* actually has the second derivatives.

Assume the existence of $\nabla^2 f_a^*(u_a)$ for each u_a. Then we may consider it appropriate to define B_k as a diagonal matrix with diagonal elements $\nabla^2 f_a^*(u_a^k)$. (If $\nabla^2 f_a^*(u_a)$ does not exists or is difficult to calculate explicitly, we may replace it by a finite difference approximation obtained from the first derivatives of f_a^*.) Note that $\nabla^2 f_a^*(u_a^k)$ is never negative thanks to the convexity of f_a^*. However, since it may happen that $\nabla^2 f_a^*$ take value zero or become unbounded, it is necessary to modify the definition of B_k as

$$B_k = \mathrm{diag}(b_a^k) , \tag{33}$$

where b_a^k are given by

$$b_a^k = \begin{cases} m_a & \text{if} \quad \nabla^2 f_a^*(u_a^k) < m_a \\ \nabla^2 f_a^*(u_a^k) & \text{if} \quad m_a \le \nabla^2 f_a^*(u_a^k) \le M_a \\ M_a & \text{if} \quad \nabla^2 f_a^*(u_a^k) > M_a \end{cases} \tag{34}$$

for some $0 < m_a < M_a$. By this modification, the matrices B_k are assured to be bounded and positive definite, and hence the hypothesis of Proposition 6 is fulfilled.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

5.3. Solving subproblems (18)

At each major iteration, we are required to solve a sequence of convex quadratic programming problems (18). Although those problems may be treated in a direct manner by any quadratic programming algorithm, we consider it more effective to transform problems (18) into problems of smaller size by means of a variable reduction technique and then solve the latter problems using a suitable optimization algorithm. It will become evident that this idea is remarkably useful when the network under consideration is large. In the development to follow, we suppose that B_k is a diagonal matrix as given by (33) and (34).

To simplify the notation, let us rewrite problem (18) as

$$\begin{aligned} (35) \qquad & \min \quad \frac{1}{2}(z-u^k)^T B_k (z-u^k) + \eta \\ & \text{s.t.} \quad G^T z - \eta e \le q \\ & \qquad\qquad z \ge \alpha \ , \end{aligned}$$

where $G = (g^0,\ldots,g^{i-1})$, $q = ((g^0)^T z^0 - F(z^0),\ldots,(g^{i-1})^T z^{i-1} - F(z^{i-1}))^T$ and $e = (1,\ldots,1)^T$. Then, the Lagrangian associated with problem (35) is defined to be

$$\theta(z,\eta,\mu,\nu) = \frac{1}{2}(z-u^k)^T B_k (z-u^k) + \eta + \mu^T(G^T z - \eta e - q) + \nu^T(\alpha - z) \ ,$$

where μ and ν are the vectors of Lagrange multipliers. Since B_k is positive definite, problem (35) has a dual which is given by direct calculation as

$$\begin{aligned} (36) \qquad & \max \quad \theta(\mu,\nu) \\ & \text{s.t.} \quad e^T\mu = 1 \ , \quad \mu \ge 0 \ , \quad \nu \ge 0 \ , \end{aligned}$$

where θ is a concave quadratic function defined by

$$(37) \qquad \theta(\mu,\nu) = \nu^T(\alpha - u^k) + \mu^T(G^T u^k - q) - \frac{1}{2}(\nu - G\mu)^T B_k^{-1}(\nu - G\mu) \ .$$

Moreover, it is not difficult to observe that, if (μ,ν) solves problem (36), then the point z given by

$$(38) \qquad z = u^k + B_k^{-1}(\nu - G\mu)$$

constitutes an optimal solution of problem (35).

The quadratic programming problem (36) can be solved in two stages: We first maximize $\theta(\mu,\nu)$ with respect to ν for fixed μ , and then maximize the resulting function with respect to μ . As B_k^{-1} is also positive

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

definite and diagonal, the first stage of the above procedure can easily be carried out and yields

$$\nu(\mu) = \max\{\, 0 \,,\, G\mu - B_k(u^k - \alpha) \,\} \tag{39}$$

as the maximizer of $\theta(\mu,\nu)$ with respect to ν. Substituting (39) into (37), we may explicitly state the second stage problem as

$$\begin{aligned} &\max && \Psi(\mu) \\ &\text{s.t.} && e^T\mu = 1 \,, \quad \mu \geq 0 \,, \end{aligned} \tag{40}$$

where Ψ is defined by

$$\Psi(\mu) = \mu^T(G^T u^k - q) - \frac{1}{2}\{ G\mu - \nu(\mu) \}^T B_k^{-1} \{ G\mu + \nu(\mu) \} \,.$$

In addition, it can be shown that Ψ is a concave function and has a continuous first derivative explicitly given by

$$\nabla\Psi(\mu) = G^T u^k - q - G^T B_k^{-1}(G\mu - \nu(\mu)) \,.$$

Thus, problem (40) is a differentiable concave programming problem with a single equality constraint and the nonnegativity constraints.

It should be noted that the number of variables in problem (40) is just equal to the iteration count of the inner iteration. More important is the fact that the size of problem (40) is, unlike problems (35) and (36), free from the size of the underlying network. Therefore, the variable reduction process described above will be considerably meritorious when we are involved in large networks.

One thing which might obstruct the present approach is the fact that problem (40) is not a quadratic programming problem any more and hence we cannot expect to obtain its exact solution finitely. Our computational experience reveals, however, that a good approximate solution to problem (40) can be obtained quite easily and such approximation does not cause breakdown of the algorithm, provided a proper approximation criterion is used.

5.4. Line search

Although the step size of unity is sufficient for convergence of the algorithm as pointed out just after the proof of Proposition 5, looking for a step length $t_k \geq 1$ seems worth considering since it would afford a more satisfactory decrease in the objective value. However, since evaluating $F(u)$ amounts to solving shortest chain problems (9) for all commodities, line search algorithms that require a large number of functional evaluations

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

are not recommended. Bearing this in mind, we present a very simple line search procedure which appears to be suitable for our problem.

Recall that the directional derivative of the convex function F at u with respect to a direction d may be characterized by

$$F'(u;d) = \sup \{ g^T d \mid g \in \partial F(u) \}$$

[27,Thm.23.4]. Therefore, for any $g \in \partial F(u)$, the quantity $\gamma = g^T d$ affords a lower bound of $F'(u;d)$ and, in particular, it coincides with $F'(u;d)$ whenever F is differentiable at u.

Given the current estimate u^k of the optimal solution and the search direction d^k, the line search algorithm proceeds as follows: First we take $t = 1$ and calculate $\gamma_1 = g^T d^k$ for an arbitrary $g \in \partial F(u^k + d^k)$. If $\gamma_1 \geq 0$, then we set $t_k = 1$ and terminate the line search procedure, because $\gamma_1 \geq 0$ implies $F(u^k + td^k)$ is nondecreasing for $t > 1$. If $\gamma_1 < 0$, then we calculate the maximum step length t_{max} satisfying the condition (22), i.e.,

$$t_{max} = \sup \{ t \mid u^k + td^k \geq \alpha \}.$$

Note that $t_{max} \geq 1$ by the construction of d^k. There are two possibilities for the values of t_{max}. That is, (i) $t_{max} < +\infty$ and (ii) $t_{max} = +\infty$.

In case (i), we calculate $\gamma_2 = g^T d^k$ for an arbitrary $g \in \partial F(u^k + t_{max} d^k)$. If $\gamma_2 \leq 0$, we may conclude that $F(u^k + td^k)$ is monotonically nonincreasing in the interval $[1, t_{max}]$. Therefore, we set $t_k = t_{max}$ if t_{max} satisfies condition (21), and set $t_k = 1$ otherwise. Note that condition (21) is normally met by t_{max} as long as the parameter m is set sufficiently small. On the other hand, when $\gamma_2 > 0$, a minimizer of $F(u^k + td^k)$ lies in the interval $[1, t_{max}]$. We may obtain an approximate minimizer, say $\tilde{t}$, by carrying out a single step of the cubic interpolation [24,p.142] using the functional values $F(u^k + d^k)$, $F(u^k + t_{max} d^k)$ and the (approximate) directional derivatives γ_1, γ_2 at $t = 1$ and $t = t_{max}$, respectively. We then set $t_k = \tilde{t}$ if $\tilde{t}$ satisfies condition (21), and set $t_k = 1$ otherwise. Again, we usually have $t_k = \tilde{t}$, since condition (21) is not very restrictive for such $\tilde{t}$ in practice.

In case (ii), we select some $t_c > 1$ and calculate $F(u^k + t_c d^k)$ and $\gamma_c = g^T d^k$, where $g \in \partial F(u^k + t_c d^k)$. Based upon these values together with those at $t = 1$, we carry out a single step of the cubic interpolation to obtain an approximate minimizer $\tilde{t}$ of $F(u^k + td^k)$. Then, as in the previous case, the actual step length is determined as $t_k = \tilde{t}$ or $t_k = 1$, depending on whether $\tilde{t}$ satisfies condition (20) or not.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

The above procedure can be implemented very easily and the generated step length t_k obviously satisfies the line search criteria (21) and (22).

5.5. Choice of the parameters ε, β and m

The parameters ε, β and m are supposed to be fixed throughout the iterations, though it is possible to modify the algorithm in such a way that those parameters vary on every iteration without impairing the convergence properties.

As shown by (26), ε is a tolerance parameter which indicates how close the approximate solution u^k obtained by the algorithm is to the true optimum. So, ε should be set small enough to meet the user's requirement on the accuracy of the approximate solution.

The parameter β used in the termination criterion of the inner iterations seems to have the greatest influence on the performance of the algorithm. In view of (20), it is clear that assigning a large value to β results in a small number of inner iterations. Therefore, it is recommended to set β sufficiently close to one, say $\beta = 0.99999$, because spending too much time in the inner iterations is obviously undesirable for the overall performance of the algorithm.

The parameter m appearing in the line search criterion (21) plays a relatively minor role in the algorithm. We suggest that a sufficiently small m, say $m = 0.1$, may be suitable for the line search procedure described in 5.4.

6. Computational Results

In order to measure the efficiency of the proposed algorithm, limited computational experiments have been conducted on several traffic assignment test problems. In the traffic assignment models, the commodities correspond to prespecified O-D pairs for which nonzero trip demands exist. For example, if the transportation network has m arcs and n nodes, and there are trip demands for all O-D pairs (i,j), $i, j = 1,\ldots,n$, $i \neq j$, then the problem involves $n(n-1)$ commodities. Thus, the traffic assignment problem as formulated by (1) contains $mn(n-1)$ variables. It is noted, however, that the dual problem (10) contains m variables which, as indicated at the end of Section 3, represent travel costs through the arcs.

In the test problems used in our experiments, the arc cost functions f_a

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

are assumed to be of the form

$$f_a(x_a) = \begin{cases} \alpha_a x_a + \dfrac{\beta_a}{5} x_a^5 & \text{if} \quad x_a \geq 0 \\ +\infty & \text{otherwise} , \end{cases}$$

where α_a and β_a are given positive parameters. Obviously, the right derivative of f_a at $x_a = 0$ is α_a. The convex conjugate functions f_a^* are directly calculated as

$$f_a^*(u_a) = \begin{cases} \dfrac{4\beta_a}{5} \left(\dfrac{u_a - \alpha_a}{\beta_a} \right)^{5/4} & \text{if} \quad u_a \geq \alpha_a \\ 0 & \text{otherwise} . \end{cases}$$

It is easily seen that f_a^* are twice differentiable for $u_a > \alpha_a$, but $\nabla^2 f_a^*(u_a)$ tends to $+\infty$ as $u_a \downarrow \alpha_a$.

The computer program was coded in FORTRAN 77 and the runs were made on a FACOM M382 computer at the Data Processing Center, Kyoto University. To solve subproblems (40), we used a subroutine LCQNDR in the NPL Numerical Optimization Software Library developed by the National Physical Laboratory, England. Also, we employed the Warshall-Floyd algorithm [7] for finding shortest chains between all pairs of nodes to solve problems (9). The parameters β and m were set at 0.99999 and 0.1, respectively. The lower and the upper bounds on the diagonal elements of B_k (cf. (34)) were specified as $m_a = 200$ and $M_a = 20{,}000$, respectively, for all $a \in A$.

First, we applied the algorithm to six fictitious test problems whose data D_k, α_a and β_a were chosen in a random but plausible manner so as to imitate actual traffic situations. Each problem was solved with two different values of ε, i.e., $\varepsilon = 0.1$ and $\varepsilon = 0.05$. The size of each problem and the experimental results are summarized in Table. Table shows that the CPU times are roughly proportional to NSUB and NFUN. This is implied by the fact that most of the computational effort is devoted to solving subproblems (40) and shortest chain problems (9). We also mention that, as far as these numerical examples are concerned, we never encountered a case in which a single major iteration involved more than fifteen inner iterations. Although the number of inner iterations to be performed in a single major iteration was gradually increasing as the major iteration proceeded, this phenomenon will not impede the use of the algorithm unless we insist on obtaining a very accurate approximate optimal solution to the problem.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Table. Problem data and computational results

Problem	# nodes	# arcs	# commodities	ε	ITER	NSUB	NFUN	CPU sec.
1	7	20	42	0.1	6	7	8	0.059
				0.05	6	8	9	0.062
2	7	20	42	0.1	5	6	6	0.054
				0.05	6	8	8	0.062
3	12	34	132	0.1	10	47	55	0.527
				0.05	13	87	97	1.077
4	12	34	132	0.1	14	46	54	0.426
				0.05	18	87	97	0.732
5	16	25	240	0.1	9	20	28	0.207
				0.05	14	38	50	0.385
6	16	25	240	0.1	28	51	79	0.465
				0.05	65	162	190	1.339

ITER: Number of major iterations.
NSUB: Number of subproblems (40) solved.
NFUN: Number of function evaluations, i.e., number of shortest chain problems (9) solved.
N.B. NSUB is equal to the total number of inner iterations.

In order to examine the algorithm further, we also solved the test problem[†] given in [20]. The problem contains 24 nodes, 76 arcs and 552 commodities. The behavior of the algorithm applied to this problem is illustrated in Figure 2. It is noted that the vertical axis in Figure 2 represents the objective value of the original dual problem (10) rather than problem (14). Therefore, the objective value corresponding the generated sequence $\{u^k\}$ is monotonically increasing. It is seen from Figure 2 that the algorithm converges very rapidly in particular at an early stage of the iterations. The same problem was solved by the Frank-Wolfe algorithm [20], which is known as one of the most effective methods for nonlinear multicommodity flow problems, and the computational result is also illustrated in Figure 2. Since a single iteration of the Frank-Wolfe algorithm consists of solving a shortest chain problem to find a search direction and performing an exact line search procedure, it is roughly comparable to a single inner iteration of the proposed algorithm. For accuracy purposes, however, the results presented in Figure 2 are based on the CPU time. We have also compared the solutions obtained after 100 iterations of the Frank-

[†]In [20], the trip table is erroneously labeled as thousands of vehicles per day. The correct units are hundreds of vehicles per day.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

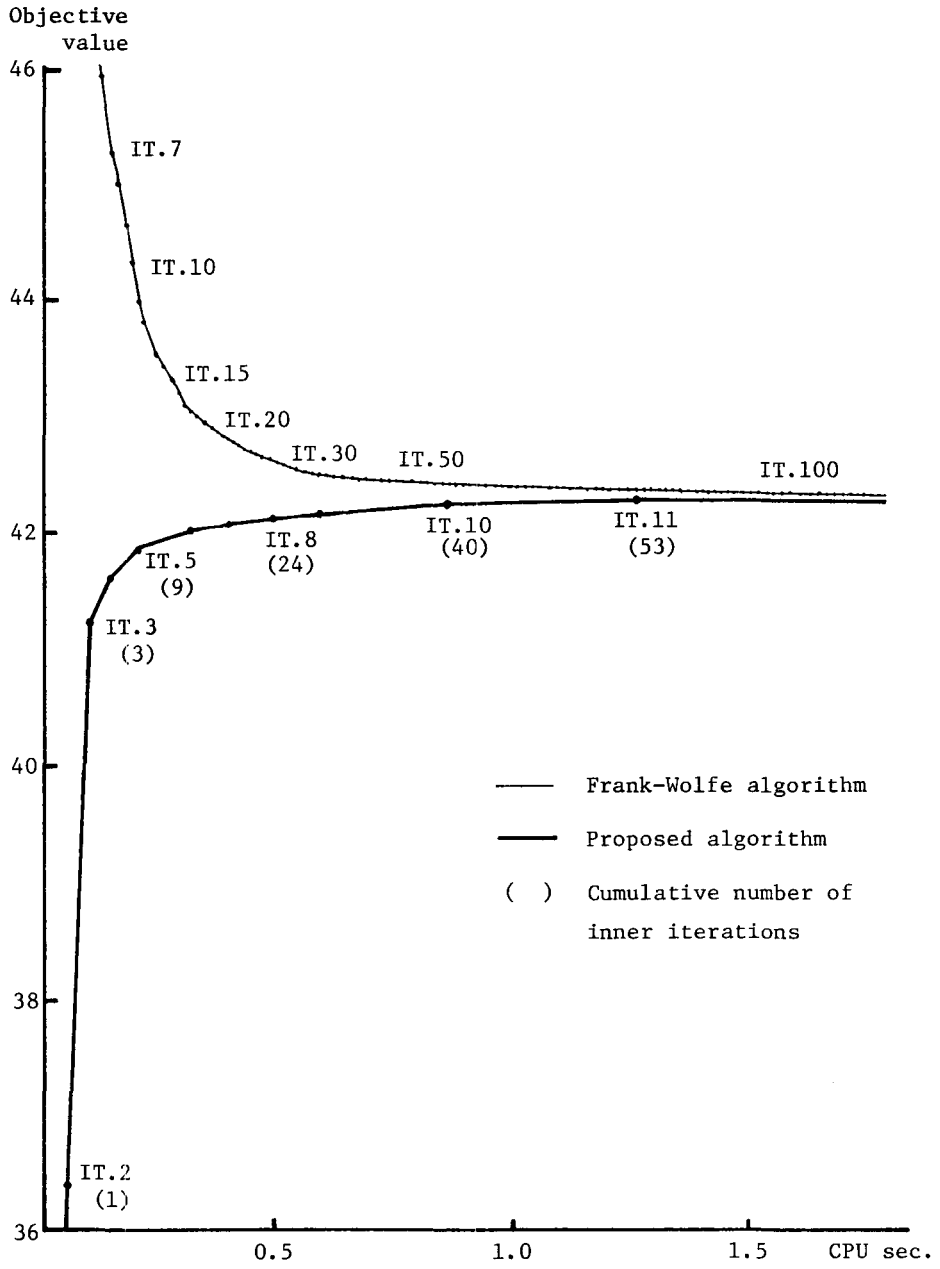


Figure 2. Comparison of the proposed algorithm with the Frank-Wolfe algorithm

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Wolfe algorithm with those obtained after 11 major iterations of the proposed algorithm. (Note that 100 Frank-Wolfe iterations are approximately equal in CPU time to 11 major iterations of the proposed algorithm.) In the comparison, the solutions obtained by the proposed algorithm were transformed into flow values using the relation (13), and we found that there was no difference between the two solutions within three significant figures for most of the variables. Consequently, we may conclude that the proposed algorithm performed at least as efficiently as the Frank-Wolfe algorithm, as far as this test problem is concerned.

7. Conclusion

We have presented a nonsmooth optimization approach to solving the nonlinear multicommodity flow problems. Our computational experience is very limited but indicates that the proposed algorithm may be quite effective when applied to actual problems. This argument may also be supported by the fact that the main constituents of the proposed algorithm are to solve shortest chain problems (9) and to solve subproblems (40) successively. Indeed, it is well appreciated that the shortest chain problems with hundreds of nodes can be efficiently solved. In addition, the size of subproblems (40) is independent of the underlying network structure. Therefore, we may expect the proposed algorithm to be able to deal with fairly large problems.

Acknowledgment

The author is indebted to Professors H. Mine, T. Hasegawa and K. Ohno for their helpful comments and support.

References

[1] Assad, A.A.: Multicommodity Network Flows - A Survey. *Networks*, Vol.8 (1978), 37-91.

[2] Beckmann, M., McGuire, C.B. and Winsten, C.B.: *Studies in the Economics of Transportation*. Yale University Press, New Haven, Conn., 1965.

[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, 1979, 210-224.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

[4] Bertsekas, D.P. and Mitter, S.K.: A Descent Numerical Method for Optimization Problems with Nondifferentiable Cost Functionals. *SIAM Journal on Control*, Vol.11 (1973), 637-652.

[5] Cantor, D.G. and Gerla, M.: Optimal Routing in a Packet-Switched Computer Network. *IEEE Transactions on Computers*, Vol.C-23 (1974), 1062-1069.

[6] Dafermos, S.C. and Sparrow, F.T.: The Traffic Assignment Problem for a General Network. *Journal of Research of National Bureau of Standards*, Vol.73B (1969), 91-118.

[7] Dreyfus, S.E.: An Appraisal of Some Shortest-Path Algorithms. *Operations Research*, Vol.17 (1969), 395-412.

[8] Florian, M.: An Improved Linear Approximation Algorithm for the Network Equilibrium (Packet Switching) Problem. *Proceedings of the 1977 IEEE Conference on Decision & Control*, 1977, 812-818.

[9] Ford, L.R., Jr. and Fulkerson, D.R.: A Suggested Computation for Maximal Multicommodity Network Flow. *Management Science*, Vol.5 (1958), 97-101.

[10] Fratta, L., Gerla, M. and Kleinrock, L.: The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design. *Networks*, Vol.3 (1973), 97-133.

[11] Fukushima, M.: A Descent Algorithm for Nonsmooth Convex Optimization. *Mathematical Programming*, to be published.

[12] Fukushima, M.: On the Dual Approach to the Traffic Assignment Problem. *Transportation Research*, to be published.

[13] Fukushima, M.: A Modified Frank-Wolfe Algorithm for Solving the Traffic Assignment Problem. *Transportation Research*, to be published.

[14] Gallager, R.G.: A Minimum Delay Routing Algorithm Using Distributed Computation. *IEEE Transactions on Communications*, Vol.COM-25 (1977), 73-85.

[15] Hall, M.A. and Peterson, E.L.: Traffic Equilibria Analysed via Geometric Programming. *Traffic Equilibrium Methods* (ed. M. Florian). Springer, New York, 1976, 53-105.

[16] Held, M., Wolfe, P. and Crowder, H.P.: Validation of Subgradient Optimization. *Mathematical Programming*, Vol.6 (1974), 62-88.

[17] Hiriart-Urruty, J.-B.: ε-subdifferential Calculus. *Convex Analysis and Optimization* (eds. J.-P. Aubin and R.B. Vinter). Pitman, London, 1982, 43-92.

[18] Kennington, J.L.: A Survey of Linear Cost Multicommodity Network Flows. *Operations Research*, Vol.26 (1978), 209-236.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

[19] Kennington, J.L. and Shalaby, M.: An Effective Subgradient Procedure for Minimal Cost Multicommodity Flow Problem. *Management Science*, Vol.23 (1977), 994-1004.

[20] 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.

[21] Lemarechal, C.: Nonsmooth Optimization and Descent Methods. RR-78-4, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

[22] Lemarechal, C. and Mifflin, R. (eds.): *Nonsmooth Optimization*. Pergamon Press, Oxford, 1978.

[23] Leventhal, T., Nemhauser, G. and Trotter, L., Jr.: A Column Generation Algorithm for Optimal Traffic Assignment. *Transportation Science*, Vol.7 (1973), 168-176.

[24] Luenberger, D.G.: *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, Mass., 1973.

[25] Nurminski, E.A. (ed.): *Progress in Nondifferentiable Optimization*. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1982.

[26] Petersen, E.R.: A Primal-Dual Traffic Assignment Algorithm. *Management Science*, Vol.22 (1975), 87-95.

[27] Rockafellar, R.T.: *Convex Analysis*. Princeton University Press, Princeton, N.J., 1970.

[28] Rockafellar, R.T.: Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, Vol.14 (1976), 877-898.

[29] Tomlin, J.A.: Minimum-Cost Multicommodity Network Flows. *Operations Research*, Vol.14 (1966), 45-51.

[30] Wardrop, J.G.: Some Theoretical Aspects of Road Traffic Research. *Proceedings of the Institute of Civil Engineers*, Part II, Vol.1 (1952), 325-378.

Masao FUKUSHIMA: Department of
Applied Mathematics and Physics,
Faculty of Engineering,
Kyoto University, Kyoto 606, Japan

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.