A polynomial algorithm for minimum quadratic cost flow problems *

M. MINOUX
Applied Mathematics Department, CNET PAA/TIM, 38-40 rue du Général Leclerc, 92131 Issy, France

Abstract. Network flow problems with quadratic separable costs appear in a number of important applications such as: approximating input-output matrices in economy; projecting and forecasting traffic matrices in telecommunication networks; solving nondifferentiable cost flow problems by subgradient algorithms. It is shown that the scaling technique introduced by Edmonds and Karp (1972) in the case of linear cost flows for deriving a polynomial complexity bound for the out-of-kilter method, may be extended to quadratic cost flows and leads to a polynomial algorithm for this class of problems. The method may be applied to the solution of singly constrained quadratic programs and thus provides an alternative approach to the polynomial algorithm suggested by Helgason, Kennington and Lall (1980).

Keywords: Network flows, quadratic programming, combinatorial optimization, complexity of algorithms, least squares approximation

* Paper presented at "Netflow 83", Pisa, Italy, March 1983.

An anonymous referee is gratefully acknowledged for his constructive remarks and suggestions. Special thanks are due to Georges Bartnik and Henri Juin for performing the computational experiments reported in this paper.

Received April 1983; revised December 1983

North-Holland
European Journal of Operational Research 18 (1984) 377-387

1. Introduction

Minimum cost flow problems with convex cost functions provide a natural extension of the classical network flow theory (cf. Ford and Fulkerson, 1962) and arise in a number of important applications such as the search for a traffic equilibrium in communication networks (e.g. urban traffic, see Nguyen, 1974, Leblanc et al., 1975), routing problems in computer communication networks (cf. Cantor and Gerla, 1974).

For this class of problems, a number of solution methods have been suggested, among which linearization techniques (Kennington and Helgason, 1980), the specialization of the Frank-Wolfe method (Cantor and Gerla, 1974), the use of Newton or quasi-Newton methods (Dembo and Klincewicz, 1978).

We focus here on the particular case where the cost function is quadratic and separable, i.e. of the form

$$\sum_{u \in U} \tfrac{1}{2} w_u (\varphi_u - \bar{\varphi}_u)^2$$

where U is the set of arcs of the graph, φ_u is the flow value on arc u ($u \in U$) and where the weights w_u as well as the $\bar{\varphi}_u$ values ($u \in U$) are given.

In the special case where the graph is *bipartite* the problem may be viewed as an extension of the well-known *transportation problem* to the case of quadratic separable cost functions. This type of problem appears, for instance, when forecasting or approximation input-output matrices in economy, or traffic matrices in communication or telecommunication networks. What is required then is to determine a matrix with given (prescribed) row and column sums, and best approximating (in the least squares sense) a given matrix (cf. Bacharach, 1965; Minoux, 1978; Debiesse and Matignon, 1980).

Transportation problems with quadratic separable costs have been studied by a number of authors, and various solution algorithms have been suggested:

– block successive relaxation technique (Cottle and Pang, 1981),

– successive projection methods (Bachem and Korte, 1980),

– specialization of the complementary pivot algorithm (cf. Wolfe, 1959; Lemke, 1962; Jaumard and Minoux, 1982).

It should be noted that, for none of the algorithms mentioned above, a polynomial complexity bound could be derived. However, quadratic cost flow problems belong to the class of linearly constrained quadratic cost programming problems which, as shown by Kozlov et al. (1979), can be solved polynomially by Khachian's algorithm (cf. Khachian, 1979).

However, if Khachian's algorithm indeed provides a very powerful tool to settle the *existence* of a polynomial algorithm for many problems (see the nice review paper from Grötschel, Lovâsz and Schrijver, 1981), the various computational experiments performed so far seem rather disappointing (for a general view, cf. Bland, Goldfarb and Todd, 1981) and leave very little hope for practical solution of high-dimensional problems by this method. Therefore, the search for other polynomial algorithms, possibly more efficient than Khachian's, is a major issue, from both theoretical and practical points of views.

The purpose of this paper is to show that the scaling technique introduced by Edmonds and Karp (1972) in the case of linear cost flows to obtain a polynomial complexity bound for the out-of-kilter method (see, for instance Ford and Fulkerson, 1962 or Gondran and Minoux, 1979, Chapter 5 section 4.1.) may be extended to the case of quadratic separable cost functions and lead to a polynomial algorithm for this class of problems.

In essence, the method consists in approximating the cost functions by piecewise linear cost functions and iteratively refining the approximation until a point, sufficiently close to the desired optimal point, is obtained. The polynomiality of the algorithm is then derived by using a well-known property of quadratic programs, namely that an optimum solution necessarily corresponds to a basic solution of the (linear) system resulting from the Kuhn–Tucker conditions: The minimal distance between two basic solutions (which is inversely proportional to the largest possible determinant of a basis) then defines the sufficient approximation to locate the optimum solution, and it is shown in section 5 that this can be achieved in a number of iterations bounded by a polynomial in the problem size (measured by the length of a binary encoding of the problem data: graph structure, costs and capacities).

It is interesting to notice that the algorithm described here may be applied, in particular, to the problem of minimizing a quadratic separable cost function under a single linear constraint (a generalization of the well-known 'knapsack' problem) and thus provides an alternative to the method suggested by Helgason et al. (1980).

2. Nonlinear separable cost network flows and kilter diagrams

We denote by $G=[X,U]$ the given graph, where X is the set of nodes ($|X|=N$), and U the set of arcs ($|U|=M$). To each arc $u\in U$, we associate:

– two real numbers b_u and c_u ($b_u\leqslant c_u$), respectively the lower and upper bound on the flow value φ_u on arc u;

– a real function $\gamma_u(\varphi_u)$ representing the cost of sending a flow φ_u on arc u.

In the following, we assume that the functions $\gamma_u(\varphi_u)$ are *convex* and *continuously differentiable* with respect to the φ_u variables, which is obviously true in the case of quadratic separable costs of the form

$$\gamma_u(\varphi_u)=\tfrac{1}{2}w_u(\varphi_u-\bar{\varphi}_u)^2,\quad w_u>0.$$

Now, the minimum cost flow problem on G can be written as:

$$\text{(P)}\quad \text{minimize}\quad \sum_{u\in U}\gamma_u(\varphi_u),$$

$$\text{subject to}$$

$$A\varphi=0, \qquad (1)$$

$$b_u\leqslant\varphi_u\leqslant c_u\quad(\forall u\in U), \qquad (2)$$

where $A=(a_{iu})$, $i=1,\dots,N$, $u=1,\dots,M$, is the node–arc incidence matrix of the graph ($a_{iu}=-1$ if i is the initial endpoint of arc u, $a_{iu}=+1$ if i is the terminal endpoint of arc u) and where $\varphi\in\mathbf{R}^M$ denotes the vector of components φ_u ($u\in U$).

Equations (1) express the conservation of the flows at each node, the inequalities (2) correspond to lower bounds and capacity constraints on each individual arc flow. The existence of a *feasible flow* (i.e. of a flow satisfying constraints (2)) will be assumed throughout though it should be noted that the algorithm presented here does not require prior knowledge of such a feasible flow. (For the conditions for a feasible flow to exist, see e.g. Ford and Fulkerson (1962).

We now start by giving an equivalent formulation for problem (P) which uses the concept of *kilter diagram* (see Lawler, 1976 or Gondran and Minoux, 1979 Chapter 5). In the case of convex continuously differentiable cost functions, the Kuhn–Tucker conditions are both necessary and sufficient for optimality, and denoting by $\pi = (\pi_i)$ ($i \in X$) the dual variables associated with constraints (1), by $\lambda = (\lambda_u)$ and $\mu = (\mu_u)$ the dual variables associated with constraints $\varphi_u \leqslant c_u$ and $-\varphi_u \leqslant -b_u$ respectively, they can be written as:

$$
\text{(KT)} \quad
\begin{aligned}
& A\varphi = 0, && (1)\\
& b \leqslant \varphi \leqslant c, && (2)\\
& \nabla\gamma(\varphi) - \pi A + \lambda T - \mu I = 0, && (3)\\
& \lambda_u(c_u - \varphi_u) = 0, && (4)\\
& \mu_u(\varphi_u - b_u) = 0, && (5)\\
& \pi \text{ has no sign constraints,} &&\\
& \lambda \geqslant 0,\ \mu \geqslant 0. &&
\end{aligned}
$$

In the above, $\nabla\gamma(\varphi)$ denotes the gradient of the cost function at φ, i.e. the M-vector of components $\nabla\gamma_u(\varphi_u) = \mathrm{d}\gamma_u(\varphi_u)/\mathrm{d}\varphi_u$. (4) and (5) are the so-called *complementary slackness conditions*. The variables π_i ($i \in X$) are usually referred to as the *potentials* associated with the nodes. Given a set of potentials π_i ($i \in X$), we denote by $\theta_u = \pi_j - \pi_i$ the *potential difference* or *tension* on arc $u = (i, j)$. In matrix form, the relation $\theta = \pi A$ holds.

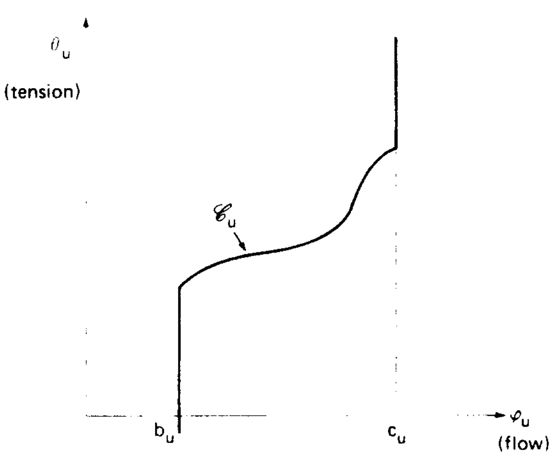

Fig. 1. Kilter diagram $\mathscr{C}_u$ associated with any arc $u \in U$.

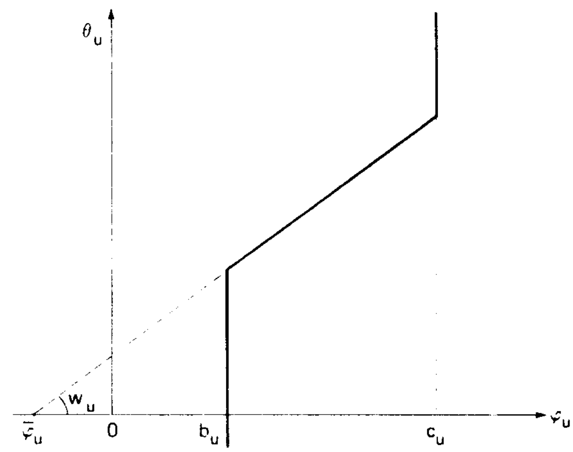

Fig. 2. Kilter diagram in the case of quadratic cost functions of the form $\sum \frac{1}{2} w_u(\varphi_u - \bar{\varphi}_u)^2$.

Now, for each arc $u \in U$, consider in $\mathbb{R}^2$ the subset $\mathscr{C}_u$ of points (φ_u, θ_u) called the *kilter diagram* and defined in the following way (cf. Fig. 1):

– $\mathscr{C}_u$ contains the graph of the function $\nabla\gamma_u(\varphi_u)$ on the interval $b_u \leqslant \varphi_u \leqslant c_u$;

– $\mathscr{C}_u$ contains the two half-lines defined as the set of points (φ_u, θ_u) such that $\varphi_u = b_u$ and $\theta_u \leqslant \nabla\gamma_u(b_u)$ on the one hand; $\varphi_u = c_u$ and $\theta_u \geqslant \nabla\gamma_u(c_u)$ on the other hand.

Since the $\gamma_u(\varphi_u)$ are convex continuously differentiable, we observe that the kilter diagrams are monotonic nondecreasing functions of the φ_u.

Also, it is easy to check that the Kuhn–Tucker conditions (KT) are satisfied by a pair of solutions φ and (π, λ, μ) respectively primal and dual feasible, if and only if, for each arc u, the point (φ_u, θ_u) belongs to the kilter diagram $\mathscr{C}_u$. Thus, problem (P) may be reformulated as: *Find a flow φ and a set of potentials π such that, for the associated tension $\theta = \pi A$, there holds, $\forall u \in U$, $(\varphi_u, \theta_u) \in \mathscr{C}_u$.* In the case of quadratic cost functions $\gamma_u(\varphi_u) = \frac{1}{2} w_u(\varphi_u - \bar{\varphi}_u)^2$, the kilter diagram takes the special form shown in Fig. 2. The part of the diagram of the abscissas ranging from b_u to c_u is an affine function of slope w_u and passing through the point $(\bar{\varphi}_u, 0)$.

3. Piecewise linear cost functions and properties of the out-of-kilter algorithm

Since the algorithm in Section 5 is based on piecewise linear approximations of the cost, we first address this class of problems and recall a number of basic properties of the out-of-kilter algorithm when applied to it.

In the case of piecewise linear convex cost functions, the kilter diagrams are of the form given in Fig. 3.

Let φ be a flow ($A\varphi = 0$) and π any set of potentials, $\theta = \pi A$ the associated tension. An arc $u \in U$ is said to be *in kilter* whenever $(\varphi_u, \theta_u) \in \mathscr{C}_u$. Otherwise, it is said to be *out-of-kilter*.

If all the arcs are in kilter, then (φ, π) is a pair of optimal primal and dual solutions, from the results of Section 2. Now, suppose that some arcs are not in kilter. For any arc $u \in U$, call the *kilter number* of arc u the quantity

$$\sigma_u(\varphi_u, \theta_u) = \min_{\tilde{\varphi}} \{ |\varphi_u - \tilde{\varphi}_u| \mid (\tilde{\varphi}_u, \theta_u) \in \mathscr{C}_u \},$$

in other words, the difference in abscissas between (φ_u, θ_u) and the closest point with the same θ_u value on the kilter diagram. Thus an arc is in kilter if and only if its kilter number is zero.

The out-of-kilter algorithm is an iterative procedure which, starting from any initial point (φ^0, θ^0), minimizes the function

$$\sigma(\varphi, \theta) = \sum_{u \in U} \sigma_u(\varphi_u, \theta_u)$$

(the sum of all kilter numbers on the arcs). We

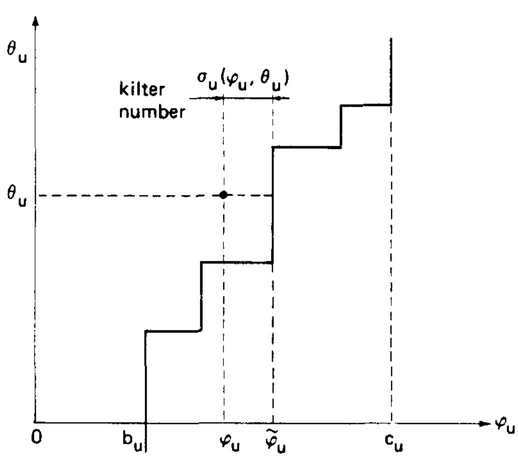

Fig. 3. Kilter diagram for a piecewise linear cost function and definition of the kilter number.

briefly review the main features and properties of the method which will be useful in the following (for a more detailed exposition, see Lawler, 1976, Gondran and Minoux, 1979, Chap. 5).

Each iteration consists in performing a flow change along a cycle in the graph (on the arcs whose orientation is compatible with the orientation of the cycle, the flow is increased by δ; on the other arcs, the flow is decreased by δ) passing through an out-of-kilter arc u_0. The transformation (and, in particular, the value δ) is chosen in such a way that no kilter number is ever increased on any arc (thus the arcs which have already been set in kilter remain in kilter).

It can be shown that the determination of such a cycle amounts to the search for an s–t path on the *incremental graph* where s and t are the endpoints of the out-of-kilter arc u_0. (For a precise definition of the concept of *incremental graph*, see Gondran and Minoux, Chapter 5). In practice, this can be achieved by a standard *labelling procedure*, the complexity of which is $O(M)$ where M is the number of arcs of the given graph G. In the situation where such a path does not exist it can be shown [1] that it is possible to perform a change in the potentials π_i ($i \in X$) such that:

- no kilter number is increased for any arc (in particular, in-kilter arcs remain in-kilter),
- by applying the labelling procedure on the incremental graph relative to the new solution (φ, θ') obtained, a new (previously unlabelled) node will become labelled (previously labelled nodes remain labelled).

Thus it is never necessary to perform more than N potential changes at each iteration, hence the complexity $O(M + N^2) \simeq O(N^2)$ for each iteration. (See e.g. Lawler, 1976).

Now, suppose that the capacity bounds b_u and c_u are *integer values* and that abscissas of the discontinuities in the kilter diagram are *integer values* (cf. Fig. 3). In this case, it is easily seen that:

(i) assuming the initial flow φ^0 integral in the starting solution (φ^0, θ^0) the integrality of the flows is maintained throughout the algorithm,

(ii) the sum of kilter numbers decreases at each iteration by at least one unit, and thus reaches 0 in a finite number of steps.

[1] Under the assumption in Section 2, namely that there exists a feasible flow.

From this, we deduce the following property, which is basic to the remainder of the paper:

Property 1. *Assume that the capacity bounds b_u and c_u are integer and that the discontinuities of the kilter diagram have integer abscissas. If the starting flow is integer valued, and if K (K integer) is the sum of kilter numbers for the starting solution, then the complexity of the out-of-kilter method is* $O(KN^2)$.

4. Approximate solution of the quadratic cost flow problem by successive approximations

It is shown here how a quadratic cost flow problem may be solved approximately by considering a sequence of piecewise linear convex approximations of the given convex cost functions. This may be considered as an extension to the quadratic case of the *scaling technique* suggested by Edmonds and Karp (1972) for linear cost flows.

Consider a quadratic cost flow problem of the form:

$$(\mathrm{P}) \quad \text{minimize} \quad \sum_{u \in U} \tfrac{1}{2} w_u (\varphi_u - \bar{\varphi}_u)^2,$$

subject to

$$A\varphi = 0, \tag{1}$$

$$b_u \leqslant \varphi_u \leqslant c_u \quad (\forall u \in U), \tag{2}$$

We shall assume that (P) has a solution and that all the w_u's are strictly positive. In this situation, the objective function being *strictly convex*, (P) has a *unique optimal solution* φ^*. Also note that it is by no means restrictive to assume that all the b_u, c_u and $\bar{\varphi}_u$ values are *integers* (if this is not the case, just multiply all the flows and capacity bounds by a large enough number). This assumption will be made throughout the rest of the paper.

Now define the *size of problem* (P) as the number of binary elements necessary to encode the problem, i.e.:

(i) The structure of the given graph (if an adjacency list is used—see for instance Gondran and Minoux (1979, Chapter 1)—$(M + N\,[\log_2 N]$ bits are necessary).

(ii) The values of the lower and upper capacity bounds b_u and c_u. Denoting

$$c_{\max} = \max\left\{ \max_{u \in U} \{|c_u|\},\ \max_{u \in U} \{|b_u|\} \right\},$$

$2M(1 + \lceil \log_2 c_{\max} \rceil)$ bits are necessary.
($\lceil\ \rceil$ denotes the least integer greater or equal than.)

(iii) The values w_u and $\bar{\varphi}_u$. Denoting

$$W = \max\left\{ \max_{u \in U} \{|w_u|\},\ \max_{u \in U} \{|\bar{\varphi}_u|\} \right\},$$

$2M(1 + \lceil \log_2 W \rceil)$ bits are necessary.

As a whole, the size of problem (P) will be

$$T = (M + N)\lceil \log_2 N \rceil + 2M(2 + \lceil \log_2 c_{\max} \rceil + \lceil \log_2 W \rceil).$$

Now we define the concept of pth order approximation of problem (P) (p integer $\geqslant 0$ or $\leqslant 0$) as the piecewise linear cost network flow problem (PA[p]) obtained in the following way.

For each arc $u \in U$ of G:

– the lower and upper capacity bounds $b_u^{(p)}$ and $c_u^{(p)}$ are given by

$$c_u^{(p)} = 2^p \left\lceil \frac{c_u}{2^p} \right\rceil, \qquad b_u^{(p)} = 2^p \left\lfloor \frac{b_u}{2^p} \right\rfloor$$

where $\lfloor\ \rfloor$ denotes the largest integer less or equal than;

– the discontinuities of the kilter diagram correspond to all abscissas $k2^p$ for integer k satisfying

$$\left\lfloor \frac{b_u}{2^p} \right\rfloor \leqslant k \leqslant \left\lceil \frac{c_u}{2^p} \right\rceil;$$

– the kilter diagrams of (P) and (PA[p]) coincide on all midpoints of the intervals $[k2^p, (k+1)2^p]$ for

$$\left\lfloor \frac{b_u}{2^p} \right\rfloor \leqslant k \leqslant \left\lceil \frac{c_u}{2^p} \right\rceil - 1.$$

The approximate problem (PA[p]) enjoys a number of properties with respect to (P), which are summarized below.

Property 2. *Let* (P) *be a quadratic cost flow problem, and* (PA[p]) *the corresponding p-th order approximation.*

(i) *if* (P) *has a feasible solution, then, for any integer p,* (PA[p]) *has a feasible solution,*

(ii) *if $\gamma_u^{(p)}(\varphi_u)$ denotes the cost function on arc u for problem* (PA[p]) *and $\gamma_u(\varphi_u) = \frac{1}{2} w_u (\varphi_u - \bar{\varphi}_u)^2$ the cost function for problem* (P), *then for all integer k,*

$$\gamma_u^{(p)}(k2^p) = \gamma_u(k2^p),$$

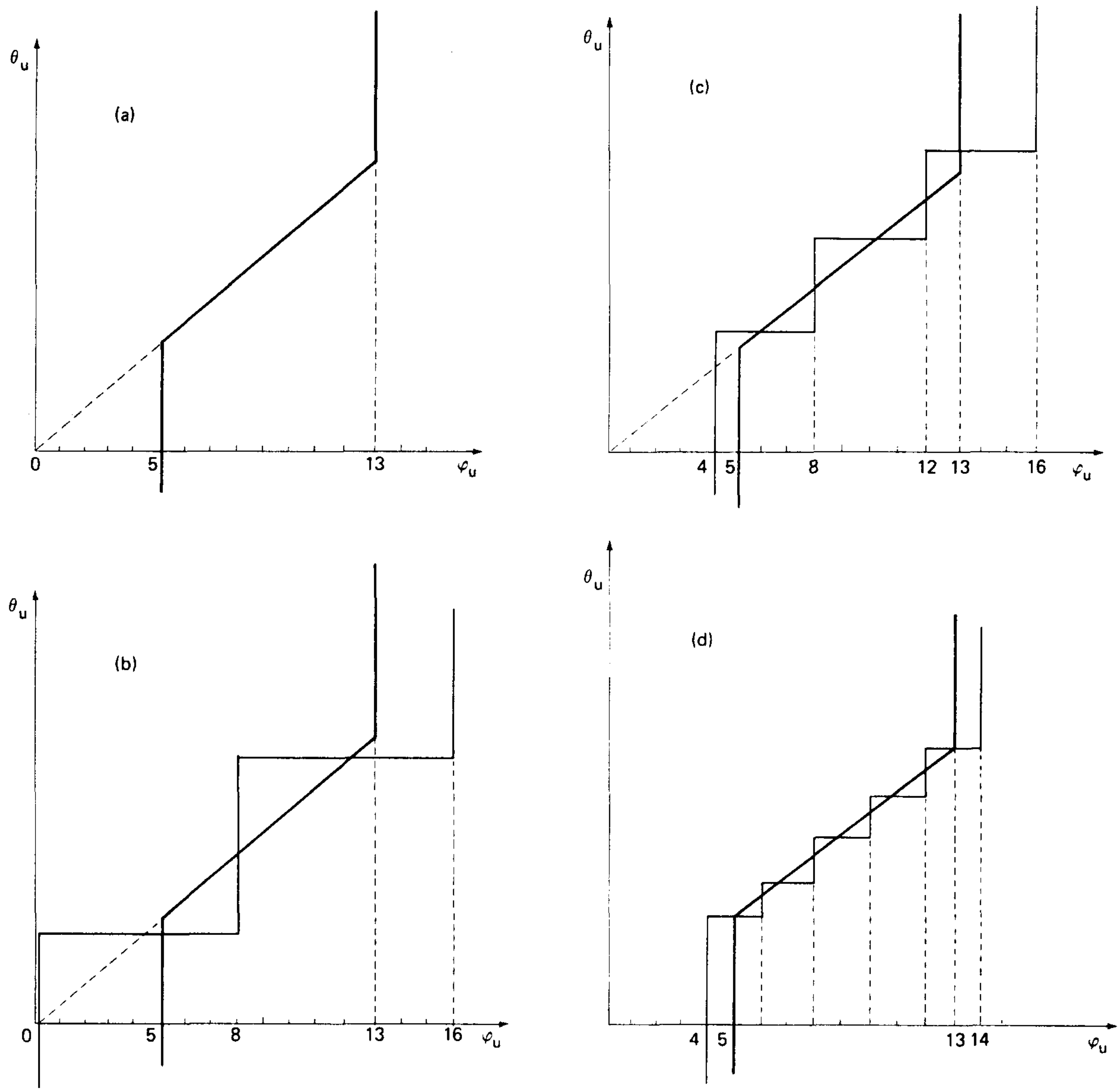

Fig. 4. The kilter diagram for a quadratic cost flow problem and various approximations by step functions of this kilter diagram.
(a) The given kilter diagram: $b_u = 5$, $c_u = 13$, ($w_u = 1$, $\bar{\varphi}_u = 0$).
(b) Approximation of order $p = 3$; $c^{(3)} = 16$, $b^{(3)} = 0$.
(c) Approximation of order $p = 2$; $c^{(2)} = 16$, $b^{(2)} = 4$.
(d) Approximation of order $p = 1$; $c^{(1)} = 14$, $b^{(1)} = 4$.

and for all φ_u,

$$\gamma_u(\varphi_u) + w_u 2^{p-3} \geqslant \gamma_u^{(p)}(\varphi_u) \geqslant \gamma_u(\varphi_u)$$
$$\geqslant \gamma_u^{(p)}(\varphi_u) - w_u 2^{p-3},$$

(iii) *for any* $p \leqslant 0$ (PA[p]) *has the same lower and upper capacity bounds than* (P). *In particular, for any* $p \leqslant 0$, *an optimal solution of* (PA[p]) *is a feasible solution of* (P).

Proof. (i) This is obvious observing that $b_u^{(p)} \leqslant b_u$ and $c_u^{(p)} \geqslant c_u$ for all p.

(ii) The relation $\gamma_u^{(p)}(k2^p) = \gamma_u(k2^p)$ for integer k follows from the definition of problem (PA[p]). The second relation is obtained by observing that, for a quadratic with second derivative $w_u > 0$, the maximum difference between the function and its linear interpolation on an interval of length δ is obtained at the middle point and its value is $w_u\delta^2/8$.

(iii) The b_u and c_u being integers, we have $b_u^{(p)} = b_u$ and $c_u^{(p)} = c_u$ as soon as $p \leqslant 0$. □

The following property concerns the behaviour of the out-of-kilter algorithm when applied to problems (PA[p]).

Property 3.

(i) *When the out-of-kilter algorithm is applied to problem* [PA[p]), *if all initial flow values on the arcs are integer multiples of* 2^p, *then the flow values remain integer multiples of* 2^p *throughout the iterations.*

(ii) *When an optimal solution of* (PA[p]) *is taken as starting solution for* (PA[$p-1$]), *then the initial value of the kilter number is at most* $M2^{p-1}$ *where* M *is the number of arcs.*

(iii) *The number of operations necessary to obtain an optimal solution of* (PA[$p-1$]) *from an optimal solution of* (PA[p]) *is at most* $O(MN^2)$.

(iv) *Let* $\bar{p}$ *be such that* $2^{\bar{p}-1} \leqslant c_{\max} \leqslant 2^{\bar{p}}$, *i.e.* $\bar{p} = \lceil \log_2 c_{\max} \rceil$. *Then the zero flow* $\varphi = 0$ *is a feasible solution of* (PA[$\bar{p}$]) *and the number of operations necessary to obtain an optimal solution of* (PA[$\bar{p}$]) *starting with* $\varphi = 0$, $\pi = 0$ *is at most* $O(MN^2)$.

Proof. (i) Divide all flow values and capacity bounds by 2^p. We thus get a minimum cost flow problem with integer values for the capacity bounds and for the breakpoints of the kilter diagram. The result then follows from the way the out-of-kilter method works (cf. Section 3).

(ii) Consider an optimal solution (φ, θ) of (PA[p]). For all $u \in U$, the point (φ_u, θ_u) belongs to the kilter diagram $\mathscr{C}_u^{[p]}$. For such a point, the maximum flow change necessary to reach the kilter diagram of problem (PA[$p-1$]) is 2^{p-1}. For problem (PA[$p-1$]) the solution (φ, θ) has thus a kilter number at most $M2^{p-1}$ where M is the number of arcs.

(iii) Properties (i) and (ii) above show that, by dividing all the flows by 2^{p-1}, we find an integer-valued flow problem for which the starting solution has kilter number at most M. From Property 1 of Section 3, we thus obtain a complexity bound $O(MN^2)$.

(iv) For $p = \bar{p} = \lceil \log_2 c_{\max} \rceil$ all the allowed intervals for the flows take one of the three forms $[0, 2^{\bar{p}}]$, $[-2^{\bar{p}}, 0]$ or $[-2^{\bar{p}}, 2^{\bar{p}}]$. Hence the zero flow is always feasible.

Moreover, by dividing all the flows and capacity bounds by $2^{\bar{p}}$, we find a minimum cost flow problem with integer $(0, +1, -1)$ components on each arc. For the starting solution $\varphi = 0$ the initial kilter number is $K \leqslant M$, from which it is deduced, as in (iii) above, that the total number of operations is at most $O(MN^2)$. □

5. A polynomial algorithm for the minimum separable quadratic cost flow problem

The suggested algorithm consists in solving problems (PA[p]) for $p = \bar{p}, \bar{p}-1, \ldots, q$ successively. At each step p, the optimum flow φ^{p*} obtained is used as a starting solution for solving (PA[$p-1$]). The procedure is as follows:

Algorithm 1.

(a) Let $\bar{p} = \lceil \log_2 c_{\max} \rceil$. Set $p = \bar{p}$, $\varphi = 0$, $\theta = 0$.

(b) Solve problem (PA[p]) by taking as initial solution the flow φ and the tension θ. Let $\bar{\varphi}$ the optimal flow and $\bar{\theta}$ the optimal tension obtained.

(c) If $p = q$: END. Otherwise set $\varphi \leftarrow \bar{\varphi}$, $\theta \leftarrow \bar{\theta}$, $p \leftarrow p - 1$ and return to (b).

The following property is easily deduced from Property 3.

Property 4. *For all* $q < \bar{p}$ *the maximum number of elementary operations required to determine an optimal solution of* (PA[q]) *by Algorithm* 1 *is* $O((\bar{p} - q)MN^2)$.

We now show how from Properties 2 and 3 of Section 4 an upper bound can be deduced on the number of operations necessary to obtain a solution within a distance $\varepsilon > 0$ of the optimal solution φ^* of (P). (ε being any given tolerance, as small as desired). We shall use the following preliminary results:

Lemma 1. *For any* $p \leqslant \bar{p}$ (*possibly* $p < 0$) *denote by* φ^{p*} *an optimal solution of* (PA[p]). *Then*

$$|\gamma(\varphi^{p*}) - \gamma(\varphi^*)| \leqslant \delta = MW2^{p-3}.$$

Proof. We have

$$\gamma_u(\varphi_u) + w_u 2^{p-3} \geqslant \gamma_u^{(p)}(\varphi_u) \geqslant \gamma_u(\varphi_u) \geqslant \gamma^{(p)_u}(\varphi_u) - w_u 2^{p-3}.$$

A fortiori, by replacing w_u by W

$$\gamma_u(\varphi_u) + W 2^{p-3} \geqslant \gamma_u^{(p)}(\varphi_u) \geqslant \gamma_u(\varphi_u) \geqslant \gamma_u^{(p)}(\varphi_u) - W 2^{p-3}. \tag{6}$$

By setting

$$\gamma(\varphi) = \sum_{u \in U} \gamma_u(\varphi_u)$$

and

$$\gamma^{(p)}(\varphi) = \sum_{u \in U} \gamma_u^{(p)}(\varphi_u)$$

by summation of inequalities (6) above, we get for all φ,

$$\gamma(\varphi) + \delta \geqslant \gamma^{(p)}(\varphi) \geqslant \gamma(\varphi) \geqslant \gamma^{(p)}(\varphi) - \delta \tag{7}$$

where $\delta = MW2^{p-3}$. For $p \leqslant 0$, φ^{p*} is a feasible solution of (P) (cf. Property 2 (iii)). Then, from (7), it follows

$$\gamma^{(p)}(\varphi^{p*}) + \delta \geqslant \gamma(\varphi^{p*}) + \delta \geqslant \gamma^{(p)}(\varphi^{p*}). \tag{8}$$

On the other hand, we have

$$\gamma^{(p)}(\varphi^{p*}) \geqslant \gamma(\varphi^{p*}) \geqslant \gamma(\varphi^*)$$

(since φ^* is the optimum of (P)) and also

$$\gamma(\varphi^*) \geqslant \gamma^{(p)}(\varphi^*) - \delta \geqslant \gamma^{(p)}(\varphi^{p*}) - \delta$$

(since φ^{p*} is an optimum of (PA[p])). Hence we deduce

$$\gamma^{(p)}(\varphi^{p*}) + \delta \geqslant \gamma(\varphi^*) + \delta \geqslant \gamma^{(p)}(\varphi^{p*}). \tag{9}$$

Now, (8) and (9) imply

$$|\gamma(\varphi^{p*}) - \gamma(\varphi^*)| \leqslant \delta. \quad \square$$

Lemma 2. *Let φ^* be the optimal solution of* (P). *Then, for any feasible solution φ of* (P),

$$|\gamma(\varphi) - \gamma(\varphi^*)| \leqslant \delta \;\Rightarrow\; \|\varphi - \varphi^*\|^2 \leqslant 2\delta.$$

Proof. We have

$$\gamma(\varphi) = \tfrac{1}{2} \sum_{u \in U} w_u(\varphi_u - \bar{\varphi}_u)^2$$

and hence

$$\nabla\gamma(\varphi^*) = \left[w_u(\varphi_u^* - \bar{\varphi}_u)\right]_{u \in U}.$$

Since φ^* is the optimal solution of (P), for any feasible solution φ of (P),

$$(\varphi - \varphi^*)\,\nabla\gamma(\varphi^*) \geqslant 0$$

holds, or equivalently

$$\sum_{u \in U} w_u(\varphi_u - \varphi_u^*)(\varphi_u^* - \bar{\varphi}_u) \geqslant 0. \tag{10}$$

Now the condition $\gamma(\varphi) - \gamma(\varphi^*) \leqslant \delta$ can be written as

$$\sum_{u \in U} w_u\left[(\varphi_u - \bar{\varphi}_u)^2 - (\varphi_u^* - \bar{\varphi}_u)^2\right] \leqslant 2\delta$$

or equivalently

$$\sum_{u \in U} \left[w_u(\varphi_u - \varphi_u^*)^2 + 2w_u(\varphi_u - \varphi_u^*)(\varphi_u^* - \bar{\varphi}_u)\right] \leqslant 2\delta.$$

In view of relation (10) this implies

$$\sum_{u \in U} w_u(\varphi_u - \varphi_u^*)^2 \leqslant 2\delta.$$

Since all w_u are integers and $w_u > 0$ we deduce

$$|\gamma(\varphi) - \gamma(\varphi^*)| \leqslant \delta \;\Rightarrow\; \sum_{u \in U} (\varphi_u - \varphi_u^*)^2 \leqslant 2\delta,$$

and Lemma 2 follows. $\square$

Suppose now that we want to determine with Algorithm 1 a solution within a distance $\varepsilon > 0$ of φ^*. The following result specifies the maximum number of operations required.

Theorem 1. *To obtain a feasible solution φ such that $\|\varphi - \varphi^*\| < \varepsilon$ it is sufficient to choose*

$$q = 2 - \lceil \log_2 MW \rceil - \lceil \log_2(1/\varepsilon^2) \rceil.$$

Then the overall complexity of the computation is $O((\bar{p} - q)MN^2)$ *or equivalently*:

$$O(\lceil \log_2 c_{\max} \rceil + \lceil \log_2 MW \rceil + \log_2 \lceil 1/\varepsilon^2 \rceil) \cdot O(MN^2).$$

Proof. In order to get $\|\varphi - \varphi^*\| \leqslant \varepsilon$ it is sufficient to choose $\delta \leqslant \frac{1}{2}\varepsilon^2$ in Lemma 2. From Lemma 1, this will be obtained as soon as $MW2^{q-3} \leqslant \frac{1}{2}\varepsilon^2$, i.e.

$$2^{q-2} \leqslant \varepsilon^2/MW.$$

It is thus sufficient to choose

$$2-q=\lceil \log_2 MW \rceil + \lceil \log_2(1/\varepsilon^2) \rceil. \qquad \square$$

We now turn to state a number of results aimed at specifying the choice of ε in Theorem 1 in order to get the exact optimal solution of (P).

Lemma 3. *The optimal solution φ^* of* (P) *necessarily has all rational components of the form a/b where a and b are integers and*

$$b \leqslant \Delta = W^M(3M+N-1)^{3M+N-1}.$$

Proof. From the results of Section 2, φ is an optimal solution of (P) if and only if the Kuhn–Tucker conditions hold at φ, which in the quadratic case where

$$\gamma(\varphi)=\sum_{u \in U} \tfrac{1}{2} w_u (\varphi_u - \bar{\varphi}_u)^2,$$

read:

$$\begin{aligned}
(\mathrm{KT})' \quad & D(\varphi-\bar{\varphi}) - \pi A + \lambda I - \mu I = 0,\\
& A\varphi = 0,\\
& I\varphi \leqslant c,\\
& -I\varphi \leqslant -b,\\
& \lambda_u(c_u - \varphi_u) = 0 \quad (\forall u \in U),\\
& \mu_u(\varphi_u - b_u) = 0 \quad (\forall u \in U),\\
& \lambda \geqslant 0,\ \mu \geqslant 0,\\
& \pi \text{ unconstrained in sign}
\end{aligned}$$

(D is the diagonal matrix with diagonal coefficients w_u). From a classical result in quadratic programming (cf. Wolfe, 1959), the optimal solution of (P) must correspond to an extreme point of the polytope defined by the linear system (KT)′ (in the space of variables (φ, π, λ, μ)). Lemma 3 is then obtained by taking for Δ an upper bound of the determinant of the bases of system (KT)′. Such an upper bound may be derived by assuming that each of the $(3M+N-1)!$ permutations is involved with the same sign in the determinant, and by bounding the weight of each permutation by W^M. Thus we can choose

$$\begin{aligned}
\Delta &= W^M(3M+N-1)^{3M-N-1}\\
&\geqslant W^M(3M+N-1)!. \qquad \square
\end{aligned}$$

Lemma 4. *Let φ a solution of* (P) *such that $\|\varphi-\varphi^*\| \leqslant 1/2\Delta^2$ where Δ is given by Lemma 3. Then in the ball of radius $1/2\Delta^2$ and centered at φ, φ is the only vector with all rational components of the form a/b, where a and b are integers and $b \leqslant \Delta$. φ^* can then be computed (in polynomial time) from φ by replacing each component of φ by the closest rational number with denominator $\leqslant \Delta$.*

Proof. If y is a vector with rational components a/b where $b \leqslant \Delta$ and if $y \neq \varphi^*$, then $\|y-\varphi^*\| \geqslant 1/\Delta^2$ must hold, which shows that y cannot belong to the ball centered at y and with radius $1/2\Delta^2$. Since φ^* is the only vector with rational components a/b ($b \leqslant \Delta$) in this ball, it can be obtained by rounding each component in turn to the closest rational number with denominator $\leqslant \Delta$. To that aim, the technique of continued fractions can be used (cf. Bland, Goldfarb and Todd, 1981; Niven and Zuckerman, 1966). $\square$

Theorem 2. *Algorithm 1 solves the minimum quadratic (separable) cost flow problem in polynomial time*

$$\begin{aligned}
O(MN^2)\cdot O\big[&\lceil \log_2 c_{\max} \rceil + \lceil \log_2 MW \rceil\\
&+4M\lceil \log_2 N \rceil\\
&+4(3M+N-1)\lceil \log_2(3M+N-1) \rceil\big],
\end{aligned}$$

to which should be added the final polynomial rounding procedure.

Proof. Just apply Theorem 1 with $\varepsilon = 1/2\Delta^2$ and take $\Delta = W^M(3M+N-1)3M+N-1$. Then we have

$$\begin{aligned}
\log_2(1/\varepsilon^2) = \log_2 4\Delta^4 &= 2 + 4\log_2\Delta\\
&= 2 + 4M\log_2 W\\
&\quad + 4(3M+N-1)\log_2(3M+N-1).
\end{aligned}$$

$\square$

We end up by observing that, in practice, the number of iterations of Algorithm 1 will be generally limited by the maximum accuracy allowed by the computer. For instance, if we use a computer with a 36 binary digits arithmetic, the maximum accuracy will be reached in about 30 iterations of the algorithm. Thus the computations will have to be stopped, in practice, well before the theoretical number of iterations specified by Theorem 2. The use of multiple word length arithmetic may turn out to be necessary in some situations.

6. Preliminary computational results and conclusions

Though polynomiality of an algorithm may usually be considered as a criterion of efficiency, some counter examples are known (such as Khachian's algorithm) where polynomial algorithms appear to be rather unefficient in practice. In order to investigate the practical usefulness of the approach describe here, Algorithm 1 has been implemented and tested on a number of small-to-medium sized problems. Computational experiments have been carried out in order to check whether the scaling approach could actually be regarded as intrinsically superior to the naive approach, namely applying the out-of-kilter method directly to the piecewise linear cost flow problem obtained by discretizing the cost functions according to some final desired accuracy. The comparative results obtained for several values of the final accuracy are displayed in Table 1 where the computing times have been indicated in terms of the total number of labellings required, in each case, by the out-of-kilter algorithm. From this it will be observed that:

- in a huge majority of cases, the scaling approach requires fewer labellings than the naive approach,
- the average rate of improvement (indicated in the last line of Table 1) seems to grow rapidly with the final required accuracy.

These preliminary results clearly confirm that, beyond its theoretical interest for constructively proving the polynomial solvability of quadratic separable cost flow problems, the scaling approach should be considered as a powerful tool for actually solving practical instances of these problems.

As a conclusion, we recall that quadratic separable cost flow problems arise in various important contexts of applications such as:

- best least squares approximation of a matrix with prescribed row and column sums with applications to approximating input–output matrices in economy, or forecasting traffic matrices in telecommunications networks (see Allen and Gossling, 1975; Minoux, 1978; Debiesse and Matignon, 1981);
- solving nondifferentiable cost flow problems via subgradient algorithms, where at each step a projection onto the polyhedron of feasible flows has to be performed (for more details see Minoux, 1982);
- the problem of minimizing a quadratic separable cost function under a single linear constraint (a generalization of the well-known 'continuous' knapsack problem) may be reformulated, as suggested in Minoux (1982), as a quadratic separable cost flow problem. The approach described here thus provides an alternative to the polynomial algorithm described by Helgason et al. (1980).

References

Allen, R.I.G. and Gossling, W.F. (1975), "*Estimating and Projecting Input–output Coefficients*", Input–output Publishing Company, London.

Table 1
Comparison of computing times obtained on a number of test problems with a standard out-of-kilter algorithm (no scaling) and with the polynomial algorithm (with scaling). Computing times are expressed in terms of the number of labellings required by the out-of-kilter method.

	Accuracy 2^{-2}		Accuracy 2^{-4}		Accuracy 2^{-6}	
	no scaling	scaling	no scaling	scaling	no scaling	scaling
Problem 1 (6 nodes 11 arcs)	18	19	59	29	221	40
Problem 2 (15 nodes 55 arcs)	158	67	483	100	1888	125
Problem 3 (20 nodes 30 arcs)	70	63	252	81	979	106
Problem 4 (20 nodes 40 arcs)	79	57	281	69	1093	81
Average rate of improvement	1.5		3.5		11	

Bacharach, M. (1965), "Estimating nonnegative matrices from marginal data", *International Economic Review* 6, 294–310.

Bachem, A. and Korte, B. (1980), "Minimum norm problems over transportation polytopes", *Linear Algebra and its Application* 31, 103–118.

Bland, R.G., Goldfarb, D. and Todd, M.J. (1981), "The ellipsoid method: A survey", *Operations Research* 29 (6) 1039–1091.

Bihain, A., Strodiot, J.J. and Nguyen, V.H. (1980), "A reduced subgradient method", Workshop on "Linear and nonlinear Programming", E.I.A.S.M., Bruxelles, 13–14 November 1980, to appear.

Cantor, D.G. and Gerla, M. (1974), "Optimal routing in a packet-switched computer network", *IEEE Transactions on Computers*, C-23 (10), 1062–1069.

Cottle, R.W. and Pang, J.S. (1980), "On the convergence of a block successive overrelaxation method for a class of linear complementarity problems", Technical Report SOL 80-17, Dept. Operations Research, Stanford University, CA.

Debiesse, J.L. and Matignon, G. (1980), "Comparison of different methods for the calculation of traffic matrices", *Annales des Télécommunications* 35, 91–102.

Dembo, R.S. and Klincewicz (1978), "A second order algorithm for network flow problems with convex separable costs", Technical Report 21, School of Business, Yale University, New Haven, CT.

Edmonds, J. and Karp, R.M. (1972) "Theoretical improvements in algorithmic efficiency for network flow problems", *Journal of the ACM* 19 (2) 248–264.

Ferland, J.A. (1974), "Minimum cost multicommodity circulation problems with convex arc costs", *Transportation Science* 8 (4) 335–360.

Ford, L.R. and Fulkerson, D.R. (1962), *Flows in Networks*, Princeton University Press, NY.

Gondran, M. and Minoux, M. (1979), *Graphes et Algorithmes*, Eyrolles, France; English translation, Wiley, 1984.

Grötschel, M., Lovàsz, L. and Schrijver, A. (1981), "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica* 1 (2) 169–197.

Helgason, R., Kennington, J. and Lall, H. (1980), "A polynomially bounded algorithm for a singly constrained quadratic program", *Mathematical Programming* 18, 338–343.

Hu, T.C. (1966), "Minimum cost flows in convex cost networks", *Naval Research Logistics Quarterly* 13 (1) 1–9.

Jaumard, B. and Minoux, M. (1982), "Un algorithme de type pivots complémentaires pour les problèmes de flots à coûts quadratiques séparables", internal note of CNET, Issy, France.

Kennington, J.L. and Helgason, R.V. (1980), *Algorithms for Network Programming*, Wiley, New York.

Kennington, J. and Shalaby, M. (1977), "An effective subgradient procedure for minimal cost multicommodity flow problems", *Management Science* 23, 994–1004.

Khachian, L.G. (1979), "A polynomial algorithm in linear programming", *Soviet Mathematics. Doklady* 20, 191–194.

Kozlov, M.K., Tarasov, S.P. and Khachian, L.G. (1979), "Polynomial solvability of convex quadratic programming", *Soviet Mathematics. Doklady 20, 1108–1111.*

Lawler, E. (1976), Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York.

Leblanc, L.J., Morluk, E. and Pierskalla, W. (1975), "An efficient approach to solving the road network equilibrium traffic assignment problem", *Transportation Research* 9, 309–318.

Lemke, C.E. (1962), "A method of solution for quadratic programs", *Management Science* 8, 442–455.

Minoux, M. (1978), "Un algorithme de sous-gradient pour l'extrapolation des matrices de trafic pour une norme quadratique" (internal note of CNET, cited in Debiesse Matignon, 1980).

Minoux, M. (1982), "Un algorithme polynomial pour les problèmes de flots à coûts quadratiques et applications", internal note of CNET, Issy, France.

Minoux, M. (1983), *Programmation Mathématique: Théorie et Algorithmes, Vols 1 and 2* Collection Technique et Scientifique des Télécommunications, Dunod, Paris.

Minoux, M. (1983), "Solution explicite d'un problème de transport à coûts quadratiques et applications", Preprint #484, University of Montréal, Departement of Informatics and Operations Research, July 1983.

Nguyen, S. (1974), "An algorithm for the traffic assignment problem", *Transportation Science* 8, 203–216.

Niven, I. and Zuckerman, H.S. (1966), *An Introduction to the Theory of Numbers*, Wiley, New York.

Rockafellar, R.T. (1970), *Convex Analysis*, Princeton University Press, NY.

Srinivasan, V. and Thompson, G.L. (1972), "An operator theory of parametric programming for the transportation problem I and II", *Naval Research Logistics Quarterly* 19, 205–252.

Srinivasan, V. and Thompson, G.L. (1977), "Cost operator algorithms for the transportation problem", *Mathematical Programming* 12, 372–391.

Wolfe, P. (1959), "The Simplex method for quadratic programming", *Econometrica*, 27 (3) 382–398.