

# CONTINUOUS-TIME QUADRATIC COST FLOW PROBLEMS WITH APPLICATIONS TO WATER DISTRIBUTION NETWORKS

A. T. ERNST<sup>1</sup>

(Received 18 April 1994; revised 19 August 1994)

## Abstract

This paper deals with a class of network optimization problems in which the flow is a function of time rather than static as in the classical network flow problem, and storage is permitted at the nodes. A solution method involving discretization will be presented as an application of the ASG algorithm. We furnish a proof that the discretized solution converges to the exact continuous solution. We also apply the method to a water distribution network where we minimize the cost of pumping water to meet supply and demand, subject to both linear and nonlinear constraints.

## 1. Introduction

The basic continuous-time network flow problem can be described as follows. Consider a network in which some of the nodes produce or consume goods at a given rate (which may change over time). The goods can either be stored at the nodes or they can be passed on to adjacent nodes *via* a given set of arcs. The storage capacity of the nodes and the rate at which the commodity can be transported across the arcs are both limited. The aim is to find the cheapest way to satisfy the demands given some cost for transporting the commodities across arcs and possibly also for storing them at the nodes. The optimization is done over a fixed interval of time in which all parameters (that is, the costs, capacities and demands) may vary.

Network optimization problems involving continuous-time flow arise naturally in many situations, for example computer networks, transportation networks, flow of water or gas through a system of pipes and reservoirs or financial transactions. The case where the cost to be minimized is linear in the flows has been studied by Anderson and Philpott [1]. This can be considered as a special case of more general continuous linear programming problems, see for example Buie and Abraham [6] or more recently

---

<sup>1</sup>Department of Mathematics, The University of Western Australia, Nedlands WA 6009.

© Australian Mathematical Society, 1996, Serial-fee code 0334-2700/96

Pullan [10]. Hence in this paper we will concentrate on the quadratic cost case, which to the best of our knowledge has not been treated before. Another way to look at this problem is as a special case of the more general nonlinear optimal control problem. The method of discretization presented in this paper can be viewed as a specialisation of the control parameter approach taken by Teo *et al.* [11].

Network problems with quadratic costs may arise in a variety of ways. In financial investments, a quadratic cost function may be used to measure variance. This was originally proposed by Markowitz [9]. In traffic networks a quadratic cost term could be used to penalize congestion. Another application, which will be addressed in this paper, is in certain types of pipe networks. Here the quadratic term in the cost arises from the hydraulic equation, relating pumping effort to the rate of flow in the pipes.

In the remainder of this section, the basic definitions and notations are introduced as well as the general form of the problem. We then proceed to show how the problem can be discretized, and that the solutions obtained using this discretization scheme converge to an optimal solution. In the last section we show how a water distribution problem can be brought into the form discussed in this paper. This is an application relevant to the administration of the water supply in a city or other area connected by a common water network. The objective is to determine how to meet the supply and demand over a period of time, such that the pumping costs are minimized and none of the physical constraints imposed on the system are violated. Some numerical results are also presented in this last section, showing that it is computationally feasible to find good approximate solutions to problems of this type.

**1.1. Definitions and Notations** Some of the definitions and notations used in this paper are:

DEFINITION 1.

1. A *directed graph* or *network* is an ordered pair  $(V, A)$  where  $V$  is the set of *nodes* or *vertices* and  $A$  is the set of directed arcs joining them. Each *directed arc*,  $a$  is defined by a source and destination node, denoted by  $\alpha(a)$  and  $\omega(a)$  respectively. In this paper the number of nodes will be denoted by  $n = |V|$  and the number of arcs by  $m = |A|$ .
2. The network structure may be represented by an *arc-node incidence matrix*  $E$ , with entries of the form

$$E_{i,k} \hat{=} \begin{cases} +1 & \text{if } i = \alpha(k), \\ -1 & \text{if } i = \omega(k), \\ 0 & \text{otherwise.} \end{cases}$$

3. The net flow out of the nodes is called the *divergence* of a the flow. If  $f \in \mathbb{R}^{|A|}$  is the flow in the arcs, then the divergence is simply  $Ef$ .

DEFINITION 2. The following matrix and vector notation will be used.

1.  $\text{diag}(d_1, d_2, \dots, d_k)$  denotes a diagonal matrix with entries  $d_1, d_2, \dots, d_k$  on the main diagonal.
2.  $\mathbf{1}$  is a vector of all ones, with the dimension given by the context.

1.2. **The continuous-time problem** The continuous-time quadratic network problem to be considered in this paper can be stated as follows.

PROBLEM CQNP.

$$\text{Min } f(\mathbf{x}) = \int_0^T \frac{1}{2} \mathbf{x}^T(t) Q(t) \mathbf{x}(t) + \mathbf{c}^T(t) \mathbf{x}(t) dt \tag{1}$$

$$\text{subject to } \dot{\mathbf{s}}(t) = -E\mathbf{x}(t) - \mathbf{r}(t) \quad \forall t \in [0, T] \text{ and } \mathbf{s}(0) = \mathbf{0}, \tag{2}$$

$$\mathbf{0} \leq \mathbf{s}(t) \leq \mathbf{a}(t) \quad \forall t \in [0, T], \tag{3}$$

$$\mathbf{0} \leq \mathbf{x}(t) \leq \mathbf{b}(t) \quad \text{for almost all } t \in [0, T], \tag{4}$$

where

$[0, T]$  is a given interval of time.

$\mathbf{x} : [0, T] \rightarrow \mathbb{R}^m$  is the control vector denoting the rate of flow in each arc.

$\mathbf{s} : [0, T] \rightarrow \mathbb{R}^n$  is the state vector denoting the storage at the nodes. The notation  $s(t|\mathbf{x})$  will be used to denote the storage at time  $t$  given by  $\mathbf{x}$  and the state equation (2).

$\mathbf{a} : [0, T] \rightarrow \mathbb{R}^n$  denotes the storage capacity at each node at any time in  $[0, T]$ . Each component of  $\mathbf{a}$  must be continuous and differentiable almost everywhere with piecewise continuous derivatives. (Note that it would make no sense to allow  $\mathbf{a}$  to be discontinuous since the storage  $\mathbf{s}$  is continuous everywhere.)

$\mathbf{b} : [0, T] \rightarrow \mathbb{R}^m$  denotes the maximum rate of flow for each arc at any time in  $[0, T]$ .

$\mathbf{c} : [0, T] \rightarrow \mathbb{R}^m$  is the linear cost coefficient.

$Q = \text{diag}(q_1, q_2, \dots, q_m)$  is the quadratic cost. Each function  $q_i : [0, T] \rightarrow \mathbb{R}$  must satisfy  $q_i(t) \geq 0, \forall t \in [0, T]$ .

$\mathbf{r} : [0, T] \rightarrow \mathbb{R}^n$  is the demand of each node (or supply when  $r_i(t) < 0$ ). For simplicity it will also be assumed that supply equals demand, that is,

$$\int_0^T \mathbf{1}^T \mathbf{r}(t) dt = 0, \tag{5}$$

so that  $\mathbf{s}(T) = \mathbf{0}$  (since  $\mathbf{s}(T) \geq \mathbf{0}$  and from  $\mathbf{s}(0) = \mathbf{0}$  we get  $\mathbf{1}^T \mathbf{s}(T) = 0$ ).

In addition, we require that each of the above functions be bounded and Lebesgue measurable functions of time. In particular, the minimization in problem CQNP is only done over the set of measurable flows.

REMARK 1. A storage cost of the form  $\mathbf{d}^\top(t)s(t)$  could be added to the integrand of the objective function. However this does not really make the problem any more general since we can write

$$\begin{aligned} \int_0^T \mathbf{d}^\top(t)s(t) dt &= \int_0^T \mathbf{d}^\top(t) dt s(T) - \int_0^T \left[ \int_0^t d(\tau) d\tau \right]^\top \dot{s}(t) dt \\ &= 0 - \int_0^T \hat{\mathbf{d}}^\top(t) (E\mathbf{x}(t) - \mathbf{r}(t)) dt, \quad \text{where } \hat{\mathbf{d}}(t) = \int_0^t d(\tau) d\tau. \end{aligned}$$

Hence the cost of storage can be included by modifying the linear cost  $c$  appropriately.

REMARK 2. CQNP is a special type of optimal control problem which in principle can be solved by standard optimal control software. However a typical network will have a large number of nodes and arcs, making the problem too large to be handled by general purpose software. Furthermore, state constraints of type (3) are well-known to be difficult to handle computationally, so special methods are warranted to exploit the network structure.

REMARK 3. The method for solving CQNP can be extended to solve problems with general convex cost functionals by a sequence of CQNPs in a similar fashion to the sequential quadratic programming technique.

## 2. Solution by discretization

**2.1. Discretizing the time variable** To approximate the continuous time problem by a more tractable static optimization problem, the interval  $[0, T]$  is partitioned into subintervals. Note that even in the linear case, all methods for solving minimum cost network flow problems resort to some form of discretization of the time variable. The only exception to this is the attempt by Anderson and Philpott [1] to produce a simplex-like method. However, as Pullan [10] has pointed out, this did not yield a practical algorithm.

DEFINITION 3. A *partition* of the interval  $[0, T]$  is a set of intervals  $\mathcal{P} \hat{=} \{I_k \mid k = 1, 2, \dots, K\}$  of the form  $I_k = (t_{k-1}, t_k]$ , where  $0 = t_0 < t_1 < \dots < t_K = T$  (except that  $I_1 = [t_0, t_1]$ ).

DEFINITION 4. For any partition, we define for all  $k = 1, 2, \dots, K$

$$a_k \hat{=} a(t_k), \tag{6}$$

$$b_k \hat{=} \int_{I_k} b(t) dt, \tag{7}$$

$$c_k \hat{=} \frac{1}{|I_k|} \int_{I_k} c(t) dt, \tag{8}$$

$$r_k \hat{=} \int_{I_k} r(t) dt \tag{9}$$

and

$$Q_k \hat{=} \frac{1}{|I_k|^2} \int_{I_k} Q(t) dt. \tag{10}$$

The discrete problem used to approximate CQNP can now be written as follows

PROBLEM DQNP.

$$\text{Min } F(z, y) = \sum_{k=1}^K z_k^T Q_k z_k + c_k^T z_k \tag{11}$$

$$\text{subject to } y_k - y_{k-1} = -Ez_k - r_k \quad \forall k \text{ and } y_0 = \mathbf{0}, \tag{12}$$

$$\mathbf{1}^T y_K = 0, \tag{13}$$

$$\mathbf{0} \leq y_k \leq a_k \quad \forall k = 1, 2, \dots, K, \tag{14}$$

$$\mathbf{0} \leq z_k \leq b_k \quad \forall k = 1, 2, \dots, K, \tag{15}$$

where  $F(z, y)$  is a discrete approximation to  $f(x(\cdot))$  (in the discrete context  $y$  may be thought of as an independent variable although it is really dependent on  $z$  by the constraint (12))

$y = [y_1^T y_2^T \dots y_K^T] \in \mathbb{R}^{Kn}, y_k \in \mathbb{R}^n$  is the storage at the end of each interval  $I_k$ ,

$z = [z_1^T z_2^T \dots z_K^T] \in \mathbb{R}^{Km}, z_k \in \mathbb{R}^m$  is the accumulated flow in interval  $I_k$ .

REMARK 4. Equation (13) is dependent on (12) since  $\sum_{k=1}^K \mathbf{1}^T r_k = 0$ , by (5). The reason for including it is to give the problem a proper network structure as shown below.

The problem DQNP is in the form of a static quadratic cost network problem which can be expressed in matrix notation as

$$\text{Min } F(z, y) = \frac{1}{2} z^T Qz + c^T z \tag{16}$$

subject to

$$\left[ \begin{array}{cccc|cccc} I & & & & E & & & \\ -I & I & & 0 & & E & & 0 \\ & & \ddots & \ddots & & & \ddots & \\ & & & & & 0 & & E \\ & 0 & & -I & I & & & \\ 0 & \dots & \dots & 0 & -\mathbf{1}^\top & 0 & \dots & \dots & \dots & E & 0 \end{array} \right] \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} -r \\ 0 \end{bmatrix}, \quad (17)$$

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} y \\ z \end{bmatrix} \leq \begin{bmatrix} a \\ b \end{bmatrix}. \quad (18)$$

The node-arc incidence matrix of this problem defines a network consisting of  $K$  copies of the original network stacked on top of each other with arcs going from each node to the corresponding node below and arcs from the last copy to a dummy node. A similar type of discretization has been suggested by Ford and Fulkerson [8] for maximal dynamic flow problems, and this scheme has since been used by other authors, for example [6, 10].

DEFINITION 5. It is convenient to let  $y(z)$  represent the value of  $y$  given by  $z$ , resulting from (12).

There are a number of algorithms which can be used to solve the discrete quadratic cost problem. The most popular algorithm for this type of problem is the convex network simplex algorithm. However it was shown in [4, 2] that a specialisation of the active set algorithm for network problems performs significantly better than the convex network simplex algorithm (by more than two orders of magnitude). A description of the algorithm and some of the underlying theory is given as an appendix to this paper.

**2.2. Convergence of the discrete approximation** We shall now show that the optimal solutions to a sequence of problems of the form DQNP, with increasing partition size, define a sequence of continuous flows which converge to the optimal solution of CQNP. Our presentation follows closely the exposition given by Teo *et al.* [11] for general optimal control problems. We define a sequence of partitions used to define discrete problems from CQNP as follows.

DEFINITION 6. The notation  $\{I_k^p\}_{k=1}^{K_p}$  for  $p = 1, 2, 3, \dots$  will be used to denote a sequence of partitions of  $[0, T]$ , where  $I_k^p \hat{=} (t_{k-1}^p, t_k^p]$ . The intervals  $I_k^p$  are chosen so that they have the following properties.

- $\forall k \exists j$  such that  $I_k^p \subseteq I_j^{p-1}$ , or equivalently  $\{t_0^{p-1}, \dots, t_{K_{p-1}}^{p-1}\} \subseteq \{t_0^p, \dots, t_{K_p}^p\}$ .

$$2. \lim_{p \rightarrow \infty} \max_{k=1,2,\dots,K_p} |I_k^p| = 0.$$

The notation  $a_k^p$ ,  $b_k^p$ , etc. will be used to denote the discretized variables with respect to the  $p^{th}$  partition.

The correspondence between the discrete solutions and time variable flows is given by the next definition.

**DEFINITION 7.** For any given  $z$ , let the corresponding flow rate for an equivalent continuous problem,  $x(t|z)$  be defined by

$$x(t|z) \hat{=} \sum_{k=1}^K \chi_{I_k}(t) \frac{z_k}{|I_k|}, \tag{19}$$

where

$$\chi_S(t) \hat{=} \begin{cases} 1 & t \in S \\ 0 & \text{otherwise.} \end{cases}$$

Also define  $s(t|z) \hat{=} s(t|x(t|z))$ .

Consider the sequence of functions  $x(t|z^p)$  for  $p = 1, 2, 3 \dots$ , where each  $z^p$  is optimal for the discretized problem corresponding to partition  $p$ . Since each of these functions is measurable and almost everywhere bounded by  $\text{esssup}_{[0,T]} b$ , independent of  $p$ , the sequence converges to a bounded and measurable function  $x^\infty(t)$ . The proof that this function is feasible for CQNP is contained in Lemmas 1 and 3. The optimality of  $x^\infty(t)$  is proven in Theorem 1.

**LEMMA 1.** *Let  $\{z^p\}_{p=1}^\infty$  be a sequence of feasible solutions to the discretized network problem DQNP. Then the arc capacity violation of the corresponding continuous solution approaches zero as  $p$  goes to infinity. More precisely,*

$$\lim_{p \rightarrow \infty} \int_0^T \max \{x(t|z^p) - b(t), \mathbf{0}\} dt = \mathbf{0},$$

where the maximum is taken component-wise.

**REMARK 5.** The advantage of dealing with the integral of the arc capacity violation, is that no special allowance has to be made for violations on sets of measure zero.

PROOF OF LEMMA 1.

$$\int_0^T \max(\mathbf{x}(t|z^p) - \mathbf{b}(t), 0) dt = \sum_{k=1}^{K_p} \int_{I_k^p} \max\left(\frac{z_k^p}{|I_k^p|} - \mathbf{b}(t), 0\right) dt$$

$$\leq \sum_{k=1}^{K_p} \int_{I_k^p} \max\left(\frac{\mathbf{b}_k^p}{|I_k^p|} - \mathbf{b}(t), 0\right) dt.$$

For any Lebesgue point  $\tau$  of  $\mathbf{b}$ ,

$$\mathbf{b}(\tau) = \lim_{p \rightarrow \infty} \frac{1}{|I_{k(p)}^p|} \int_{I_{k(p)}^p} \mathbf{b}(t) dt = \lim_{p \rightarrow \infty} \frac{\mathbf{b}_{k(p)}^p}{|I_{k(p)}^p|},$$

where for any  $p$ ,  $k(p)$  is chosen so that  $\tau \in I_{k(p+1)}^{p+1} \subset I_{k(p)}^p$ .  
 Almost all points of  $[0, T]$  are Lebesgue points of  $\mathbf{b}$ . Hence

$$\sum_{k=1}^{K_p} \int_{I_k^p} \max\left(\frac{\mathbf{b}_k^p}{|I_k^p|} - \mathbf{b}(t), 0\right) dt \rightarrow \mathbf{0} \text{ as } p \rightarrow \infty.$$

LEMMA 2. For any partition of  $[0, T]$  with end-points  $t_0, t_1, \dots, t_K$

$$\mathbf{y}_k(\mathbf{z}) = \mathbf{s}(t_k|\mathbf{z}) \quad \forall k = 0, 1, \dots, K.$$

PROOF. This just requires some straightforward manipulation of the definition of  $\mathbf{s}(t_k|\mathbf{z})$  and  $\mathbf{y}_k$ .

LEMMA 3. Let  $\{z^p\}_{p=1}^\infty$  be a sequence of feasible solutions to the discretized network problem DQNP. Then

$$\lim_{p \rightarrow \infty} \sup_{t \in [0, T]} (\mathbf{s}(t|z^p) - \mathbf{a}(t)) \leq \mathbf{0},$$

where the supremum is taken component-wise.

PROOF. For all times  $t$  such that  $t = t_k^p$  for some  $p, k$ ,  $\mathbf{s}(t|z^p) - \mathbf{a}(t) \leq \mathbf{0}$  by the previous lemma. For the points in between, the result follows from the continuity of  $\mathbf{s}$  and  $\mathbf{a}$  and from the observation that both  $\dot{\mathbf{s}}$  and  $\dot{\mathbf{a}}$  are bounded.

LEMMA 4. Let  $\mathbf{z}$  be a feasible solution for DQNP with partition  $\{I_k\}_{k=1}^K$ . Then

$$f(\mathbf{x}(\cdot|\mathbf{z})) = F(\mathbf{z}, \mathbf{y}(\mathbf{z})).$$



PROOF.

$$\begin{aligned}
 f(\mathbf{x}(\cdot|\mathbf{z})) &= \sum_{k=1}^K \int_{I_k} \mathbf{x}^\top(t|\mathbf{z}) Q(t)\mathbf{x}(t|\mathbf{z}) + \mathbf{c}^\top(t) \mathbf{x}(t|\mathbf{z}) dt \\
 &= \sum_{k=1}^K \frac{\mathbf{z}_k^\top \int_{I_k} Q(t)dt \mathbf{z}_k}{|I_k|^2} + \int_{I_k} \mathbf{c}^\top(t) dt \frac{\mathbf{z}_k}{|I_k|} \\
 &= \sum_{k=1}^K \mathbf{z}_k^\top Q_k \mathbf{z}_k + \mathbf{c}_k^\top \mathbf{z}_k = F(\mathbf{z}, \mathbf{y}(\mathbf{z})).
 \end{aligned}$$

LEMMA 5. Let  $\hat{\mathbf{x}}$  be an optimal solution of CQNP. Let  $\bar{\mathbf{z}}_k^p \hat{=} \int_{I_k^p} \hat{\mathbf{x}}(t) dt$ . Then

$$\lim_{p \rightarrow \infty} f(\mathbf{x}(\cdot|\bar{\mathbf{z}}^p)) = f(\hat{\mathbf{x}}(\cdot)).$$

PROOF. For any Lebesgue point  $\tau$  of  $\hat{\mathbf{x}}$  such that  $\tau \in I_{k(p)}^p \forall p$ ,

$$\hat{\mathbf{x}}(\tau) = \lim_{p \rightarrow \infty} \frac{1}{|I_{k(p)}^p|} \int_{I_{k(p)}^p} \hat{\mathbf{x}}(t) dt = \lim_{p \rightarrow \infty} \frac{\bar{\mathbf{z}}_{k(p)}^p}{|I_{k(p)}^p|}.$$

Almost all points in  $[0, T]$  are Lebesgue points of  $\hat{\mathbf{x}}$  since  $\hat{\mathbf{x}}$  is a Lebesgue measurable function. Hence  $\mathbf{x}(\cdot|\bar{\mathbf{z}}^p) \rightarrow \hat{\mathbf{x}}$  pointwise almost everywhere as  $p \rightarrow \infty$ . Therefore

$$\begin{aligned}
 &\lim_{p \rightarrow \infty} \int_0^T \mathbf{x}^\top(t|\bar{\mathbf{z}}^p) Q(t)\mathbf{x}(t|\bar{\mathbf{z}}^p) + \mathbf{c}^\top(t) \mathbf{x}(t|\bar{\mathbf{z}}^p) dt \\
 &= \int_0^T \hat{\mathbf{x}}^\top(t) Q(t)\hat{\mathbf{x}}(t) + \mathbf{c}^\top(t) \hat{\mathbf{x}}(t) dt.
 \end{aligned}$$

THEOREM 1. For any partition  $\{I_k^p\}_{k=1}^{K_p}$  let  $\hat{\mathbf{z}}^p$  be an optimal solution to the discrete quadratic cost network problem. Let  $\hat{\mathbf{x}}$  denote the optimal solution of the continuous problem. Then

$$\lim_{p \rightarrow \infty} F(\hat{\mathbf{z}}^p, \mathbf{y}(\hat{\mathbf{z}}^p)) = f(\hat{\mathbf{x}}(\cdot)).$$

PROOF. This follows in a straightforward manner from the previous lemmas and the optimality of  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{z}}^p$ :

$$f(\hat{\mathbf{x}}(\cdot)) \leq f(\hat{\mathbf{x}}(\cdot|\hat{\mathbf{z}}^p)) = F(\hat{\mathbf{z}}^p, \mathbf{y}(\hat{\mathbf{z}}^p)) \leq F(\bar{\mathbf{z}}^p, \mathbf{y}(\bar{\mathbf{z}}^p)) \rightarrow f(\hat{\mathbf{x}}(\cdot)).$$

Therefore  $F(\hat{\mathbf{z}}^p, \mathbf{y}(\hat{\mathbf{z}}^p)) \rightarrow f(\hat{\mathbf{x}}(\cdot))$  as  $p \rightarrow \infty$ .

### 3. Water distribution networks

**3.1. Problem description** As an application of the theory above, we will consider the following pipe network problem. Let the arcs represent a set of pipes connecting some storage tanks. The flow in each pipe is assisted by a pump, which may be used to control the flow through the pipe at a cost which is proportional to the amount of power used to pump water through the pipe. This cost may well vary with time since the cost of the electricity used by the pumps is usually cheaper during off-peak times for large power users. The rate at which water actually flows through the pipe depends not only on the setting of the pump but also on the pressure difference between the two tanks connected by the pipe. The nonlinear equation relating water flow rate to pump assisted pressure difference is called the hydraulic equation, which is usually determined empirically. Note that depending on such factors as the flow speeds involved and the pipe diameters, some quite different equations may be appropriate for modelling this relationship between pressure difference, pump activity and flow speed; for more details on this subject see for example [7, 5]. The pressure at the ends of the pipe, in turn, depends on the amount of water in the tank, as well as the height at which the tank is located relative to the other nodes in the network. The problem is to minimize the pumping costs while meeting all of the demands. The demand is assumed to be known since accurate forecasts of the daily water demand can generally be made on the basis of data about the past demands. Also the supply in the form of water available in reservoirs or the amount of ground water that can be pumped into the system is sufficiently well-known. There are of course restrictions on the amount of water that can be stored in the tanks as well as on the maximum pressure and possibly on the rate of flow in the pipes. Taking into account all of these constraints, the problem can be written as

$$\begin{aligned}
 & \text{Min} \quad \int_0^T \mathbf{c}^\top(t) \mathbf{p}(t) dt \\
 & \text{subject to} \quad \dot{\mathbf{s}}(t) = -E\mathbf{x} - \mathbf{r}, \\
 & p_a(t) + h_{\omega(a)}(t) - h_{\alpha(a)}(t) = \phi_a(x_a(t)) \quad \forall a \in A, t \in [0, T], \\
 & h_n(t) = \psi_n(s_n(t)) \quad \forall n \in V, \\
 & \mathbf{0} \leq \mathbf{h}(t) \leq \bar{\mathbf{h}}(t), \\
 & \mathbf{0} \leq \mathbf{s}(t) \leq \bar{\mathbf{s}}(t), \\
 & \mathbf{0} \leq \mathbf{x}(t) \leq \bar{\mathbf{x}}(t),
 \end{aligned}$$

where

- $\mathbf{p}$  is the pump assisted pressure vector for each pipe;
- $\mathbf{c}$  is a vector of costs;

- $\mathbf{h}, \bar{\mathbf{h}}$  are the pressure vectors at the tank and the maximum allowable pressure;
- $\mathbf{s}, \bar{\mathbf{s}}$  represent the tanks' storage vector and the tanks' capacity vector respectively;
- $\mathbf{x}, \bar{\mathbf{x}}$  represent the rate of flow vector and the upper bound on this flow rate;
- $\mathbf{r}$  is the demand vector as before;
- $\phi_a$  is a continuous function in its arguments, which models the physical relationship between pump speed, pressure difference and flow rate for each pipe  $a$ ; and
- $\psi_n$  is a continuous function giving the pressure created in tank  $n$  by a given amount of water stored.

According to Collins *et al.* [7], the hydraulic equation can be approximated by a function  $\phi_a$  which may, in the first instance, be assumed to be quadratic in the flow rate. Hence we can write for each pipe  $a$

$$p_a(t) = h_{\alpha(a)}(t) - h_{\omega(a)}(t) + \beta_a x_a^2(t) + \gamma_a x_a(t) + \delta_a,$$

for some empirically determined constants  $\beta_a, \gamma_a, \delta_a$ . If we assume that the storage tanks have vertical walls, then the relationship between storage and pressure is linear, so for each tank  $n$

$$h_n(t) = \eta_n s_n(t) + \zeta_n.$$

By substituting these equations into the problem definition above, we get a new problem involving only  $\mathbf{x}$  and  $\mathbf{s}$ :

$$\begin{aligned} \text{Min} \quad & \int_0^T \mathbf{c}_\epsilon^\top(t) \mathbf{s}(t) + \frac{1}{2} \mathbf{x}^\top(t) \mathbf{Q}(t) \mathbf{x}(t) + \mathbf{c}_\gamma^\top(t) \mathbf{x}(t) dt \\ \text{subject to} \quad & \dot{\mathbf{s}}(t) = -\mathbf{E}\mathbf{x} - \mathbf{r}, \\ & \mathbf{0} \leq \mathbf{x}(t) \leq \bar{\mathbf{x}}(t), \\ & \mathbf{0} \leq \mathbf{s}(t) \leq \hat{\mathbf{s}}(t), \end{aligned}$$

where

$$\begin{aligned} \mathbf{c}_\epsilon^\top &= \mathbf{c}^\top \text{diag}(\alpha_1, \dots, \alpha_n) \mathbf{E}^\top, \\ \mathbf{Q} &= 2 \text{diag}(c_1 \beta_1, c_2 \beta_2, \dots, c_m \beta_m), \\ \mathbf{c}_\gamma^\top &= \mathbf{c}^\top \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_m), \\ \hat{s}_v(t) &= \min \left\{ \bar{s}_v(t), \frac{\bar{h}_v(t) - \zeta}{\eta_v} \right\} \quad \forall v \in V, t \in [0, T]. \end{aligned}$$

Integrating the term involving  $\mathbf{s}$  in the objective function by parts gives

$$\begin{aligned} \int_0^T \mathbf{c}_\epsilon^\top(t) \mathbf{s}(t) dt &= \int_0^T \mathbf{c}_\epsilon^\top(t) dt s(T) - \int_0^T \left[ \int_0^t \mathbf{c}_\epsilon^\top(\tau) d\tau \right] \dot{\mathbf{s}}(t) dt \\ &= 0 + \int_0^T \left[ \int_0^t \mathbf{c}_\epsilon^\top(\tau) d\tau \right] \mathbf{E}\mathbf{x}(t) dt + \text{constant}. \end{aligned}$$

This finally gives us a problem in the form of CQNP described above.

**3.2. An example** As an illustration, we shall present a small numerical example that models the movement of water in a small network over a twenty-four hour period. The topology of the network consisting of four nodes and five arcs is shown in Figure 1.

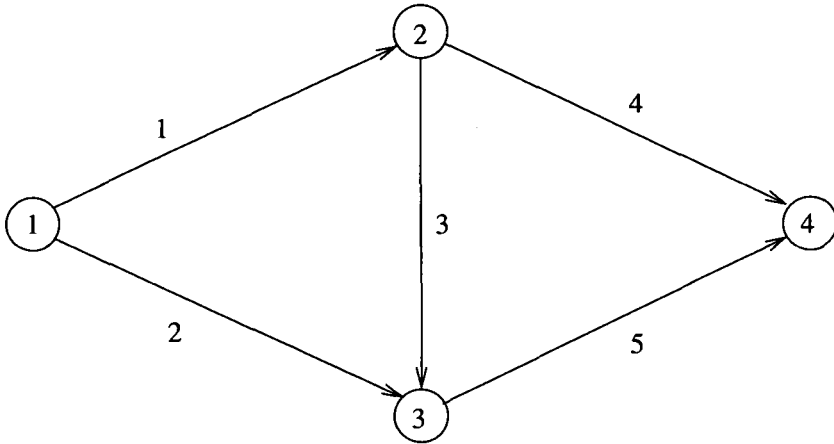


FIGURE 1. The network structure.

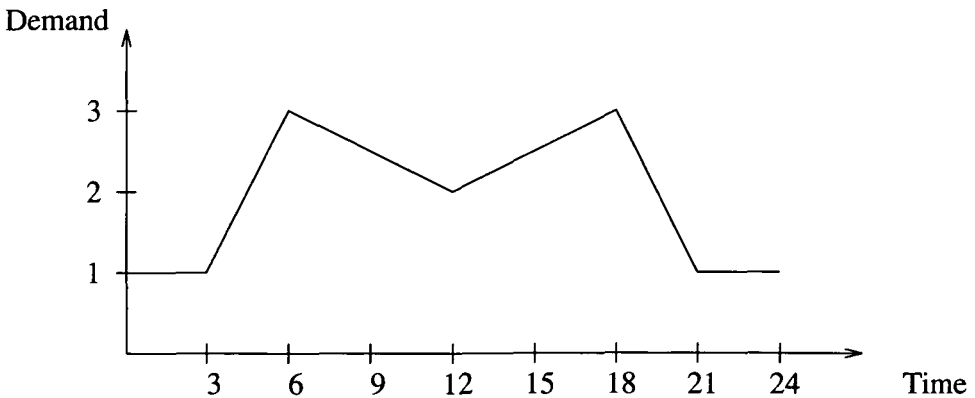


FIGURE 2. Demand function for nodes 3 and 4.

The water supply comes from two sources: node 1 is a ground water pump, supplying water at constant rate of two units per hour and node 2 is a reservoir which

supplies the remaining demand. The reservoir is modeled as a source where the water arrives in the first hour to bring it into the form discussed above. The remaining two nodes both consume water at the same rate. The demand function for nodes 3 and 4 is shown in Figure 2.

The capacities of all of the pipes is 2, except pipe 3 which has a capacity of 3. The ground water pump has no storage capacity, while the reservoir has an unlimited storage capacity. Node 3 can store at most 7 units of water at any particular time, while node 4 has a storage tank of size 15. The relationship between pump generated pressure, pressure difference and flow rate in the pipes is approximated by

$$p_a(t) = 5(h_{\omega(a)}(t) - h_{\alpha(a)}(t)) + 3x_a^2(t) - x_a(t)$$

for each pipe  $a$  and all times  $t \in [0, 24]$ . The units are assumed to be scaled such that the pressure has the same value as the storage in the tanks. Finally the power cost of operating the pumps is \$0.10 per hour per unit of power consumed between 5am and 9pm and only \$0.06 at night.

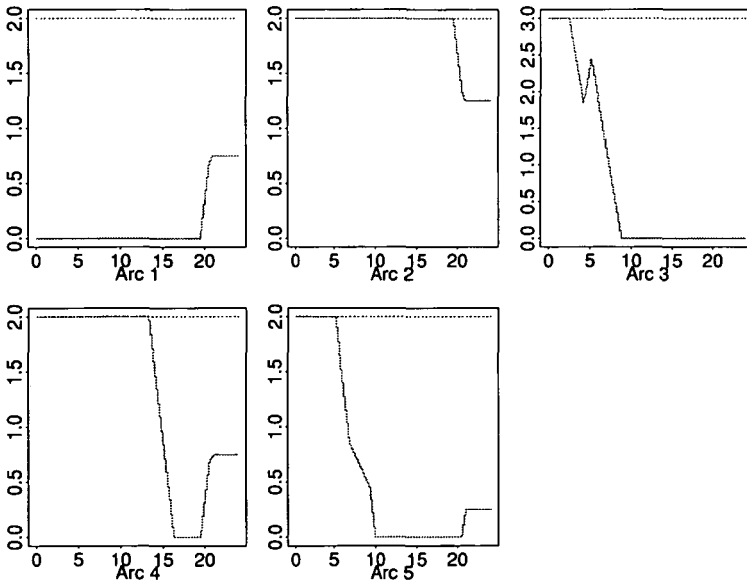


FIGURE 3. Optimal flow in the pipes: the flow is plotted for each arc with time along the  $x$ -axis and the rate of flow along the  $y$ -axis. The top line indicates the bound on the rate of flow, while the bottom line is the actual flow rate.

The appropriately discretized problem was solved by the ASG algorithm resulting in an optimal control with a cost of approximately \$178.75. The optimal flows and

the storage generated by these are shown in Figures 3 and 4.

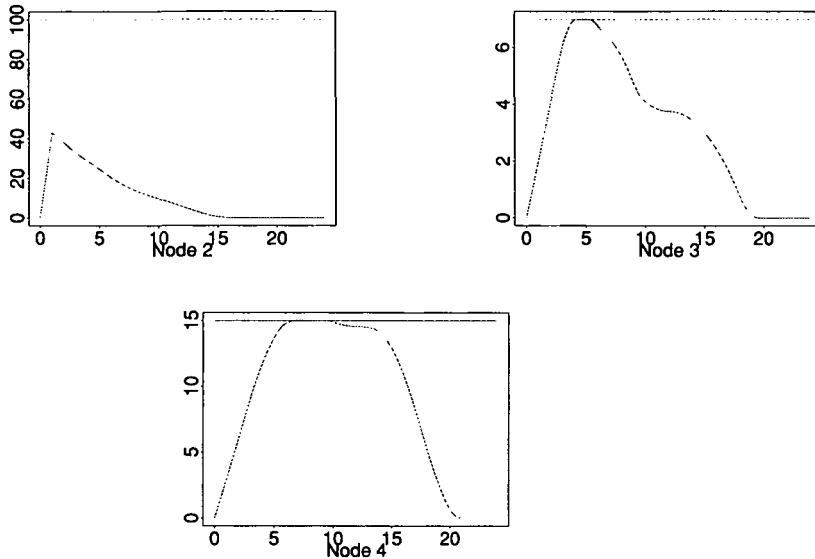


FIGURE 4. Storage in nodes 2 to 4 generated by optimal flows: the storage for each node is plotted with time along the  $x$ -axis and the amount stored along the  $y$ -axis. The capacity of the node is indicated by the top line.

TABLE 1. Numerical results for the pipe network example.

Number of Intervals	Optimal Cost	CPU Time
7	179.61	0.09
15	178.77	0.61
20	178.75	1.19
30	178.76	3.85
40	178.75	7.57
50	178.74	16.39
60	178.75	29.37
144	178.75	359.03

Finally to show that one can get quite accurate results with a reasonably small number of intervals, Table 1 shows the optimal value and running times for different numbers of intervals used to discretize the problem. The running times are from a computer using the Intel 80486DX2 CPU running at 66 MHz. Note that the

running time is purely a function of the number of intervals; it does not depend on the complexity of the problems parameter functions. Of course if, for example, the demand function is highly nonlinear then more intervals are needed to obtain the same accuracy. Hence the amount of computation time required increases. However the objective function is not very sensitive to the number of intervals used, even for examples more nonlinear than the one presented here.

### Acknowledgement

This research was supported by a grant from the Australian Research Council.

### Appendix A: The ASG algorithm

In this section, the Active-Set-on-a-Graph (ASG) algorithm for quadratic cost network flow problems is described. This is a variation of the general active set algorithm for quadratic programming problems specialized for quadratic cost networks. The version of the algorithm presented in this paper is based on a combination of [4, 2] but includes a slightly different treatment of the linear cost arcs. For further details see [3, 4, 2].

The quadratic network program may be stated as:

PROBLEM QNP.

$$\text{Min } \frac{1}{2} \mathbf{x}^T Q \mathbf{x}^T + \mathbf{c}^T \mathbf{x} \quad (20)$$

$$\text{subject to } E \mathbf{x} = \mathbf{b}, \quad (21)$$

$$\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \quad (22)$$

where  $Q = \text{diag}(q_1, q_2, \dots, q_m)$  is a positive semidefinite matrix;  $\mathbf{c}, \mathbf{u}, \mathbf{x} \in \mathbb{R}^m$  and  $\mathbf{b} \in \mathbb{R}^n$ . The basic strategy of the ASG algorithm is to repeatedly fix the flow in some arcs at the upper or lower bound, then find the optimal solution for the remaining arcs without considering the capacity constraints, until the solution to this modified problem is feasible and optimal for the original QNP.

DEFINITION 8. An arc  $a \in A$  is *active* if  $a \in \mathcal{A} \hat{=} \mathcal{A}^+ \cup \mathcal{A}^-$ , where

$$\mathcal{A}^+ \hat{=} \{a \mid x_a = u_a\} \quad \text{and} \quad \mathcal{A}^- \hat{=} \{a \mid x_a = 0\}.$$

and  $\mathcal{A}$  is called the *active set*. The complement  $\mathcal{N} \hat{=} A \setminus \mathcal{A}$  is the *non-active set*.

**A.1 The unbounded subproblem** For any given active set  $\mathcal{A}$ , the ‘Network Quadratic Equality Program’ is solved.

PROBLEM NQEP( $\mathcal{A}$ ).

$$\text{Min } \frac{1}{2} \mathbf{x}^T Q \mathbf{x}^T + \mathbf{c}^T \mathbf{x} \tag{23}$$

$$\text{subject to } E \mathbf{x} = \mathbf{b}, \tag{24}$$

$$x_a = 0 \quad \forall a \in \mathcal{A}^-, \tag{25}$$

$$x_a = u_a \quad \forall a \in \mathcal{A}^+. \tag{26}$$

Before proceeding further it is convenient to introduce some more notation. Let the sets of linear and quadratic cost arcs be denoted by

$$\mathcal{L} \hat{=} \{a \in A \mid q_a = 0\} \quad \text{and} \quad \mathcal{Q} \hat{=} \{a \in A \mid q_a > 0\}.$$

Also define  $\hat{\mathcal{L}} \hat{=} \mathcal{L} \cap \mathcal{N}$  and  $\hat{\mathcal{Q}} \hat{=} \mathcal{Q} \cap \mathcal{N}$ .

By defining  $\hat{b}_n \hat{=} b_n - \sum_{a \in \mathcal{A}^+ : \alpha(a)=v} u_a + \sum_{a \in \mathcal{A}^+ : \omega(a)=v} u_a$ , and dropping the constant terms in the objective function, we arrive at the slightly simpler but equivalent version of the NQEP( $\mathcal{A}$ ).

$$\begin{aligned} \text{Min } & \sum_{a \in \hat{\mathcal{Q}}} \left( \frac{1}{2} q_a x_a^2 + c_a x_a \right) + \sum_{a \in \hat{\mathcal{L}}} c_a x_a \\ \text{subject to } & \sum_{\substack{a \in \mathcal{N} \\ \alpha(a)=v}} x_a - \sum_{\substack{a \in \mathcal{N} \\ \omega(a)=v}} x_a = \hat{b}_v \quad \forall v \in V. \end{aligned}$$

The first order necessary conditions for optimality of this problem are

$$\forall a \in \hat{\mathcal{Q}} \quad q_a x_a + c_a + \pi_{\alpha(a)} - \pi_{\omega(a)} = 0, \tag{27}$$

$$\forall a \in \hat{\mathcal{L}} \quad c_a + \pi_{\alpha(a)} - \pi_{\omega(a)} = 0, \tag{28}$$

$$\forall v \in V \quad \sum_{\substack{a \in \mathcal{N} \\ \alpha(a)=v}} x_a - \sum_{\substack{a \in \mathcal{N} \\ \omega(a)=v}} x_a = \hat{b}_v, \tag{29}$$

where  $\pi_v$  is the dual variable associated with the flow conservation constraint at node  $v$ , called the *node potential*. From (28) we can see that analogous to the purely linear network flow problem the following condition holds.

**PROPOSITION 1.** If  $\mathbf{x}$  is optimal then there does not exist a cycle  $C \subseteq \mathcal{L}$  such that  $0 < x_a < u_a \forall a \in C$ .



In other words, the set  $\mathcal{L}$  determines a forest in the graph. For each tree in the forest we can pick an arbitrary root node  $r$ . For each node  $v$  in the same tree let  $P_v$  be the unique path from  $r$  to  $v$ , then the potential of  $v$  is just the potential of the root node plus some constant,

$$\pi_v = \pi_r + \sum_{a \in P_v^+} c_a - \sum_{a \in P_v^-} c_a. \tag{30}$$

Equation (30) can now be substituted into (27). Also adding together all the equations (29) corresponding to nodes of the same tree will eliminate all  $x_a$  joining two nodes in the same tree. The result of these operations is to contract each tree of linear arcs into a single node, to arrive at a new network which has only strictly convex arcs. That is we have a network  $(\bar{V}, \bar{A})$  with each node representing a (possibly trivial) tree in the original network. Each arc in  $\bar{A}$  corresponds to an arc in  $\hat{\mathcal{Q}}$  but the linear cost coefficient may be different. Let  $\bar{E}$  be the node-arc incidence matrix for this new network,  $\bar{x}$  the components of  $x$  corresponding to arcs joining two different trees etc. Then the optimal flows and potentials satisfy

$$\bar{Q}\bar{x} + \bar{c} + \bar{E}^T \bar{\pi} = \mathbf{0}, \tag{31}$$

$$\bar{E}\bar{x} = \bar{b}. \tag{32}$$

Substituting for  $\bar{x}$  in (32) using (31) gives

$$\bar{E}\bar{Q}^{-1}\bar{E}^T \bar{\pi} = -\bar{b} - \bar{E}\bar{Q}^{-1}\bar{c}. \tag{33}$$

Since  $\bar{Q}^{-1}$  is positive definite, we have  $\text{rank}(\bar{E}\bar{Q}^{-1}\bar{E}^T) = \text{rank}(\bar{E}) = |\bar{V}| - 1$  provided  $(\bar{V}, \bar{A})$  is connected. Hence if  $\mathcal{N}$  spans  $V$ , (33) can be solved for  $\bar{\pi}$  once one of the values has been set arbitrarily. This gives the potential of all the root nodes from which the remaining potentials and the arc flows can be found using (30), (27) and (29).

**DEFINITION 9.** The difference in potential across arc  $a$  is called the *tension*  $\tau_a \hat{=} \pi_{\omega(a)} - \pi_{\alpha(a)}$ .

**A.2 The algorithm** The optimality conditions for the original problem QNP consist of the optimality conditions of NQEP( $\mathcal{A}$ ) plus (22) and the inequalities  $\tau_a \leq c_a \forall a \in \mathcal{A}^-$  and  $\tau_a \geq q_a u_a + c_a \forall a \in \mathcal{A}^+$ . All the algorithm does is to swap all arcs which don't satisfy these conditions for any particular solution of NQEP( $\mathcal{A}$ ) between  $\mathcal{A}$  and  $\mathcal{N}$  to produce a new active set until the solution of NQEP( $\mathcal{A}$ ) is optimal for QNP. Unfortunately this process is complicated slightly, since one always needs to ensure that  $(V, \mathcal{N})$  is connected and  $\hat{\mathcal{L}}$  contains no cycles, so that the subproblem NQEP( $\mathcal{A}$ ) can be solved. The outline of the ASG algorithm is as follows.

**Algorithm 1**

```

Find a spanning forest of  $(N, \mathcal{L})$ , denote its arcs by  $S(\mathcal{L})$ ;
Set  $\mathcal{A} \leftarrow S(\mathcal{L})$ ;
Solve NQEP( $\mathcal{A}$ ) to get primal-dual solution  $\mathbf{x}, \pi$ ;
while  $\mathbf{x}, \pi$  is not optimal do
  for all arcs  $a$  with  $x_a = 0$  &  $\tau_a > 0$ , or  $x_a = u_a$  &  $\tau_a < q_a u_a$  do
    if  $a \in \mathcal{L}$  and  $\hat{\mathcal{L}} \cup \{a\}$  contains a cycle then
      Pick some arc  $b \neq a$  in the cycle;
       $\mathcal{A} \leftarrow \mathcal{A} \cup \{b\}$ ;
    endif
     $\mathcal{A} \leftarrow \mathcal{A} \setminus \{a\}$ ;
  next  $a$ 
  for all arcs  $a$  with  $x_a > u_a$  do
    if  $\mathcal{N} \cup \{a\}$  does not span  $V$  then
      Find a cut  $\mathcal{C} = \mathcal{C}^+ \cup \mathcal{C}^-$  with  $a \in \mathcal{C}^+$  and  $\mathcal{C} \setminus \{a\} \subseteq \mathcal{A}$ ;
      if  $\exists b \in (\mathcal{A}^+ \cap \mathcal{C}^-) \cup (\mathcal{A}^- \cup \mathcal{C}^+)$  then  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{b\}$ ;
      else STOP /* the problem is infeasible */
    endif
     $\mathcal{A}^+ \leftarrow \mathcal{A}^+ \cup \{a\}$ 
  next  $a$ 
  for all arcs  $a$  with  $x_a < 0$  do
    if  $\mathcal{N} \cup \{a\}$  does not span  $V$  then
      Find a cut  $\mathcal{C} = \mathcal{C}^+ \cup \mathcal{C}^-$  with  $a \in \mathcal{C}^+$  and  $\mathcal{C} \setminus \{a\} \subseteq \mathcal{A}$ ;
      if  $\exists b \in (\mathcal{A}^+ \cap \mathcal{C}^+) \cup (\mathcal{A}^- \cup \mathcal{C}^-)$  then  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{b\}$ ;
      else STOP /* the problem is infeasible */
    endif
     $\mathcal{A}^- \leftarrow \mathcal{A}^- \cup \{a\}$ ;
  next  $a$ 
  Resolve NQEP( $\mathcal{A}$ ) for the new  $\mathbf{x}$  and  $\pi$ ;
wend
end algorithm 1.

```

REMARK 6. There are two places in the above algorithm where an arc has to be chosen to be (de)activated so that the conditions on the active set hold. In both cases a random choice could lead to cycling. For the choice of linear arcs to activate, the obvious method is to go around the cycle of linear cost arcs in the direction of flow increase, and pick the arc which has the smallest capacity. Using this rule the above algorithm reduces to the network simplex algorithm for purely linear problems.

The choice of arc to deactivate, to prevent a cut, can be made similarly by choosing the arc which is closest to the dual variable bound  $\tau_a = c_a$ . Note that although this

method works in practice, a somewhat more complicated version of this algorithm is required to allow a proof of convergence. However that is beyond the scope of this paper.

## References

- [1] E. J. Anderson and A. B. Philpott, "A continuous-time network simplex algorithm", *Networks* **19** (1989) 395–425.
- [2] N. L. Boland, A. T. Ernst, C. J. Goh and A. I. Mees, "A faster version of the ASG algorithm", *Appl. Math. Letters* **7** (1994) 23–27.
- [3] N. L. Boland, C. J. Goh and A. I. Mees, "An algorithm for solving quadratic network flow problems", *Appl. Math. Letters* **4** (1991) 61–64.
- [4] N. L. Boland, C. J. Goh and A. I. Mees, "An algorithm for non-linear network programming: implementation, results and comparisons", *J. Opl. Res. Soc.* **42** (1992) 979–992.
- [5] C. A. Brebbia and A. J. Ferrante, *Computational hydraulics* (Butterworths, London, 1983).
- [6] R. N. Buie and J. Abrham, "Numerical solutions to continuous linear programming problems", *Zeitschrift für Oper. Res.* **42** (1973) 107–117.
- [7] M. Collins, L. Cooper, R. Helgason, J. Kennington and L. Le Blanc, "Solving pipe network analysis problems using optimization techniques", *Mang. Sci.* **24** (1978) 747–760.
- [8] L. R. Ford and D. R. Fulkerson, *Flows in networks* (Princeton University Press, Princeton, 1962).
- [9] H. M. Markowitz, *Portfolio selection: efficient diversification of investments* (Wiley, New York, 1959).
- [10] M. Pullan, *Separated continuous linear programs: theory and algorithms*, Ph. D. Thesis, Judge Institute of Management Studies, University of Cambridge, 1992.
- [11] K. L. Teo, C. J. Goh and K. H. Wong, *A unified computational approach to optimal control problems* (Wiley, New York, 1991).