Hydraulic Network Analysis Using (Generalized) Geometric Programming

M. A. Hall
Nova Scotia Technical College
Halifax, Nova Scotia

ABSTRACT

The problem of determining the flows (and pressures) in the pipes of a general hydraulic network, for given input and output flows and/or given input and output pressure heads, is shown to be equivalent to either of a pair of convex programming problems with linear constraints. This is accomplished via the theory of (generalized) Geometric Programming. The equivalence of these problems is exploited to prove existence and uniqueness of the flow solution under certain conditions as well as to derive an algorithm which calculates this solution. Computational aspects of implementing the algorithm are considered in some detail and results obtained for a general example problem are presented. A brief discussion of the application of the methods of the paper to problems in electrical network analysis, transportation network analysis and the elastic analysis of structural trusses, is also given.

1. INTRODUCTION

It has long been known that the determination of equilibrium currents and voltages in some electrical networks can be reduced to the solution of certain minimization problems (Duffin [3]). In this paper we consider the problem of determining flows and pressures in a fluid pipe distribution network when inlet and outlet flow rates and/or pressure heads are known, from this viewpoint.

In order to solve the problem in question it is necessary to obtain a set of flows in the pipes which simultaneously satisfy two sets of equations. The first is the set of linear equations which specify that, at each junction of the network, the inflow equals the outflow. The second is the (usually)

Networks, 6: 105-130

nonlinear set which results from substituting the nonlinear expression for pressure difference between the ends of a pipe in terms of flow, into the set of linear equations which specify that the pressure change around each closed loop of pipes is zero. It is the nonlinearity of the second set that makes explicit solution of the equation system impractical, so that all widely used methods are iterative in nature.

In this paper we show that if a certain function of the pipe flows is minimized subject to the condition that the node laws are satisfied then the minimizing flows satisfy the loop laws automatically. (There is a dual principle, that the minimizing point of a certain function of the pressure differences between pipe ends satisfies the node laws, when the minimization is done subject to the condition that the pressure differences satisfy the loop laws. Although it is advantageous to do so in many cases we shall not explicitly make use of this fact in this paper.) This allows us to prove existence and uniqueness of the flow solution under rather general conditions. We then give an algorithm which solves the problem exploiting this principle. We describe our objectives in more detail after a precise formulation of the problem is given in section 3.

A variational principle similar to the one we give, which characterizes the flows and pressures that satisfy the governing equations of the system has been described by Prager [13]. However, the characterization is much less complete than that which we give. Also Prager did not consider his principle from the viewpoint of modern mathematical programming theory as we have done and, in addition, did not explicitly consider computational aspects of using the principle. Our development of the principles mentioned is somewhat similar to the approaches taken by Minty [10], Berge [2] and Rockafeller [14] in their study of "monotone networks." However, like Prager, none of these authors considered directly exploiting the variational principles they developed to compute solutions to the problem, and all required a great deal of work in carrying out their deductions since they did not make use of an established body of theory as we do. In addition, they did not show precisely how to apply their results to actual problems in hydraulic network analysis, which we do.

To complete this section we give an outline of some currently used methods of solving the problem. Most of the literature on hydraulic network analysis is explicitly concerned with the case where the fluid flowing in the pipes is water at ambient temperature. The references mentioned in the outline of current methods of solving the problem, all fall into this category.

Traditionally the problem has been solved using the Hardy-Cross technique [1]. One way of doing this involves making an initial estimate of the flows which satisfy the node equations and then correcting the flow in each pipe, in order to bring about satisfaction of the loop equations. The procedure involves correcting flows in the pipes, one loop at a time. This makes convergence of the method (when it occurs) rather slow. A similar method which instead corrects the pipe flows by considering all loops simultaneously is due to Epp and Fowler [4]. This method results in much more rapid convergence. The Epp and Fowler algorithm makes fundamental use of the Newton Raphson method of solving nonlinear equations, a technique which also forms the basis for several other popular algorithms, all of which operate by iteratively solving systems of equations (e.g., [8], [9], [16]). Wood and Charles [17] recently proposed a solution procedure that involves successively linearizing the loop equations. However, the method essentially involves repeatedly inverting large matrices - a task which is usually quite cumbersome computationally.

2. PRELIMINARY DEFINITIONS

For completeness, we set out at this stage some definitions that will be used in subsequent sections. Apart from the definitions of the matrices B and G, the reader who is unfamiliar with them may skip over the others without serious loss.

Definition: The incidence matrix B of a general directed network (i.e., having directed links) with nodes numbered 1, ..., m and links numbered 1, ..., n is an $m \times n$ matrix whose j^{th} row B_j has the following form:

B_{ji} = +1 if link i is incident at node j and has a direction towards j.

= -1 if link i is incident at node j and has a direction away from j.

= 0 if link i is not incident at node j.

Definition: A circuit of an m-node, n-link directed network is a set of links joined end-to-end with the additional property that every node at which a link of the set is incident has precisely two links of the set incident at it.

Each circuit has an n-vector representation v which is defined as follows. The direction of the lowest numbered link j which is in the circuit defines the orientation of the circuit as clockwise or counterclockwise. The entries of v are then

v_j = +1 if link j is in the circuit and has the same orientation as the circuit.

= -1 if link j is in the circuit and has the opposite orientation to that of the circuit.

= 0 if link j is not in the circuit.

Definition: The fundamental circuit matrix G of a general directed network with m nodes and n links is closely related to the incidence matrix B and is constructed as follows:

First form any tree of the network (a tree is a set of links with the property that it contains a unique path between each pair of nodes in the network; it can be shown that each tree has precisely m-1 links) and order the indices of the non-tree links so they correspond to the integers 1, 2, ..., n-m+1. The links of the tree are ordered so they correspond to the integers n-m+2, ..., n. The j^{th} row of the matrix G_j is defined as follows. Consider the path through the tree joining the initial and terminal node of the nontree link corresponding to the integer j, for each $j \in \{1,2,\ldots,n-m+1\}$ and let

G_{ji} = +1 if i corresponds to j or if link i is on this path and has the same orientation as the link corresponding to j.

= -1 if link i is on this path and has the opposite orientation to that of the link corresponding to j.

= 0 if link i is not on this path.

It can be shown that each circuit of the network is a linear combination of the rows G_j of G (that is, the vector representation v of any circuit may be written

$$v = \sum_{j=1}^{n-m+1} a_j G_j$$

where the a_j are the real numbers).

It is well-known [2] that the vector spaces formed by the rows of B and G are orthogonally complementary. The dimension of the row space of B is m-1 and that of G is n-m+1. It can be shown that any one of the rows of B, as we have defined it, may be omitted to give it full row rank. We assume henceforth that this has been done.

It can also be shown that the vector spaces formed by the set of solutions u to $Bu = 0$, and the set of solutions v to $Gv = 0$ (known as the "null spaces" of B and G) are orthogonally complementary. We shall use this fact quite often.

Definition: The epigraph of the real valued function $w(\cdot)$ with domain W, a subset of Euclidean n - space (written $W \subseteq E_n$), is the set

$$\text{epi } w = \{(z,r) \in E_{n+1}: z \in W \text{ and } w(z) \leq r\}.$$

Definition: A function is said to be closed if its epigraph is topologically closed (contains all its limit points).

Definition: The affine hull of an arbitrary set S in E_n, denoted aff S, is the set of points of the form

$$\delta_1 s^1 + \delta_2 s^2$$

where $s^1, s^2 \in S$ and δ_1, δ_2 are real numbers whose sum is 1.

Definition: The relative interior of an arbitrary set S in E_n, denoted ri S, is the interior of S relative to aff S.

3. FORMULATION OF THE PROBLEM

We represent the system of pipes and junctions as a directed network. The nodes numbered 1, 2, ..., m represent the junctions while the links numbered r+1, ..., n represent the pipes (the purpose of starting the enumeration at r+1 will be explained later). Each link is assigned an arbitrary positive direction with flow along the link in this direction being considered positive, while flow in the opposite direction is considered negative. For ease in formulating the problem we add to the network a total of r fictitious links and one fictitious node. Each fictitious link is incident at the fictitious node which we number zero. The first $q(\leq r)$ fictitious links are also incident at the points of the network with known inflow rate. The direction of link j, $j \in \{1,...,q\}$ is towards (away from) node zero if there is positive outflow (inflow) at its point of contact with the real part of the network. The fictitious links q+1, q+2, ..., r all have a direction away from node zero and are incident at the points of the network where pressure head is known (relative to some datum). In the example of Figure 1, q=2, r=4 and n=11. For the network illustrated there is a fixed rate of supply at node 4, and a fixed drawoff rate at node 5. In addition, fluid is supplied at fixed pressure heads at nodes 1 and 2, while node 3 is neither a point of inflow or outflow.

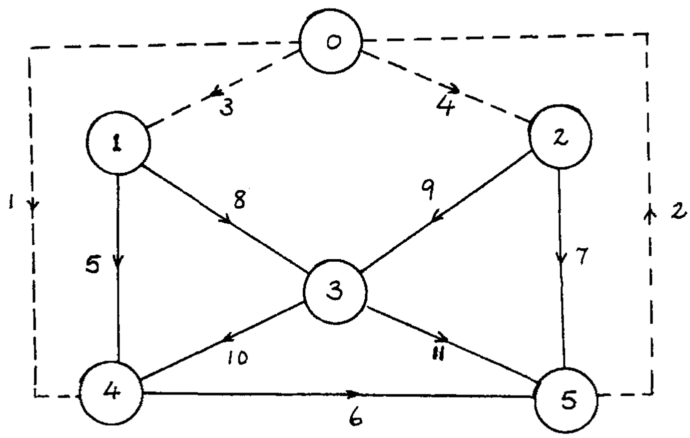

Fig. 1

Let x_i denote the flow on link i, i = 1, ..., n and let B denote the m × n incidence matrix for the whole network. Then the conservation of flow equations for the network may be written

$$Bx = 0$$

$$x_i = F_i \qquad i = 1, 2, \ldots, q \tag{1}$$

where F_i is the known inflow or outflow corresponding to link i. Let p_i be the negative of the change in pressure between the ends of link i, measured in the direction of the links arrow (i.e., the pressure "drop" along link i). Let G' denote the fundamental circuit matrix for the network consisting of nodes 0, 1, ..., m and links q+1, ..., r, r+1, ..., n then the loop laws for the network may be written

$$G'p' = 0$$

$$p_i = P_i \qquad i = q+1, \ldots, r \tag{2}$$

where $p' = (p_{q+1}, p_{q+2}, \ldots, p_r, p_{r+1}, \ldots, p_n)$ and $-P_i$ is the known pressure head corresponding to (the terminal node of) link i for i = q+1, ..., r. For each real link i the pressure change p_i is assumed to be related to the flow x_i by the equation

$$p_i = f_i(x_i) \tag{3}$$

where $f_i(\cdot)$ is a continuous monotone increasing function with domain and range consisting of the whole real line E_1 and with zero value at $x_i = 0$.* For example, the formula

$$f_i(x_i) = \begin{cases} +r_i\, x_i^{a_i} & 0 \le x_i \\ -r_i\, x_i^{a_i} & x_i \le 0 \end{cases}$$

where r_i is a positive constant (depending on the geometry and material of the pipe) and a_i is a positive constant close in value to 2.0, is commonly used when the fluid being considered is water at ambient temperature. The problem may now be summarized as that of finding a vector of flows $x' = (x_1, x_2, \ldots, x_q, x_{q+2}, \ldots, x_r, x_{r+1}, \ldots, x_n)$ and pressures $p' = (p_{q+1}, p_{q+2}, \ldots, p_r, p_{r+1}, \ldots, p_n)$, that satisfy relations (1), (2), and (3).†

The reader may now recognize the similarity of the problem to that of determining currents and voltages in certain electrical networks. For example, the problem of analyzing a network of nonlinear resistors with constant current and voltage inputs is seen to be almost identical to the hydraulic problem posed above, when one recognizes that the flow rates play the role of

The results to be given are valid under much weaker assumptions in the form of the functions $f_i(\cdot)$ than those made here. However, for the vast majority of problems of practical interest the assumptions we make are satisfied; extending the results to the more general case is not difficult.

†*We are not specifically interested in the values of x_i, $i = q+1, \ldots, r$ but they will automatically be determined by relations (1) once the values of the other x_i are known.*

currents, the pressures that of voltages and the relations (3), that of Ohms Law. The developments of sections 5 and 6 below may be directly interpreted in terms of electrical network analysis in the light of these observations. The same approach can be used to study problems in the analysis of transportation networks [6], and of structural trusses [7]. In the transportation case the analogues of pressure and flow are cost of travel and traffic flow rate while in the truss case they are displacement and force.

To conclude this section we outline the contents of the remainder of the paper. First we show that under some very weak conditions there does in fact exist a set of flows for the real links of the network that satisfy relations (1), (2) and (3); furthermore, this set is unique. Again, under reasonable conditions, we show that the solution to (1), (2) and (3) can be obtained by solving a convex programming problem with linear constraints. We then give an algorithm which solves this convex programming problem, taking advantage of its special (network induced) nature and discuss its convergence properties. We consider its efficiency and give the solution of an example problem obtained using it.

In the next section we outline some results from the theory of Geometric Programming which we use in subsequent sections. Most of these results can be established directly using the definitions to be given but a much more thorough treatment of the subject is given by Peterson [11].

4. (GENERALIZED) GEOMETRIC PROGRAMMING

Consider the following optimization problem:

Problem A: Using the "feasible solution" set

$$S = X \cap C$$

calculate both the "problem infimum"

$$\phi = \inf_{x \in S} g(x)$$

and the "optimal solution" set

$$S^* = \{x \in S: \phi = g(x)\}.$$

The set X is assumed to be a vector subspace of E_n and the set C is the domain of a closed convex function $g(\cdot)$.

A second problem (problem B), which has the same form as Problem A, is defined in terms of the following entities.

The set Y is taken to be the orthogonal complement in E_n of X. The function h(•) with domain

$$D = \{y \in E_n : \sup_{x \in C} [\langle x,y \rangle - g(x)] < +\infty\}$$

and functional values

$$h(y) = \sup [\langle x,y \rangle - g(x)]$$

is termed the "conjugate transform" of g:C.

The function g:C and its conjugate transform h:D are usefully related by the "conjugate inequality" which is stated partly in terms of the "subgradient set" of g(•) at x, i.e.

$$\partial g(x) = \{z \in E_n : g(x) + \langle z, w-x \rangle \leq g(w) \text{ for each } w \in C\}.$$

Lemma 1: ("Conjugate inequality") For each $x \in C$ and each $y \in D$

$$\langle x,y \rangle \leq g(x) + h(y)$$

with equality holding if and only if $y \in \partial g(x)$.

Problem B: Using the "feasible solution" set

$$T = Y \cap D$$

calculate both the "problem infimum"

$$\psi = \inf_{y \in T} h(y)$$

and the "optimal solution" set

$$T^* = \{y \in T: \psi = h(y)\}.$$

Problem B is termed the "(geometric) dual" of problem A.

Dual pairs of geometric programming problems have properties similar to those for dual pairs of linear programming problems. In particular we have the following theorems that are stated again in terms of the subgradient set.

Theorem 1: If x and y are feasible for the dual problems A and B respectively, then

$$0 \leq g(x) + h(y)$$

with equality holding if and only if

$$y \in \partial g(x).$$

in which case x and y are optimal solutions to A and B respectively.

Corollary 1: If dual problems A and B both have feasible solutions then their infima ϕ and ψ are both finite and

$$0 \leq \phi + \psi.$$

Dual problems A and B, which have nonempty feasible solution sets S and T for which $0 < \phi + \psi$, are said to have a *duality gap* of $\phi + \psi$. Duality gaps are undesirable from both a theoretical and a computational point of view, but fortunately we shall be able to show that they do not occur in the present† geometric programming formulation of the problem under consideration.

For those dual problems A and B that do not have a duality gap, Theorem 1 provides a useful characterization of dual optimal solutions x^* and y^* in terms of the following extremality conditions

$$\text{(I)} \quad x \in X,\ y \in Y$$

$$\text{(II)} \quad y \in \partial g(x).$$

We formalize this characterization as the following corollary.

Corollary 2: Suppose that dual problems A and B have feasible solutions and do not have a duality gap. Then arbitrary vectors x and y are optimal solutions to A and B respectively if and only if x and y satisfy the extremality conditions I and II.

The most important result in this paper makes use of Corollary 2, and thus it is important to be sure that the hypotheses of this corollary hold. We discuss verification of these hypotheses in section 5. In order to show that no duality gap exists we shall apply the following theorem of Fenchel [11].

---

†Other geometric programming formulations of the problem are possible.

Theorem 2: If problem A has a feasible solution x_0 in ri C and a finite infimum ϕ, then the optimal solution set T^* is not empty and $0 = \phi + \psi$.

The theorem is equally valid in its dual form with problem A replaced by problem B, the set C replaced by the set D, and the set T^* by the set S^*. Application of this theorem will provide an additional piece of information in that together with the main theorem to be proved, it will assert the existence of a solution to our problem under some fairly weak assumptions.

For a complete development of the theory given in this section see [11].

5. A CHARACTERIZATION OF THE SOLUTION SET OF RELATIONS (1), (2), AND (3)

In order to use the theory of section 4 we must make some definitions.

For each link i of the network we define a function $g_i(\cdot)$ with domain C_i as follows:

For $i = 1, \ldots, q$

$$g_i(x_i) = 0 \text{ for each } x_i \in C_i = \{F_i\};$$

for $i = q+1, \ldots, r$

$$g_i(x_i) = P_i x_i \text{ for each } x_i \in C_i = E_1,$$

and for $i = r+1, \ldots, n$

$$g_i(x_i) = \int_0^{x_i} f_i(t)\,dt \text{ for each } x_i \in C_i = E_1.$$

The integrals of the last equation, which are defined in the sense of Riemann, exist by virtue of the assumed monotonicity of the functions $f_i(\cdot)$. In addition, the functions $g_i{:}C_i$ as defined can all be shown to be closed and convex on their domains and hence the function $g{:}C$ defined so that

$$g(x) = \sum_1^n g_i(x_i) \text{ for each } x \in C = \mathop{\times}_1^n C_i$$

is closed and convex (see [15]). Here the symbol "X" denotes cartesian product. Now consider the optimization problem

$$\text{infimize } g(x)$$
$$\text{subject to } x \in N(B) \cap C \qquad (4)$$

where N(B) denotes the null-space of matrix B. It is clear that problem (4) is a special case of problem A of the last section. Notice also that the constraints of (4) are equivalent to relations (1).

In the proof of the main results of this paper we make use of Corollary 2; certain assumptions are needed to ensure that its hypotheses are satisfied. To begin with we assume that there is a solution to relations (1); the equivalence of relation (1) and the constraints of problem (4) then guarantee the consistency of problem A (problem (4)). Before we discuss consistency of problem B we prove the following lemmas.

Lemma 2: The feasible solution set for the geometric dual problem of (4) is the intersection of the null-space for the fundamental circuit matrix G of the full network with the set

$$\{p \in E_n\colon p_i = P_i \; i = q+1, \ldots, r\}.$$

Proof: The definition of conjugate transform shows that the function $g:C$ of problem (4) has a conjugate transform of the form $h:D$.

$$D = \mathop{\chi}_{1}^{n} D_i$$

$$h(y) = \sum_{1}^{n} h_i(y_i) \text{ for each } y \in D$$

where $h_i:D_i$ is the conjugate transform of $g_i:C_i$.

For each $i = 1, \ldots, q$

$$D_i = \{y_i \in E_1\colon \sup_{x_i \in \{F_i\}} [x_i \cdot y_i - 0] < \infty\} = E_1.$$

For each $i = q+1, \ldots, r$

$$D_i = \{y_i \in E_1\colon \sup_{x_i \in E_1} [x_i \cdot y_i - P_i \cdot x_i] < \infty\},$$

and clearly $\sup_{x_i \varepsilon E_1} [x_i \cdot y_i - P_i \cdot x_i]$ which can be rewritten

$$\sup_{x_i \varepsilon E_1} [x_i \cdot (y_i - P_i)]$$

is finite if and only if $y_i = P_i$, hence

$$D_i = \{P_i\}.$$

For each $i = r+1, \ldots, n$, D_i consists of E_1. To see this note that

$$\partial g_i(x_i) \subseteq D_i \text{ for each } x_i \; \varepsilon \; C_i,$$

-- a consequence of the definitions of subgradient set and conjugate transform. Also at each point x_i where $g_i(\cdot)$ is differentiable

$$\partial g_i(x_i) = \{g_i'(x_i)\} = \{f_i(x_i)\},$$

(see [15]), thus the fact that the range of $f_i(\cdot)$ is E_1 implies

$$E_1 \subseteq D_i,$$

so that D_i is in fact equal to E_1.

To complete the proof, recall (from section 2) that the orthogonal complement of N(B) is N(G) so that the required dual feasible solution set is

$$N(G) \cap D = N(G) \cap \{p \; \varepsilon \; E_n : p_i = P_i \;\; i = q+1, \ldots, r\}$$

as claimed.

Lemma 3: If the (n-q)-vector $p^1 = (p_{q+1}, \ldots, p_r, p_{r+1}, \ldots, p)$ *satisfies relations (2), then there exist numbers* $p_1, p_2, \ldots, p_q$ *such that the n-vector* $p = (p_1, \ldots, p_q, p_{q+1}, \ldots, p_r, p_{r+1}, \ldots, p_n)$ *satisfies the constraints of the dual problem of (4).*

Proof: The fact that $p^1 \varepsilon N(G')$ means that if we set the pressure head to zero at node zero there is a unique pressure head associated with each node of the network (without links 1, 2, ..., q) and the components of p^1 represent differences between these pressure heads (see [2]). For each $i = 1, \ldots, q$ we set p_i equal to plus or minus the pressure head at the node where link i is in contact with the real part of the network. The minus sign applies if link i has a direction away from node zero and the plus if the direction is toward node zero. The resulting vector p clearly has the property claimed in the light of the result of Lemma 2. This completes the proof.

Lemmas 2 and 3 show that consistency of the dual problem of (4) is assured if we assume that there exists a solution to relations (2); this we now do. Notice that this assumption is of the same type as the assumption that relations (1) possess a solution and is no more restrictive. To show that problem (4) and its geometric dual have zero duality gap we use Theorem 2 of the last section.

Lemma 4: Suppose that the sets of relations (1) and (2) both possess solutions, then problem (4) and its geometric dual possess optimal solutions and have a duality gap of zero.

Proof: Observe that

$$\text{ri } C = C.$$

This equation follows from the definition of relative interior, and from a theorem in convex analysis which states that the relative interior of the cartesian product of a group of sets is the cartesian product of the relative interiors of the individual sets. Thus the assumed existence of a solution to relations (1) implies consistency of problem (4), which in turn implies consistency in the sense required by Theorem 2. Also, Corollary 1 together with our hypotheses, ensures that ϕ is finite. These facts alone imply the absence of a duality gap and that the dual optimal solution set is not empty. Nonemptiness of the optimal solution set of problem (4) itself follows from the dual version of Theorem 2. Lemma 2 shows that by analogy with the case for the set C, ri D = D so the consistency of the dual of problem (4) implies consistency in the sense required by Theorem 2. Also, Corollary 1 implies finiteness of ψ since problem (4) is consistent. This completes the proof.

We are now in a position to state and prove the main theorems of this paper.

Theorem 3: Let the n-vector $x = (x_1, x_2, \ldots, x_n)$ *and the (n-q)-vector* $p^1 = (p_{q+1}, p_{q+2}, \ldots, p_n)$ *satisfy relations (1), (2), and (3). Then there exist numbers* $p_1, p_2, \ldots, p_q$*, such that the n-vector x and the augmented n-vector* $p = (p_1, p_2, \ldots, p_q, p_{q+1}, \ldots, p_n)$ *satisfy the extremality conditions (I) and (II) associated with problem (4) and its geometric dual.*

Proof: Suppose that (x, p^1) satisfy (1), (2) and (3). Then immediately we see that x is feasible for problem (4), i.e., $x \in X \cap C$. Also the vector p as constructed in Lemma 3 is feasible for the dual of problem (4), i.e., $p \in Y$. It can easily be verified that the condition

$$p \in \partial g(x)$$

is, in the case where $g{:}C$ has the form $\sum_1^n g_i : \chi_1^n C_i$, equivalent to the conditions $p_i \in \partial g_i(x_i)$, $i = 1, \ldots, n$.

Recall from the proof of Lemma 2 that for $i = r+1, \ldots, n$

$$p_i \in \partial g_i(x_i) \text{ if and only if } p_i = f_i(x_i),$$

and for $i = q+1, \ldots, r$

$$p_i \in \partial g_i(x_i) \text{ if and only if } p_i = P_i$$

the last assertion following from the fact that $g_i'(x_i) = P_i$ for each $x_i \in C_i$.

Note also that for $i = 1, \ldots, r$, C_i is a singleton and

$$\partial g_i(x_i) = \partial g_i(F_i) = E_1$$

so effectively $p_i \in \partial g_i(x_i)$ in this case constitutes no restriction on the value of p_i. Thus it is clear that $p \in \partial g(x)$. This concludes the proof.

Theorem 4: Suppose a pair of n-vectors (x,p) satisfies the extremality conditions (I) and (II) associated with problem (4) and its geometric dual, then the vectors $x = (x_1, x_2, \ldots, x_n)$ *and* $p^1 = (p_{q+1}, p_{q+2}, \ldots, p_n)$ *satisfy relations (1), (2) and (3).*

Proof: The conditions $x \in X$ and $p \in \partial g(x)$ imply that relations (1) and (3) are satisfied and that $p_i = P_i$ for each $i = q+1, \ldots, r$. The latter fact, together with that fact that $p \in Y = N(G)$, implies that p^1 satisfies (2). This concludes the proof.

Theorems 3 and 4 essentially imply that the extremality conditions for problem (4) and its geometric dual and the relations (1), (2) and (3) that characterize the "equilibrium" set of flows and pressures are equivalent. In particular we have the following key result.

Theorem 5: If relations (1) and (2) possess solutions individually, then relations (1), (2) and (3) simultaneously possess a solution. Moreover, each optimal solution $x^* = (x_1^*, x_2^*, \ldots, x_r^*, x_{r+1}^*, \ldots, x_n^*)$ *to problem (4) produces a solution to relations (1), (2) and (3),* [*in the sense that* x^* *and* $p^1 = (P_{q+1}, \ldots, P_r, f_{r+1}(x_{r+1}^*), \ldots, f_n(x_n^*))$ *satisfy (1), (2) and (3)*] *and conversely. Furthermore, the (flows)* x_i^* *for the real links are unique.*

Proof: Lemma 4 shows that there exist optimal solutions to problem (4) and its geometric dual and that there is no duality gap. Hence there exist pairs of n-vectors (x^*, p^*) which satisfy the extremality conditions for these problems and each vector x^* is optimal for problem (4). Theorem 4 then shows that x^* produces a solution to relations (1), (2) and (3). This proves the existence of solutions to (1), and (2) and (3) and shows that each optimal solution to problem (4) provides such a solution. To show the converse of the latter statement we note that via Theorem 3 each solution to (1), (2) and (3) provides a pair of vectors (x^*, p^*) which satisfy the extremality conditions. Each such x^* is optimal for problem (4) by Corollary 2. We have thus shown that x^* is optimal for problem (4) if and only if x^* provides a solution to relations (1), (2) and (3). To show that x_i^*, $i = r+1, \ldots, n$ are unique note that each optimal solution x^* to (4) must satisfy the extremality conditions with each dual optimal solution p^*. Thus for a given p^* and each x^*

$$f_i(x_i^*) = p_i^* \text{ for each } i = r+1, \ldots, n;$$

the strict monotonicity of $f_i(\cdot)$ then implies the uniqueness of the x_i^*, $i = r+1, \ldots, n$. This concludes the proof.

To conclude this section and for completeness of our exposition we set down a detailed formulation of the variational principle which can be derived from the dual problem to (4). Justification for its form can be found in the results of this and the previous section but we shall not go into the details here. It may be noted, however, that it is again a separable convex programming problem with linear constraints (with a form very similar to problem (4)). In fact, we could have based our developments on it rather than on (primal) problem (4) as we did. Note also that the algorithm which solves (4) to be given in the next section could be used equally well on the dual problem.

Remark: (Dual Variational Principle) The optimal solution p^* to the problem

$$\text{minimize } \sum_{1}^{n} h_i(p_i)$$

$$\text{s.t.} \quad p \in N(G) \cap D$$

where $D = \chi_{1}^{n} D_i$ and the functions $h_i{:}D_i$ have the form

$$h_i(p_i) = p_i \cdot F_i \text{ for each } p_i \in D_i = E_1,\ i = 1, \ldots, q$$

$$= 0 \text{ for each } p_i \in D_i = \{P_i\},\ i = q+1, \ldots, r$$

$$= \int_0^{p_i} f_i^{-1}(t)\,dt \text{ for each } p_i \in D_i = E_1,\ i = r+1, \ldots, n$$

produces a solution to relations (1), (2) and (3) in that

$$(P_{q+1}, \ldots, P_r,\ p^*_{r+1}, p^*_{r+2}, \ldots, p^*_n)$$

satisfies (2) and there exist x_i, $i = q+1, \ldots, r$ such that

$(F_1, \ldots, F_q,\ x_{q+1}, \ldots, x_r,\ f_{r+1}^{-1}(p^*_{r+1}), \ldots, f_n^{-1}(p^*_n))$ satisfies (1).

6. THE ALGORITHM

The following algorithm finds an optimal solution to problem (4) or the equivalent problem

$$\text{minimize} \sum_{q+1}^{n} g_i(x_i)$$

$$\text{s.t.} \qquad B'x = b$$

where B' is the node arc incidence matrix of the full network minus links 1, ..., q and b is an m-vector with q nonzero entries consisting of the F_j, j = 1, ..., q (one of the F_j can be replaced by the net supply rate from the sources at fixed pressure head if the input node associated with F_j is omitted in defining B'). In the statement of the algorithm we shall view $g(x) = \sum_{q+1}^{n} g_i(x_i)$ as a function of $x_{q+1}, \ldots, x_n$.

Iteration 1:

Step (0): Choose suitable numbers $\varepsilon > 0$, $\alpha^k > 0$, $k = 1, 2, \ldots$

Step (1): Find a vector x^1 that satisfies

$$B'x^1 = b.$$

Set $k = 1$.

Iteration k:

Step (2): Calculate

$$d^k = [I - B'^T (B'B'^T)^{-1} B'] \nabla g(x^k)$$

Step (3): Calculate x^{k+1} such that

$$g(x^{k+1}) = \min_{0 \leq \tau \leq \alpha^k} g(x^k - \tau d^k)$$

Step (4): If $(G'\nabla g(x^{k+1}))_i < \varepsilon$ for each $i = 1, \ldots, (n-q)-m$, stop; otherwise, set $k = k+1$ and go to Step 2.

With the formal statement of the algorithm completed we now discuss the rationale behind it, and its convergence properties.

First note that it is a simple adaptation of the standard gradient descent method for unconstrained minimization problems.

For a differentiable convex function $g(\cdot)$ with domain E_n, given a starting point x^1, the gradient descent algorithm finds an optimal point for the unconstrained problem

$$\min g(x),\ x \in E_n$$

by, at each iteration k, computing $\nabla g(x^k)$ and (assuming $\nabla g(x^k) \neq 0$) calculating x^{k+1} such that

$$g(x^{k+1}) = \min_{\tau \in [0,\alpha^k]} g(x^k - \tau \nabla g(x^k))$$

where $\alpha^k > 0$ is fixed in advance. The algorithm stops when some appropriate stopping condition is satisfied.

Our algorithm operates in a very similar fashion. We start at a point x^1 which satisfies the linear constraints $B'x = b$. Next we compute $\nabla g(x^1)$ and project it onto the null space of the matrix B'; then we proceed, as in the unconstrained case, using the projected gradient instead of the gradient. Use of the projected gradient (which we term d^k at iteration k) ensures that the line segment

$$\{y \in E_{n-q}: y = x^k - \tau d^k;\ 0 \leq \tau \leq \alpha^k\}$$

is feasible for the constrained problem.

We compute d^k as follows. From linear algebra we know each (n-q)-vector x is expressible as $x^1 + x^2$ where $x^1 \in N(B')$ and $x^2 \in [N(B')]^{\perp}$ and that x^1 and x^2 are unique. (The symbol $\perp$ denotes orthogonal complement.) We also know that each vector in $[N(B')]^{\perp}$ is expressible as the product $B'^T z$ for some vector $z \in E_m$. Hence

$$x^1 = x - B'^T z \text{ for some } z \in E_m.$$

Now the definition of $N(B')$ implies

$$0 = B'x^1 = B'x - B'B'^T z$$

and since $B'B'^T$ is nonsingular when B' has full row rank, we have

$$z = (B'B'^T)^{-1} B'x.$$

Thus

$$x^1 = x - B'^T(B'B'^T)^{-1}B'x$$
$$= [I - B'^T(B'B'^T)^{-1}B']x,$$

or

$$d^k = [I - B'^T(B'B'^T)^{-1}B']\nabla g(x^k).$$

It should be noted that since Theorem 5 reduces computation of solutions to (1), (2) and (3) to a convex programming problem with linear constraints* we could use any one of several algorithms which solve such problems (see Chapter 8 of [18]). Use of some of these algorithms (which for the most part are designed for more general problems with nonnegativity restrictions) would in many cases involve introducing extra variables as well as other complications. There seems to be no justification for taking such an approach here.

It is worth bearing in mind that any algorithm which involves the direct solution of (Kuhn-Tucker like) conditions, necessary and sufficient for optimality, may fail to take advantage of the variational principle we have developed. This is because such conditions are equivalent to the extremality conditions for problem (4) and its geometric dual which we have seen are equivalent to relations (1), (2) and (3). It was consideration of this point that led us to choose a descent type algorithm.

The steps of the algorithm we have given are simple to understand and program and convergence (in the sense defined in [18]) can be shown to occur under certain reasonable conditions. A set of such conditions is given in [18] where the steps of a convergence proof for the case where algorithm is applied to a more general problem than the one we consider are also outlined. The actual details of the proof are simple once the outline is followed; the key to the proof is that the projected gradient is non-zero unless the current point is a solution. Thus, the algorithm never stops except at a solution. The details of the assumptions are somewhat tedious to explain so we shall do no more in this paper than to say that in practice they are usually met.

**It can also be shown that the unconstrained minimizing point of a convex function provides a solution to (1), (2) and (3). In later work we shall make use of that fact.*

7. IMPLEMENTATION OF THE ALGORITHM, ITS EFFICIENCY AND AN EXAMPLE

The first step is to find a solution x^1 to

$$B'x = b$$

This may be done in at least two different ways. One way is to solve the linear equation $\begin{bmatrix} B' \\ G' \end{bmatrix} x = \begin{pmatrix} b \\ 0 \end{pmatrix}$.

The properties of B' and G' described in section 2 simply state that $\begin{bmatrix} B' \\ G' \end{bmatrix}$ is nonsingular so that a solution exists. Note that the solution of this equation is the solution to (1), (2) and (3) for the case where $f_i(x_i) = x_i$ for each $x_i \in C_i$, $i = r+1, \ldots, n$, and $x_i = p_i$ for each $i = q+1, \ldots, r$.

Another way is to form a tree of the network (many algorithms exist for doing this, see [2]) and to assign flows to the tree links which satisfy the conservation of flow law while assigning zero flow to the non-tree links. The latter is accomplished by considering first nodes j at which only one tree link is incident; these links are then assigned flows of zero or $\pm b_j$ as appropriate. Next consider nodes at which only one tree link is incident with unknown flow and assign flows which satisfy the conservation laws at these nodes. Continue in this fashion till all tree links have been assigned flows.

There are several points worth noting concerning the implementation of the second step of the algorithm; namely, that of computing

$$d^k = [I - B'^T(B'B'^T)^{-1}B']\nabla g(x^k)$$

To begin with, the matrix

$$I - B'^T(B'B'^T)^{-1}B'$$

need only be computed once; also the $m \times m$ matrix $B'B'^T$ can be written down by inspection of the network. The diagonal entry $(B'B'^T)_{ii}$ is simply the number of links incident at node i. The entry $(B'B'^T)_{ij}$ $i \neq j$ is the negative of the number of links which directly connect node i and node j. Both these facts are easily verified by considering the form of the matrix B'. In addition, the vector $\nabla g(x^k)$ may be written down directly as

$$\nabla g(x^k) = \begin{bmatrix} P_{q+1} \\ \vdots \\ P_r \\ f_{r+1}(x^k_{r+1}) \\ \vdots \\ f_n(x^k_n) \end{bmatrix}$$

This is a consequence of the separability and differentiability of $g(\cdot)$ as well as the forms of $g_i(\cdot)$ $i = q+1, \ldots, n$.

A wide variety of methods exist for effecting Step 3 (see Reference [18]). In solving the example problem below we used the popular Fibonnaci method, stopping the search when the interval within which the minimizing value of τ was known to lie (the error tolerance) was sufficiently small. In the neighborhood of the optimal solution this tolerance must be made very small indeed or the algorithm may "fluctuate" about the solution.

The stopping criterion of the algorithm itself (Step 4) is designed to reflect the fact that we are interested in an optimal solution vector for problem (4) and not in the optimal objective value. Since the algorithm produces a sequence of points $\{x^k\}$ a subsequence of which converges to an optimal vector of problem (4), it follows that $\{G'\nabla g(x^k)\}$ for this subsequence converges to zero. Thus we are guaranteed that the stopping criterion will be eventually satisfied.

In the literature the Newton-Raphson method of solving a system of nonlinear equations is widely advocated as being the most efficient technique for finding a solution to relations (1), (2) and (3). Thus it is of interest to give some consideration to the efficiency of the proposed algorithm as compared to that of the NR method. Suppose we wish to find x such that $F(x) = 0$. In performing an iteration of the NR method (other than the first), the current Jacobian matrix of $F(\cdot)$ is calculated, a correction to the current solution is computed (this step involves inverting the Jacobian) and the proximity of the new solution to the required solution is evaluated. That is, three basic steps are performed - the same number as in a general iteration of the proposed algorithm. A detailed comparison of a general iteration of the NR procedure with one of the proposed algorithms shows that the amount of computation is roughly the same for corresponding steps with one important exception. The matrix inversion operation of the NR method corresponds to the line search of the proposed algorithm and in general the latter can be performed much more efficiently than the former [18]. It is well known that matrix inversion accounts for a large part of the computation cost per iteration in the NR method. Indeed, most of the recent work on this method has concentrated on exploiting special structural features of the Jacobian matrix (sparseness and symmetry) in order to expedite its inversion: the effort required however, is still considerable [4], [8]. This indicates the advantage of removing matrix inversion from the solution procedure at each iteration.

On a cost per iteration basis then the proposed algorithm seems to compare favorably with the NR method but of course consideration must also be given to the relative numbers of iterations required for convergence. Although the results of limited testing show promise, further testing and research is needed on the merits of the proposed algorithm in this respect. Choice of starting point, choice of step size α^k in performing the line search, whether the direction vector d^k is normalized or not, are all factors that influence this aspect of the algorithms performance.

To illustrate the use of the algorithm we solve the following example problem. The network has the configuration as illustrated in Figure 1. The fixed pressure heads at nodes 1 and 2 are -5000 and -8000 respectively (relative to some datum). The fixed inflow at node 4 is 100 and the fixed outflow at node 5 is 200. Valid choices for the matrices B' and G' are

$$B' = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$G' = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 \end{bmatrix}$$

where node 5 is omitted in constructing B' and the tree used in constructing G' consists of links 4, 8, 9, 10 and 11. The simple formula given after equation (3) is assumed to hold for each real link (i.e., each pipe) of the network.

The value of a_i was chosen as 2.0 for all real links and the values of r_i were chosen as $r_5 = 2.0$, $r_6 = 3.0$, $r_7 = 1.0$, $r_8 = 5.0$, $r_9 = 4.0$, $r_{10} = 2.0$, $r_{11} = 3.0$. For real link i the function $g_i(\cdot)$ has the form

$$g_i(x_i) = \begin{cases} r_i x_i^3/3 & x_i \geq 0 \\ -r_i x_i^3/3 & x_i < 0. \end{cases}$$

The matrix $B'B'^T$ becomes

$$\begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & 0 & -1 & -1 \\ -1 & 0 & 3 & -1 & 0 \\ 0 & -1 & -1 & 4 & -1 \\ 0 & -1 & 0 & -1 & 3 \end{bmatrix}$$

and the vector b = (100, 0, 0, 0, 100).

The initial flow vector x^1 was chosen as (50, 50, 150, 0, 0, 50, 0, 50) and the quantities ε of the stopping criterion (4), and α^k of the minimization step (3) were set at 1.0 and 200.0. The algorithm was programmed in FORTRAN and run on a CDC 6400 computer; the solution vector was obtained as

$$x_3^* = 2.4,\ x_4^* = 79.6,\ x_5^* = 3.0,\ x_6^* = 60.0,$$

$$x_7^* = 88.2,\ x_8^* = 23.3,\ x_9^* = -8.5,\ x_{10}^* = -37.0,\ x_{11}^* = 51.8.$$

In conclusion it should be noted that there are other advantages to taking the variational approach to solution advocated in this paper. An approximate solution to the problem may be determined by performing sufficient iterations of the proposed algorithm to determine an approximate minimum of the function $g(\cdot)$ (see Prager [13]). It is conceivable that simultaneous analysis and design of networks can be achieved using the variational approach. This is so since the design problem is also usually one of minimization - in this case cost. Finally, the viewpoint developed in this paper is mathematically interesting of itself and it also affords a means of applying modern mathematical programming theory to the problem. This provides useful information about the solution to the problem (e.g., existence and uniqueness conditions) as well as new and promising methods for obtaining the solution.

ACKNOWLEDGMENT

The author would like to acknowledge the helpful criticisms of one of the referees.

REFERENCES

1. Barlow, J. F. and E. Markland, "Computer Analysis of Pipe Networks," *Proc. Inst. Civ. Engrs.*, 43, 1969, pp. 249-259.

2. Berge, C. and A. Ghouila-Houri, *Programming Games and Transportation Networks*, John Wiley & Sons, New York, 1965.

3. Duffin, R. J., "Nonlinear Networks," *Bull. Amer. Math. Soc.*, 1947, pp. 963-971.

4. Epp, R. and A. G. Fowler, "Efficient Code for Steady State Flows in Networks," *Journal of the Hydraulics Division, ASCE*, 96, 1970, pp. 43-56.

5. Hachtel, G. C., R. K. Brayton and F. G. Gustavson, "The Sparse Tableau Approach to Network Analysis and Design," *IEEE Trans. Circuit Theory*, 18, 1971, pp. 101-113.

6. Hall, M. A. and E. L. Peterson, "Traffic Equilibria Analyzed via Geometric Programming," paper presented at the *Intern. Symp. on Traffic Equilibrium Methods*, University of Montreal, 1974.

7. Hall, M. A., "Civil Engineering Applications of (Generalized) Geometric Programming," *Journal of Engineering Optimization*, to appear in Vol. 2.

8. Lemieux, P. F., "Efficient Algorithm for Distribution Networks," *Journal of the Hydraulics Division, ASCE*, 98, 1972, pp. 1911-1919.

9. Martin, D. W. and G. Peters, "The Application of Newtons' Method of Analysis by Digital Computer," *Journal of the Institute of Water Engineers*, 17, 1963, pp. 115-129.

10. Minty, G. J., "Monotone Networks," *Proc. of Royal Society*, London, Series A257, 1960, pp. 194-212.

11. Peterson, E. L., "Symmetric Duality for Generalized Unconstrained Geometric Programming," *SIAM Journal Applied Math.*, 19, 1970, pp. 487-526.

12. Peterson, E. L., "The Mathematical Foundations of Geometric Programming," appendix in *Optimization and Design*, M. Avriel, M. Rijckaert and D. Wilde, eds., Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

13. Prager, W., "Problems of Network Flow," *Zeits F. Angew Math. Physik.*, 16, 1965, pp. 185-190.

14. Rockafeller, R. T., "Convex Programs and Systems of Elementary Monotonic Relations," *Journal of Math. Analysis Applications*, 19, 1967, pp. 543-564.

15. Rockafeller, R. T., *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.

16. Shamir, U. and C. C. D. Howard, "Water Distribution System Analysis," *Journal of the Hydraulics Division, ASCE*, 94, 1968, pp. 119-234.

17. Wood, D. J. and C. O. Charles, "Hydraulic Network Analysis Using Linear Theory," *Journal of the Hydraulics Division, ASCE*, 98, 1972, pp. 1157-1170.

18. Zangwill, W. I., *Nonlinear Programming - A Unified Approach*, Prentice-Hall, Englewood Cliffs, New Jersey, 1970.

The author is currently affiliated with Bell Laboratories, Piscataway, New Jersey.

Paper received July 17, 1974.