A RESOURCE LOCATION PROBLEM ON GRAPHS*

by

B. D. McKay and I. G. Tollis

Computer Science Department

Vanderbilt University

Nashville, Tennessee 37235

Abstract

Consider a graph G with distinguished vertices s and t. Each edge has a positive length and may be directed or undirected. We wish to locate a minimum number of "gas-stations" on the vertices or edges of G such that a car with a range of R may start at s with no gas and drive to t along any simple path. We investigate the theoretical complexity of the problem, and describe an $O(|E| \min\{R, |V|\})$ algorithm for the case when G is a series-parallel graph.

0. Introduction

Consider a network of cities and highways. Each highway joins two cities and doesn't intersect any other highways except at the ends (but over-passes are allowed). Some of the highways may be declared "one-way" in some direction. We wish to drive from one fixed city s (the *source*) along the highways to a second fixed city t (the *sink*), and feel free to use any simple path. However, our car is almost out of gas and, even with a full tank, cannot go more than R miles without running out. So, in order to make our journey possible, we need gas-stations. Clearly, we need one at s but (since we don't mind arriving with an empty tank) we don't need one at t. The problem is this: What is the smallest number of gas-stations, positioned anywhere on the cities or along the highways, so that we can successfully drive from s to t along any simple path?

An example with 5 cities and 6 highways is shown in Figure 0.1. The numbers on the highways represent their lengths (always positive), and the value of the range R is 200.

---

*This paper contains original research and will not be published elsewhere.

CONGRESSUS NUMERANTIUM, VOL. 40 (1983), pp. 223-234

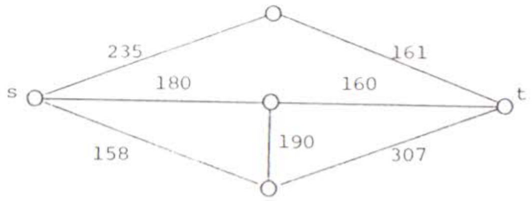

Figure 0.1

By considering each of the 5 simple paths from s to t we see that at least 5 gas-stations are needed. One possible configuration of these is indicated by the crosses in Figure 0.2.

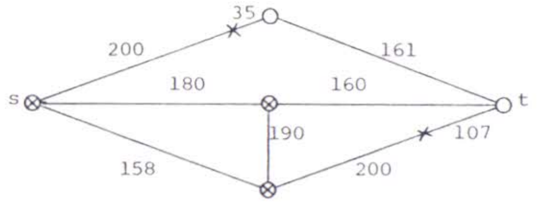

Figure 0.2

Before proceeding with our analysis, we need some more terminology. Given a network G (including edge-lengths), a range R, and a configuration of gas-stations, a simple path P from s to t is *reliable* if we can drive from s to t along P without running out of gas. More formally, P is reliable if there is a gas-station at s, one on P at most R miles from t, and if any two consecutive gas-stations on P are at most R miles apart. All distances here are measured along P. The configuration itself is *reliable* if every simple path from s to t is reliable. An *optimum reliable configuration* is one for which the number of gas-stations is minimum.

At first glance, there appears to be difficulty in devising even a crude exhaustive search for an optimum reliable configuration, since there is an infinity of legal positions for the gas-stations. This difficulty will

be solved in the next section, at least when the edge-lengths and R are integers. We will then examine the theoretical complexity of the problem. In section 2 we will describe a polynomial time algorithm for series-parallel graphs.

As we have begun to do already, highways will sometimes be referred to as *edges*. Also, note that an undirected edge is not equivalent to two oppositely directed edges, since a gas-station on an undirected edge can service traffic moving in either direction.

1. Complexity results

We begin by showing that a solution by exhaustive search is possible when all the edge-lengths and R are positive integers. An *integer point* is a point (i. e., a position) on an edge whose distance from either end of the edge is an integer. A *half-integer point* is one whose distance from either end of the edge is half an odd integer. Note that cities are integer points if the edges have integer lengths. The proof of the following theorem is partly due to Hang Tong Lau.

Theorem 1.1 *Suppose that all edge-lengths and R are positive integers. Then there is an optimum reliable configuration for which all gas-stations lie on integer points or half-integer points.*

Proof. Let C be an optimum reliable configuration. We can convert C into another optimum reliable configuration C' satisfying the required property thus: take each gas-station of C which does not lie on an integer point or half-integer point and move it to the closest integer point. We only need to show that C' is reliable.

Take any simple path P from s to t and consider two consecutive gas-stations, g_1 and g_2, on P in C. Let $z_1+\varepsilon_1$ and $z_2+\varepsilon_2$ be the distances along P from s to g_1 and g_2, respectively, where z_1 and z_2 are integers and $0\le\varepsilon_1, \varepsilon_2<1$. Since C is reliable,

$$z_2+\varepsilon_2 - (z_1+\varepsilon_1) \le R.$$

In C', these same gas-stations have distances $z_1+\varepsilon'_1$ and $z_2+\varepsilon'_2$ along P from s, respectively, where $0 \le \varepsilon'_1, \varepsilon'_2 \le 1$. We need to show that $z_2+\varepsilon'_2 - (z +\varepsilon'_1)\le R$. There are nine cases to consider, depending on whether $0\le\varepsilon_i < \frac{1}{2}$, $\varepsilon_i = \frac{1}{2}$ or $\frac{1}{2}< \varepsilon_i < 1$ for $i=1, 2$. We will do just two of these and leave the rest to the reader.

Case 1: $0 \le \varepsilon_1 < \frac{1}{2}$ and $0 \le \varepsilon_2 < \frac{1}{2}$.

By (1), $z_2 - z_1 \le R + \varepsilon_1 - \varepsilon_2 < R + \frac{1}{2}$. But z_1, z_2 and R are integers, so $z_2 - z_1 \le R$. In C' we have $\varepsilon_1' = \varepsilon_2' = 0$, so the distance along P from g_1 to g_2 is still at most R.

Case 2: $0 < \varepsilon_1 < \frac{1}{2}$ and $\frac{1}{2} < \varepsilon_2 < 1$.

By (1), $z_2 - z_1 \le R + \varepsilon_1 - \varepsilon_2 < R$. But z_1, z_2 and R are integers, so $z_2 - z_1 \le R-1$. Thus the distance along P from g_1 to g_2 in C' is again at most R.

An identical argument successfully handles the portion of P from the last gas-station to t. □

The example drawn in Figure 1.1 shows that it is not always sufficient to consider only integer points. However, there is an important case where

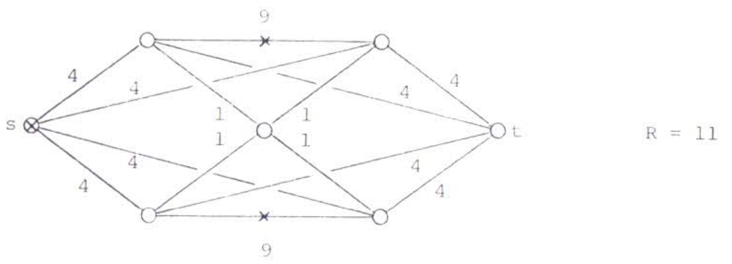

Figure 1.1

we can. Call a network *essentially one-way* if, for each edge e, all the simple paths from s to t using e traverse e in the same direction. This is trivially true if only one-way highways exist, but it is also true in other important cases, for example, if the network is series-parallel (see Section 2).

Theorem 1.2 *Suppose that an essentially one-way network has only integer edge-lengths and R is an integer. Then there is an optimum reliable configuration for which all gas-stations lie on integer points.*

Proof. Let C be an optimum reliable configuration. We can convert C into an optimum reliable configuration C' using only integer points thus: take each gas-station g of C which does not lie on an integer point and move it to the closest integer point which precedes g on some simple path from s to t. C' is well-defined because C is essentially one way. Note that there is at least one simple path through s, g, and t, or else g can be removed altogether, contradicting the optimality of C. It is now a simple exercise to see that C' is reliable. □

We turn now to theoretic compexity questions and consider two decision problems.

Problem 1

Instance: A configuration C, consisting of a network with positive integral edge-lengths, specified cities s and t, a set of gas-stations on integer or half-integer points, and a positive integral range R.

Question: Is C unreliable?

Problem 2

Instance: A network with positive integral edge-lengths, specified cities s and t, a positive integral range R, and an integer $K \ge 0$.

Question: Is there a reliable configuration using K or fewer gas-stations?

Problem 1 is in NP obviously. To see that it is NP-complete, take the case where all edge-lengths are 1, R=n-2 (where n is the total number of cities) and the only gas-station is at s. The question is then "is there a hamiltonian path from s to t?", which is well-known to be NP-complete. This is true whether the edges are restricted to be all undirected, all directed, or not so restricted.

On the other hand, if R is taken to be a fixed integer and all edges are undirected we can find a polynomial-time algorithm. Take each gas-station g in turn and generate all the simple paths of length R which start at g. There are at most Δ^R of these, where Δ is the maximum degree. Let P be one of these paths, with end-points g and h (neither of which must be a city). If P contains a gas-station other than g we discard it; otherwise we use a network-flow algorithm to determine whether there are vertex-disjoint simple paths P' and P", each vertex-disjoint from P, from s to g and from h to t. If such paths exist, the concatenation P'PP" is unreliable. Conversely, any unreliable path will be found by this means. If directed edges are allowed, Problem 1 is NP-complete even when all edges have unit length, R=1, and all gas-stations are in cities. To see that, recall from [1] that the problem of determining whether there are two vertex-disjoint paths between two pairs of vertices in a directed graph is NP-complete. The last problem is clearly equivalent to that of finding whether there is a simple path from s to t which uses a specified vertex v. This problem in turn can be reduced to an instance of Problem 1 restricted as we agreed.

If the network is acyclic, planar or chordal, Problem 1 again becomes polynomial for fixed R, even for directed edges. See [1].

Now consider Problem 2. Our results on Problem 1 raise doubts as to whether Problem 2 is even in NP. Another potential difficulty is that the number of gas-stations in an optimum reliable configuration may be exponential in the size of the input (to Problem 2). This can be circumvented by noting that all we really need to know about the gas-stations on each edge is their number and the positions of the first and last of them. Those in between can be assumed to be evenly spaced without losing reliability. With this compact representation in mind, we see that Problem 2 becomes polynomial if there are no directed edges and both R and K are fixed. This follows from Theorem 1.1 and the fact that Problem 1 has a polynomial time solution if there are no directed edges and R is fixed.

If either R or K or both are allowed to vary, most versions of Problem 2 become NP-hard. For example, the case K=1 is co-NP-complete even if all edges have length 1. Another interesting subcase comes from disallowing directed edges, giving all edges length 1, and having R=3. This is NP-complete as can be seen by reduction from the vertex-cover problem: "Given a graph G and integer $K \geq 0$, is there a set of K or fewer vertices which cover the edges of G?". Take an arbitrary instance (G, K) of the vertex-cover problem, and from it construct an instance (G', K') of (the restricted version of) Problem 2: $V(G') = V(G) \cup \{s, t\}$, $E(G') = E(G) \cup \{sv | v \in V(G)\} \cup \{tv | v \in V(G)\}$, $K' = K+1$. For this special instance of Problem 2, it is easy to see that there is at least one optimum reliable configuration for which all gas-stations lie on cities. The NP-completeness follows immediately.

2. Series-Parallel graphs

By the notation $G = (V, E, s, t)$ we mean that G is a network with cities V, edges E, source $s \in V$ and sink t. As before, edges have positive lengths and may be either directed or undirected.

Given two networks $G = (V_1, E, s_1, t_1)$ and $G_2 = (V_2, E_2, s_2, t_2)$, we can define two composite networks as follows. Assume that $V_1 \cap V_2 = \emptyset$. The *series connection* of G_1 and G_2 is the network $Ser(G_1, G_2)$ formed from $G_1 \cup G_2$ by identifying t_1 and s_2, then taking s_1 as the source and t_2 as

the sink. The *parallel connection* of G_1 and G_2 is the network $Par(G_1, G_2)$ formed from $G_1 \cup G_2$ by identifying s_1 and s_2 (with the resulting single city becoming the new source) and identifying t_1 and t_2 (making the new sink). These are shown in Figure 2.1.

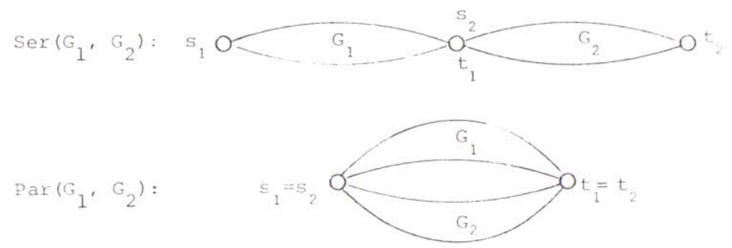

Figure 2.1

The class of *series-parallel networks* is defined recursively as follows:

(i) A network consisting of a single undirected edge joining s to t is a series-parallel network.

(ii) If $G_1 = (V_1, E_1, s_1, t_1)$ and $G_2 = (V_2, E_2, s_2, t_2)$ are series-parallel networks, then so are $Ser(G_1, G_2)$ and $Par(G_1, G_2)$.

Before we can present our algorithm for series-parallel networks, we need to generalize the concept of reliability. Consider a configuration C of gas-stations on a network and let P be simple path from s to t which contains m gas-stations $g_1, g_2, \ldots, g_m$. Define $g_0 = s$ and $g_{m+1} = t$. Then $\ell_i(P)$ is the distance along P from g_i to g_{i+1} $(0 \leq i \leq m)$. For notational convenience define $\ell_\infty(P) = \ell_m(P)$. We say that P is *r-reliable* if $\ell_0(P) \leq r$ and $\ell_i(P) \leq R$ $(i \geq 1)$. The configuration C itself is *r-reliable* if every simple path from s to t is r-reliable. Clearly reliability in our previous sense is the same as 0-reliability.

Let $G = (V, E, s, t)$ be a network and let $0 \leq r < R$. For an r-reliable configuration C on G define $slop(C) = \min (\{R - \ell_\infty(P) | P \in P_1\} \cup \{r - \ell_\infty(P) | P \in P_2$ where P_1 is the set of simple paths from s to t which contain at least one gas-station, and P_2 is the set of simple paths from s to t which don't. Essentially, slop(C) is a measure of how much gas we can be sure of having left on arriving at t. Now define

OPT(G, R, r) = minimum number of gas-stations in an r-reliable configuration on G, and

EXC(G, R, r) = maximum slop(C) for any r-reliable configuration on G with exactly OPT(G, R, r) gas-stations

A few elementary properties of OPT and EXC are listed in the following lemma.

Lemma 2.1 *Let* $0 \le r_1 < r_2 < R$. *Then:*

(i) $0 \le EXC(G, R, r_1) < R$.

(ii) $OPT(G, R, r_1) - 1 \le OPT(G, R, r_2) \le OPT(G, R, r_1)$.

(iii) *If* $OPT(G, R, r_1) = OPT(G, R, r_2)$, *then* $EXC(G, R, r_1) \le EXC(G, R, r_2)$.

(iv) *If* $OPT(G, R, r_1) > OPT(G, R, r_2)$, *then* $EXC(G, R, r_1) \ge EXC(G, R, r_2)$.

Proof. The only parts which are not immediate consequences of the definitions are (iv) and the left inequality of (ii). These follow easily on noticing that an r_2-reliable configuration can be made r_1-reliable by inserting an extra gas-station at s, if there isn't one there already. □

We begin our determination of OPT and EXC with the network consisting of a single edge.

Theorem 2.2 *Let G be a network consisting of a single undirected edge of length* ℓ *from* s *to* t. *Let* $0 \le r < R$. *Then*

(i) $OPT(G, R, r) = \left\lceil \frac{\ell - r}{R} \right\rceil$ and

(ii) $EXC(G, R, r) = r - \ell + R\left\lceil \frac{\ell - r}{R} \right\rceil$.

Proof. Both claims follow easily from the definitions. The optimum configuration has gas-stations at distances r, r+R, r+2R, . . . from s.

We will next investigate the effect of the operations Ser and Par on OPT and EXC.

Theorem 2.3 *Let* $G_1 = (V_1, E_1, s_1, t_1)$ *and* $G_2 = (V_2, E_2, s_2, t_2)$ *be networks with* $V_1 \cap V_2 = \emptyset$. *Let* $H = Ser(G_1, G_2)$ *and* $0 \le r < R$. *Then*

(i) $OPT(H, R, r) = OPT(G_1, R, r) + OPT(G_2, R, EXC(G_1, R, r))$, *and*

(ii) $EXC(H, R, r) = EXC(G_2, R, EXC(G_1, R, r))$.

Proof. An r-reliable configuration on H with the required parameters can be constructed from one on G_1 realizing $OPT(G_1, R, r)$ and $EXC(G_1, R, r)$, then one on G_2 realizing $OPT(G_2, R, r')$ and $EXC(G_2, R, r')$, where $r' = EXC(G_1, R, r)$. We only need to demonstrate that it is impossible to do better.

Let C be an r-reliable configuration on H realizing OPT(H, R, r) and EXC(H, R, r), and define C_1 and C_2 to be the restrictions of C to G_1 and G_2, respectively. Any gas-stations on $s_2 = t_1$ belongs in C_2 but not in C_1. Define e = slop(C_1).

Since C_1 is an r-reliable configuration on G_1, $|C_1| \ge OPT(G_1, R, r)$. Also, $|C_2| = OPT(G_2, R, e)$ by the minimality of $|C|$. If $|C_1| = OPT(G_1, R, r)$, then $e \le r'$ so $|C_2| \ge OPT(G_2, R, r')$. If $|C_1| > OPT(G_1, R, r)$ then $|C_2| = OPT(G_2, R, r') - 1$, by Lemma 2.1 (ii). In either case $|C| \ge OPT(G_1, R, r) + OPT(G_2, R, r')$, which establishes (i).

To prove (ii), note that slop(C_2) = $EXC(G_2, R, e)$ by the maximality of slop(C). If slop(C_2) > $EXC(G_2, R, r')$ then, by Lemma 2.1, either $OPT(G_2, R, e) = OPT(G_2, R, r')$ and $e > r'$, or $OPT(G_2, R, e) = OPT(G_2, R, r') + 1$. Each case is clearly impossible by part (i). □

Theorem 2.4 *Let* $G_1 = (V_1, E_1, s_1, t_1)$ *and* $G_2 = (V_2, E_2, s_2, t_2)$ *be networks with* $V_1 \cap V_2 = \emptyset$.

Let $H = Par(G_1, G_2)$, $0 \le r < R$ *and define*

$N_1 = OPT(G_1, R, r) + OPT(G_2, R, r)$,

$N_2 = OPT(G_1, R, 0) + OPT(G_2, R, 0) - 1$,

$M_1 = \min\{EXC(G_1, R, r), EXC(G_2, R, r)\}$, *and*

$M_2 = \min\{EXC(G_1, R, 0), EXC(G_2, R, 0)\}$.

Then

(i) $OPT(H, R, r) = \min\{N_1, N_2\}$, *and*

(ii) $$EXC(H, R, r) = \begin{cases} M_1, & N_1 < N_2, \\ \max\{M_1, M_2\}, & N_1 = N_2, \\ M_2, & N_1 > N_2. \end{cases}$$

Proof. Let C be an r-reliable configuration for H realizing OPT(H, R, r) and EXC(H, R, r). Define C_1 and C_2 to be the restrictions of C to G_1 and G_2, respectively. Any gas-station in C at $s_1 = s_2$ appears both in C_1 and C_2. Obviously, C_1 and C_2 are r-reliable and slop(C) = min{slop(C_1), slop(C_2)}. We now consider four separate cases.

(a) Suppose $OPT(G_1, R, r) = OPT(G_1, R, 0)$ and $OPT(G_2, R, r) = OPT(G_2, R, 0)$. Then $N_2 < N_1$. Obviously, C must have a gas-station at $s_1 = s_2$, and so $|C| = N_2$ and slop(C) = M_2.

(b) Suppose $OPT(G_1, R_1, r) < OPT(G_1, R, 0)$ and $OPT(G_2, R, r) < OPT(G_2, R, 0)$. Then $N_1 < N_2$. Obviously, C must have had a gas-station at $s_1 = s_2$, and so $|C| = N_1$ and slop(C) = M_1.

(c) Suppose $OPT(G_1, R, r) = OPT(G_1, R, 0)$ and $OPT(G_2, R, r) < OPT(G_2, R, 0)$. By Lemma 2.1 (ii), $OPT(G_2, R, r) = OPT(G_2, R, 0)-1$, so $N_1 = N_2$. If C has a gas-station at $s_1 = s_2$, then $|C| = N_2$ and $slop(C) = M_2$. This is clearly realizable. If C does not have a gas-station at $s_1 = s_2$, then $|C| \geq N_1$ and, if $|C| = N_1$, then $slop(C) \leq M_1$. We must only show that if $M_1 > M_2$ we can realize N_1 and M_2. By Lemma 2.1, $EXC(G_1, R, r) \geq EXC(G_1, R, 0)$ and $EXC(G_2, R, r) \leq EXC(G_2, R, 0)$ so $M_1 > M_2 \Rightarrow EXC(G_1, R, r) > EXC(G_1, R, 0)$. Thus, the r-reliable configuration on G_1 which realizes $OPT(G_1, R, r)$ and $EXC(G_1, R, r)$ does not have a gas-station at s_1. This, together with an optimal r-reliable configuration on G_2, realizes N_1 and M_1.

(d) The remaining case is equivalent to case (c). □

It is now a simple matter to devise an algorithm for determining $OPT(G, R, 0)$, where $G = (V, E, s, t)$ is a series-parallel network. From the definition of the class of series-parallel networks we see that G can be represented by the root of a rooted tree which has one leaf for each edge. Internal nodes of the tree are classified as type S or P. A type S node represents the series-connection of its children. Formally, if the children of v represent $G_1, G_2, \ldots, G_m$, left to right, then v represents $Ser(G_1, Ser(G_2, \ldots, Ser(G_{m-1}, G_m)) \ldots)$. A type P vertex represents the parallel-connection of its children. If we stipulate that no two internal nodes of the same type are adjacent, the tree is unique up to the ordering of the children of type P vertices. An example is shown in Figure 2.2.

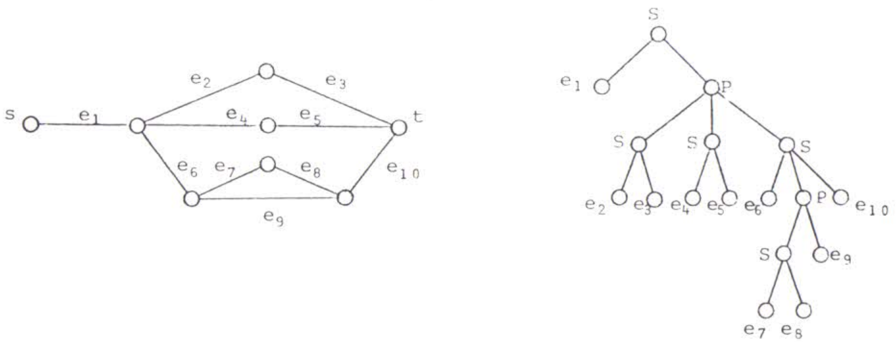

Figure 2.2

It is known ([4], [3]) that the tree representation of G can be found from G in $O(|E|)$ time. Once we have the tree representation, we can apply Theorems 2.2-2.4 recursively to find $OPT(G, R, 0)$.

Theorem 2.5 *Let $G = (V, E, s, t)$ be a series-parallel network. Then $OPT(G, R, 0)$ can be found in $O(|V||E|)$ time if all arithmetic operations count as one unit.*

Proof. The algorithm is as we have described: Find the tree representation in $O(|E|)$ time, then apply Theorems 2.2-2.4 recursively.

Let $G(v)$ be the subgraph of G represented by node v of the tree, and let $N(v)$ be the number of times $OPT(G(v), R, r)$ and $EXC(G(v), R, r)$ need to be computed for some r. The total time required is clearly $O(\sum_v N(v))$. If the ancestor w of v has type S, then $N(v) \leq N(w)$ by Theorem 2.3. If w has type P, then $N(v) \leq N(w) + 1$ by Theorem 2.4. Thus $N(v)$ is bounded by the number of ancestors of v (including v itself) which have type P, except that $N(root) = 1$ always. Therefore, $N(v) \leq |V|$ always, which implies the Theorem as stated. □

If R and all edge-lengths are integers, and $R = o(|V|)$, the time bound above can be reduced to $O(R|E|)$. This is done by storing all computed values $OPT(G(v), R, r)$ and $EXC(G(v), R, r)$ for each v and r, then not ever computing them again. Since series-parallel networks are essentially one-way [2] we can place all gas-stations on integer points, by Theorem 1.2. This implies that only integer values of r are ever encountered, so at most R different values are possible for each v.

A combination of the two methods produces an algorithm of time complexity $O(\min\{R, |V|\}|E|)$. It is not difficult to obtain the actual positions of gas-stations in an optimal configuration at the same time.

3. Problems

(1) What is the complexity of Problem 2 for acyclic digraphs?

(2) Is there a finite algorithm for solving the original problem for a general network when R and the edge-lengths are arbitrary positive real numbers (using exact arithmetic)?

Acknowledgement

We are indebted to Hang Tong Lau, who introduced us to the problem and contributed many useful ideas.

References

1 Y. Perl and Y. Shiloach, Finding two disjoint paths between two pairs of vertices in a graph, *J. Assoc. Comp. Mach.* 25(1978) 1-9.

2 J. Riordan and C. Shannon, The number of two-terminal series-parallel networks, *J. Math. Phys. Mass. Inst. Tech.* 21(1942) 83-93.

3 J. Valdes, Parsing flowcharts and series-parallel graphs, Tech. Report STAN-CS-78-682, Computer Science Dept., Stanford University, 1978.

4 J. Valdes, R. E. Tarjan and E. L. Lawler, The recognition of series-parallel digraphs, Proc. 11th Annual ACM Symp. on Theory of Computing, Atlanta 1979, 1-12.

EMBEDDING MAXIMAL PACKINGS OF TRIPLES †

Eric Mendelsohn* and Alexander Rosa**

1. *Introduction.*

A *Steiner triple system* of order v (STS(v)) is a pair (V,B) where V is a v-set, and B is a collection of 3-subsets of V (called *triples*) such that each 2-subset of V is contained in exactly one triple. It is well known [2] that an STS(v) exists if and only if $v \equiv 1$ or 3 (mod 6).

An STS(u) (U,A) is said to be *embedded* in an STS(v) (V,B) if $U \subset V$ and $A \subset B$ (written $(U,A) \leq (V,B)$). Doyen and Wilson proved in 1973 [1] that any STS(u) can be (properly) embedded in an STS(v) if and only if $v \geq 2u+1$. (Recently, a new simplified proof was given by Stern and Lenz [10].)

When $v \not\equiv 1,3$ (mod 6), one has the following analogue of STSs. A *maximal packing of triples* of order v (MPT(v)) is a pair (V,B) where V is a v-set, and B is a collection of 3-subsets of V (called triples) such that (i) each 2-subset of V is contained in *at most* one triple, (ii) no triple can be adjoined to B without violating (i), and (iii) if C is any collection of 3-subsets satisfying (i), (ii) then $|B| \geq |C|$.

If (V,B) is an MPT(v), let $C(V,B)$ be the graph induced by the "uncovered" pairs (i.e. pairs not occurring in any triple of B). It is well known [8,9] that $C(V,B)$ is as follows.

† This paper contains original research and will not be published elsewhere

* Research supported by NSERC Grant No. A7681

** Research supported by NSERC Grant No. A7268