SOME PROBLEMS ON DYNAMIC/PERIODIC GRAPHS*

James B. Orlin

Sloan W.P. No. 1399-83 January 1983

*We gratefully acknowledge partial support received from the National Science Foundation Grant ECS-8205022, entitled "Research Initiation: Dynamic/Periodic Optimization Models."

James B. Orlin, Massachusetts Institute of Technology, Sloan School of Management, Cambridge, Massachusetts 02139.

Abstract

A dynamic graph is a (locally finite) infinite graph $G=(V,E)$ in which the vertex set is $V = \{i^p : i=1,\dots,n \text{ and } p \in Z\}$, where Z is the set of integers, and the edge set has the following periodic property: (i^p, j^r) is an edge of E if an only if (i^{p+1}, j^{r+1}) is an edge of E. Dynamic graphs may model a wide range of periodic combinatorial optimization problems in workforce scheduling, vehicle routing, and production scheduling.

Here we provide polynomial time algorithms for several elementary problems including the following: determining the connected components, determining the strongly connected components, determining an eulerian path if one exists, and determining a 2-coloring if one exists. (Here, the polynomial is in the finite number of bits needed to describe a dynamic graph.) In each case the problem on the dynamic graphs reduces to a related (but distinct) problem on a finite graph.

0745219

1. Introduction

In this paper we consider dynamic/periodic combinatorial optimization problems. The problems are dynamic in the sense that the best schedule on any day depends on the schedules on the preceding and succeeding days. The problems are periodic in the sense that the demands and constraints for any week are the same as those of preceding and succeeding weeks.

A number of problems in transportation planning, communications, and operations management may be modeled as dynamic/periodic optimization problems. For example, Simpson (1968) considers a number of different models for airplane scheduling. As another example, Baker (1976) considers a number of models relating to (cyclical) workforce scheduling. There have also been a number of papers relating to processor scheduling in a periodic environment, e.g. Dhall and Liu (1978), Labetoulle (1974), Lawler and Martel (1981) and Liu and Layland (1973).

The above list of papers comprises just a small sample of papers relating to applications in dynamic/periodic scheduling. In this paper we focus on fundamental combinatorial structures relating directly to dynamic/periodic combinatorial optimization problems. In particular we will investigate and analyze problems on dynamic/periodic graphs. These graphs are infinite horizon graphs that are "time expansions" of finite graphs. They may be viewed as dynamic/periodic analogs of finite graphs.

Dynamic Graphs

Let $G = (V,E,T)$ be a directed graph with vertex set $V = \{1,\ldots,n\}$ and such that each edge $(u,v) \in E$ has an integral (possibly non-positive) transit time t_{uv}, which may be interpreted as the number of time periods that it takes to travel from u to v along the edge. We make the simplifying assumption that there is at most one edge from u to v so as to simplify the notation (it is easy to show that the results in this paper do extend to graphs in which multiple edges are permitted.)

A static graph G = (V,E,T) is said to induce a directed dynamic graph $G^\infty = (V^\infty, E^\infty)$ via time expansion as follows: Let Z denote the set of integers. Then

$$V^\infty = \{v^p : v \in V \text{ and } p \in Z\}$$

$$E^\infty = \{(u^p, v^{p+t_{uv}}) : (u, v) \in E , p \in Z\}.$$

The vertex v^p of G^∞ represents vertex v of G in period p, and edge $(u^p, v^{p+t_{uv}})$ represents "traveling" from u to v starting in period p and arriving t_{uv} periods later. A static graph is portrayed in Figure 1, and the induced dynamic graph is portrayed in Figure 2.

In this paper we analyse dynamic/periodic analogs of the classical "easy" problems of graph theory, viz., the problems of determining (1) the weakly connected components, (2) the strongly connected components, (3) eulerian paths, (4) odd length circuits, and (5) minimum average cost spanning trees. In particular, we provide polynomial time algorithms for each of the above problems.

In each case the dynamic/periodic problem reduces to a static problem on a finite graph. However, in no case does the dynamic/periodic graph theoretic problem reduce to the same problem on a finite graph; (e.g., determining the strongly connected components of a dynamic graph does not reduce to finding the strongly connected components of a related static graph unless we allow the related static graph to have an exponentially large number of vertices.) Moreover, there are some problems that cannot be solved in polynomial time on

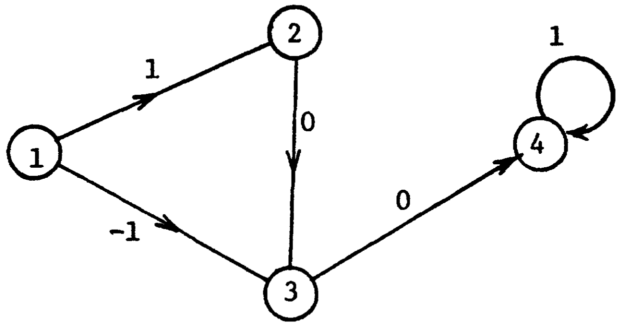


Figure 1. A static network. The arc numbers are the transit times.

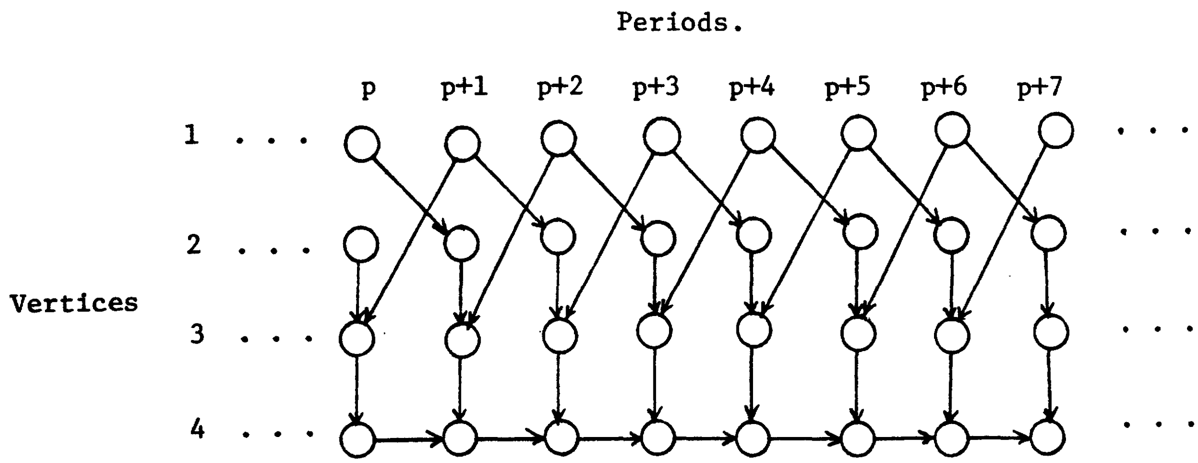


Figure 2. The dynamic network derived by expanding the static network of Figure 1.

dynamic graphs. For example, we show that determining if there is a directed path from vertex u to vertex v in a dynamic graph is NP-complete.

This paper may be viewed as a companion paper to Orlin (1981b) in which the author analysed the complexity of the classical "difficult problems" of graph theory as generalized to dynamic graphs. In that paper, the author showed that many NP-complete problems become PSPACE-hard when generalized to dynamic graphs.

Some other dynamic/periodic combinatorial optimization problems that are solvable in polynomial time include: the minimum cost-to-time ratio circuit problem (see Dantzig et al. (1979) and Lawler (1967)), the problem of minimizing the number of vehicles to meet a fixed periodic schedule (see Bartlett (1957), Bartlett and Charnes (1957), Dantzig and Simpson (1962), Simpson (1969), Wollmer (1980), and Orlin (1982a and b)). In fact, both of the above problems are special cases of both the minimum cost dynamic network flow problem solved by Orlin (1981a) and of the dynamic matching problem solved by Orlin (1982c).

The results of this paper contrast with those of Ford and Fulkerson (1958) who considered a finite horizon version of the dynamic network and those of Gale (1959) who considered dynamic networks with a fixed starting period (and the starting period was an essential element of his theory.)

Some Empirical Observations

There is no guaranteed method of assessing the complexity of a problem on a dynamic graph given only the complexity of the problem on finite graphs. Nevertheless, there are certain "rules of thumb" for assessing the complexity, and these rules are quite robust in practice.

(1) If X is an NP-hard problem on graphs, then the problem X as applied to dynamic graphs is strongly PSPACE-hard. (See Orlin (1981b) for a detailed explanation of this phenomenon).

(2) If X is a graph problem that may be solved in polynomial time, then problem X as applied to dynamic graphs may also be solved in polynomial time. Moreover, we may solve the dynamic variant of X as follows. First define an "appropriate" graph problem X' on static graphs that is solvable in polynomial time. Next show that a feasible solution for X' on G induces a feasible solution for X on G^{∞}. Finally, show via a duality result for X' that an optimal solution for X' on G induces an optimal solution for X on G^{∞}.

Although the above rule of thumb is accurate in general, it does fail for specific problems. Perhaps its failure in certain instances is not so surprising since a corollary of the rule of thumb would be that P = NP if and only if P = PSPACE.

2. Elementary Properties of Dynamic Graphs

Most of the polynomial time algorithms developed in succeeding sections rely on classical elementary results from graph theory along with two other elementary results that are established in this section. In particular we establish two connections between the circuits of the static graph and paths of the dynamic graph.

Graph Theoretic Terminology

Suppose that G is either a static or a dynamic graph. A path in G is an alternating sequence of vertices and edges $P = v_0, e_1, \ldots, e_k, v_k$ where $e_i = (v_{i-1}, v_i)$ or else $e_i = (v_i, v_{i-1})$. In the former case edge e_i is a forward edge of P; in the latter case, edge e_i is a backward edge of P. If all edges are forward, then P is called a directed path. The transit time of a path P is the sum of the transit times of the forward edges of P minus the sum of the transit times of the backward edges of P, and we denote it as $t(P)$.

A path is called closed if the initial vertex of the path is also the terminal vertex. A closed path P is called trivial if each edge $e \in E$

occurs an equal number of times as a forward edge and as a backward edge of P. A <u>circuit</u> of G is a non-trivial closed path $P = v_0, e_1, \ldots, v_{k-1}, e_k, v_k$ such that $v_0, \ldots, v_{k-1}$ are distinct.

The vertex v^p will be called the p^{th} <u>copy</u> of vertex v and the edge (u^r, v^p) for $t_{uv} = p-r$ will be called the r^{th} <u>copy</u> of edge (u,v). For any path P in G^∞ whose sequence of vertices is $v_0^{r_0}, v_1^{r_1}, \ldots, v_k^{r_k}$ there is a corresponding path P' in G whose vertex sequence is $v_0, v_1, \ldots, v_k$ and such that the j^{th} edge has a transit time $r_j - r_{j-1}$. Similarly P' induces an infinite number of copies in G^∞, and path P above is the r_0^{th} <u>copy</u> of P. The correspondence between paths in G and paths in G^∞ is described in the following Lemma.

LEMMA 1. Suppose that G = (V,E,T) is a static graph. Let u,v be vertices of V and suppose that r, p $\in$ Z. Then there is a 1:1 canonical correspondence between the set of finite paths from u^r to v^p in G^∞ and the set of paths in G from u to v with transit time p-r. □

The above elementary lemma was proved by the author (1982b) via a simple inductive argument.

LEMMA 2. Let G = (V,E,T) be a static graph and let C be a circuit of G. If t(C) = 0, then C induces an infinite number of vertex disjoint circuits of G^∞. If $t(C) \neq 0$, then C induces $|t(C)|$ vertex-disjoint infinite length paths in G^∞. □

Lemma 2 is an elementary consequence of Lemma 1 as can be seen by concatenating the infinite number of paths induced by a circuit C with $t(C) \neq 0$.

We illustrate Lemma 2 as follows. Consider the circuit C = 1,(1,2),2,(2,3),3,(1,3),1 of Figure 1. Then t(C) = 2 and one can observe that

C induces 2 vertex-disjoint infinite length paths in the dynamic graph of Figure 2.

A <u>vertex assignment</u> for a static graph $G = (V,E,T)$ is an integral valued vector d with $|V|$ components. Each vertex assignment d induces a <u>reduced transit time</u> vector $T^d = (t^d_{uv})$ where

$$t^d_{uv} = t_{uv} + d_u - d_v .$$

We let $t^d(P)$ denote the transit time of a path in which T is replaced by T^d.

The following lemma is a well-known property of reduced transit times and is easily established by induction.

LEMMA 3. Suppose that $G = (V,E,T)$ is a static network and that d is a vertex assignment. Then for any path P from vertex u to v it follows that $t^d(P) = t(P) + du - d_v$.□

LEMMA 4. Suppose that $G = (V,E,T)$ is a static graph and that d is a vertex assignment. Then the dynamic graph induced by G is isomorphic to the dynamic graph induced by $\overline{G} = (V,E,T^d)$.

PROOF. Let $f : V^\infty \to V^\infty$ be defined as follows: $f(v^p) = v^{p-d_v}$ for all $v \in V$ and $p \in Z$. Then (u^p, v^r) is an edge of G^∞ if and only if $(f(u^p), f(v^r))$ is an edge of $\overline{G}^\infty$.□

3. Connectivity In Dynamic Graphs

In this section we consider the following three problems for a given dynamic graph G^∞. First, what are the connected components of G^∞? Second, what are the strongly connected components of G^∞? Thirdly, for two specified vertices u, v of V^∞, is there a directed path from u to v? For the first two problems we provide polynomial time algorithms. The third problem is

NP-hard and is linearly equivalent to the knapsack problem.

Weak Connectivity

In order to determine the weakly connected components of the graph G^{∞} we first make two simplifying assumptions.

(1) The static graph $G = (V,E,T)$ is connected,

and (2) G has a spanning tree S such that each arc of S has a transit time of 0.

The first assumption is routine since if G is the union of connected components $G_1, \ldots, G_k$ then the set of components of G^{∞} is the union of the sets of components of $G_1^{\infty}, \ldots, G_k^{\infty}$.

We also can make assumption 2 without loss of generality because of Lemma 4. In particular we can choose a vertex $v \in V$ and let d_u be the distance in S from v to u for all $u \in V$. Then each edge of S in $\overline{G} = (V,E,T^d)$ has a transit time of 0.

For each integral valued vector w we let $\gcd(w)$ be the greatest common divisor of the components of w. For example, $\gcd(-12, -20, 30) = 2$.

THEOREM 1. Let $G = (V,E,T)$ be a static graph and suppose that there is a spanning tree of G consisting of edges whose transit time is 0. Then the number of components of G^{∞} is $\gcd(T)$.

PROOF. Let $g = \gcd(T)$. We first observe that the number of components of G^{∞} is at least g since by Lemma 1 there is no path from v^p to v^{p+j} for $1 \le j \le g-1$.

To see that G^{∞} has exactly g components, let

$$V_i = \{v^r \in V^{\infty} : r \equiv i \pmod{g}\} \quad \text{for } i = 1, \ldots, g.$$

We shall show that the subgraph of G^{∞} induced by V_i is connected and thus

G^∞ has at most g components.

By assumption there is a path P from v to w in G with $t(P) = 0$. Thus by Lemma 1, v^r is in the same component as w^r for all $v, w \in V$ and $r \in Z$. It follows that v^r is in the same component as v^{r+g} if and only if the subgraph of G^∞ induced by V_r is connected. Thus it suffices to show that there is a path P in G from v to v such that $t(P) = g$.

To this end let S denote the spanning tree whose edges have a transit time of 0, and for each $(u,v) \notin S$ let C_{uv} denote the unique circuit induced by adding edge (u,v) to S. By assumption $t(C_{uv}) = t_{uv}$.

Let $\lambda = (\lambda_{uv})$ be an integral vector determined by Euclid's algorithm such that

$$\sum_{(u,v)\in E-S} \lambda_{uv} t_{uv} = g \ .$$

Let C^* be the eulerian graph consisting of $\lambda^+_{uv} = \max(0,\lambda_{uv})$ copies of C_{uv} and λ^-_{uv} copies of the reversal of C_{uv} for each $(u,v) \in E-S$ and two copies of S so as to ensure that C^* is connected. Then C^* induces a path from v to v of transit time g, completing the proof.□

We observe that we can obtain a spanning tree and compute all reduced costs in O(E) steps using virtually any tree search approach, and we can compute the gcd in $O(|E| \log (t_{max} + 1))$, where $t_{max} = \max(|t_{uv}| : (u, v)\in E)$. Also note that the components of G^∞ are the subgraphs of G^∞ induced by the vertex sets $V_1, \ldots, V_g$ as defined in the above proof. Finally, we observe that each component of G^∞ is isomorphic to the dynamic graph induced by $G' = (V,E,T')$ where $t'_{uv} = t_{uv}/g$.

Strong Connectivity

A graph is strongly connected if for every ordered pair u, v of vertices there is a directed path from u to v. The strongly connected components of a graph G are the maximal induced subgraphs that are strongly connected. There are several very efficient algorithms for computing the strongly connected components including Tarjan's (1972) algorithm that runs in $O(|E|)$ steps using a depth first search approach.

In order to determine the strongly connected components of a dynamic graph G^∞ we first make 2 simplifying assumptions:

(1) G is strongly connected,

and (2) G^∞ is connected.

If G were not strongly connected, we could determine the strongly connected components of G^∞ by applying the procedure described below to each of the strongly connected components of G. Similarly, if G^∞ were not connected, then each component of G^∞ is itself a dynamic graph as mentioned above, and thus we could apply the algorithm below to each of the components.

We shall develop an algorithm for determining the strongly connected components of G^∞ by considering two separate cases.

THEOREM 2. Suppose that $G = (V,E,T)$ is a strongly connected static graph and that G^∞ is connected. Suppose in addition that there are directed circuits C^- and C^+ in G such that

$$t(C^-) < 0 < t(C^+) .$$

Then G^∞ is strongly connected.

PROOF. Let S be an arborescence of G with root $\bar{v}$ for some $\bar{v} \in V$. (An arborescence with root $\bar{v}$ is a spanning tree in which there is a directed path from $\bar{v}$ to every other vertex.) Without loss of generality, assume that each edge of S has a transit time of 0.

(Otherwise, let d_u be the distance of the path in S from $\bar{v}$ to u and replace T by T^d). Such an arborescence exists because there is a path from $\bar{v}$ to every other vertex.

We will show below that there is a path from $\bar{v}^p$ to $\bar{v}^r$ for any $p, r \in Z$. In this case, the graph G^∞ is strongly connected. To see this, note that since G is strongly connected there is a directed path P from u to $\bar{v}$ in G and thus a path from u^r to $\bar{v}^{r+t(P)}$ in G^∞ for all $r \in Z$. Similarly there is a directed path from $\bar{v}$ to w in S and thus there is a path from $\bar{v}^p$ to w^p in G^∞. Thus if we assume that there is a path from $\bar{v}^{r+t(P)}$ to $\bar{v}^p$ then there is also a path from u^r to w^p in G^∞ for any u^r, $w^p \in V^\infty$.

Our proof that there is a path P from $\bar{v}$ to $\bar{v}$ with $t(P) = 1$ proceeds similarly to the proof of Theorem 1. For each $v \in V$, let P_v denote the path on S from $\bar{v}$ to v and let P'_v denote some path in G from v to $\bar{v}$. Let $C_v = P_v, P'_v$ and let $D_{uv} = P_u, (u, v), P'_v$. We have constructed the directed circuits C_v and D_{uv} so that

$$t(D_{uv}) - t(C_v) = t_{uv} .$$

Let $q = |t(C^+)t(C^-)|$. By assumption, $\gcd(T) = 1$ and thus we can use Euclids algorithm to determine an integral vector $\lambda = (\lambda_{uv})$ such that

$$\sum_{(u,v)\in E} \lambda_{uv} t_{uv} \equiv 1 \pmod q$$

and $\quad 0 \le \lambda_{uv} \le q-1$ for all $(u, v) \in V$.

Let C^* be the directed eulerian graph which is the sum of λ_{uv} copies of C_v and $q - \lambda_{uv}$ copies of D_{uv} for all $(u, v) \in E$. Then

$$t(C^*) = \sum_{(u,v)\in E} \lambda_{uv} t_{uv} + \sum_{(u,v)\in E} q t(P'_v) \equiv 1 \pmod q .$$

We can then add sufficient multiples of C^+ and C^- so as to obtain a directed circuit whose transit time is 1. Thus there is a path from $\bar{v}^p$ to $\bar{v}^{p+1}$ for all $p \in Z$. Similarly,

$$\sum_{(u,v)\in E} (q-\lambda_{uv})t_{uv} \equiv -1 \pmod q$$

so that we can construct a path from $\bar{v}^{p+1}$ to $\bar{v}^p$ for each $p \in Z$. Thus there is a directed path from $\bar{v}^p$ to $\bar{v}^r$ for all $p, r \in Z$, completing the proof.□

The remaining case to analyse is the case in which either all circuits C of G are such that $t(C) \geq 0$ or else all circuits are such that $t(C) \leq 0$. The cases are symmetric, so that we may consider without loss of generality only the case that each directed circuit of G has a non-negative transit time.

Since $G = (V,E,T)$ is strongly connected and has no negative transit time circuits, there is a spanning tree S rooted at vertex $\bar{v} \in V$ (for any specified $\bar{v} \in V$) such that the path in S from $\bar{v}$ to u is the minimum distance directed path in G from $\bar{v}$ to u. Let us assume without loss of generality that each edge of S has a transit time of 0 and thus $T \geq 0$.

LEMMA 5. Let $G = (V,E,T)$ be a strongly connected directed graph with $T \geq 0$ and such that there is a directed path from $\bar{v}$ to u of transit time 0 for each $u \in V$. Then

(i) u^p and v^r are in different strongly connected components if $p \neq r$,

(ii) u^p and v^p are in the same strongly connected component of G^∞ if and only if there is a directed path from u to v and another directed path from v to u in G such that each path consists only of edges whose transit time is 0.

PROOF. If $p > r$, then there is no path P in G from u to v with $t(P) = r - p < 0$ and thus by Lemma 1 there is no path in G^∞ from u^p to v^r. If $p < r$, then there is no path in G^∞ from v^r to u^p.

If $p = r$ then u^p and v^r are in the same strongly connected component if and only if there is a path P from u to v with $t(P) = 0$ and also a path P' from v to u with $t(P') = 0$. Since $T \geq 0$, it follows that all directed paths P with $t(P) = 0$ must consist solely of edges of transit time 0.□

To summarize the results concerning the strongly connected components, we first decompose G into strongly connected components and further decompose if necessary so that each induced dynamic graph is connected.

We then consider each resulting strongly connected static graph $\overline{G}$ in the partition. If $\overline{G}$ has both positive transit circuits and negative transit time circuits, then $\overline{G}^\infty$ is strongly connected. Otherwise, each strongly connected component of G^∞ has at most $|V|$ vertices, and these components can be located by letting S be a minimum distance spanning tree (resp., maximum distance spanning tree if $t(C) \leq 0$ for all circuits C) and applying Lemma 5.

PROBLEM: DIRECTED PATHS IN G^∞

INPUT: A directed graph $G = (V,E,T)$ and two vertices u^p, v^r of G^∞.

QUESTION: Is there a path from u^p to v^r in G^∞?

THEOREM 3. The directed path problem in G^∞ is NP-complete.

PROOF. By Lemma 1, there is a directed path in G^∞ from u^p to u^r if and only if there is a directed path in G from u to v with transit time $r - p$. We shall show that this latter problem is transformable to the knapsack recognition problem which was proved to be NP-complete by Karp (1972), and can be described as follows:

INPUT: non-negative integers $a_1,\ldots,a_n$, b.

QUESTION: is there an index set $S \subseteq \{1,\ldots,n\}$ such that $\Sigma_{i\in S}\, a_i = b$?

Let $G = (V,E,T)$ be the graph described in Figure 3. Then it is clear that there is a path from vertex 1 to vertex n+1 of transit time b if and only if there is an index set S such that $\Sigma_{i\in S}\, a_i = b$.□

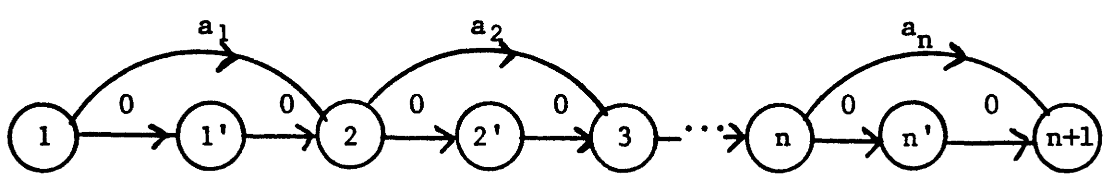


Figure 3. A directed graph such that there is a path from 1 to n+1 of transit time b if and only if there is an index set S with $\Sigma_{i\in S}\, a_i = b$.

Below we consider the problem of determining a minimum cost path in G from $\bar{v}$ to u of transit time r. In particular, we show that the problem is solvable in pseudo-polynomial time. A problem is solvable in pseudo-polynomial time if it is solvable in time polynomial in the data and in the largest integer in the data, in this case $\max(c_{max}, t_{max}, r)$. Such an algorithm is not possible for the following NP-complete problem. Is there a simple directed path P in G from $\bar{v}$ to u such that $t(P) = r$. This latter problem is easily shown to be a generalization of the hamiltonian path problem.

Let $f(u, p, j)$ be the minimum cost of a path from $\bar{v}$ to u of transit time p and with at most j edges, and let

$$f(v,p) = \min_{1\le j<\infty} f(v, p, j) \quad \text{for each } p \in Z \text{ and } v \in V.$$

The pseudo-polynomial algorithm for computing $f(u, r)$ is a consequence of Lemma 6 below.

LEMMA 6. Let $G = (V,E,T)$ be a network with transit times. Suppose $\hat{t} = |V| t_{max}$ and $t^* = 2|V|(\hat{t} + |V| + |r|)^3$. Then $f(u, r) = f(u, r, t^*)$ or else there is no minimum cost path from $\bar{v}$ to u of transit time r.

<u>PROOF</u>. We develop a proof by contradiction. Suppose that P is a minimum cost path from $\bar{v}$ to u such that $t(P) = r$ and such that P has a minimum number of edges with respect to all such paths. Suppose further that P has at least t^* edges.

By flow decomposition theory, we can decompose path P into the sum of one directed path P_u from $\bar{v}$ to u and a collection $\mathscr{C}$ of directed circuits. Let S be the union of P_u and at most $|V|$ circuits of $\mathscr{C}$ so that S is strongly connected and such that the vertex set of S is the vertex set of P. Let $\mathscr{C}^* = \mathscr{C} - S$. Thus we can delete any collection of circuits of $\mathscr{C}^*$ from P and the resulting graph is connected.

Let n_p be the number of circuits of $\mathscr{C}^*$ that have a transit time equal to p.

Since each circuit $C \in \mathscr{C}^*$ has at most $|V|$ edges and $|t(C)| \le \hat{t}$, it follows that the number of circuits in $\mathscr{C}^*$ is

$$\sum_{p=-\hat{t}}^{\hat{t}} n_p \ge (t^* - |V|^2)/|V| \ge 2\hat{t}^3 + |r| + |V| \quad . \tag{3.1}$$

Moreover, since $t(P) = r$ it follows that

$$\sum_{p=-\hat{t}}^{\hat{t}} p n_p = r - \sum_{(u,v) \in S} t_{uv} \quad . \tag{3.2}$$

We next note that $n_0 = 0$. Else there is a circuit $C \in \mathscr{C}^*$ with $t(C) = 0$. If the cost of C is negative, then P is not a minimum cost path. If the cost of C is not negative, then P is not an optimum path with the fewest number of edges.

From (3.2) and the fact that $n_0 = 0$ we obtain the inequality

$$\sum_{p=-\hat{t}}^{-1} -pn_p \geq \sum_{p=1}^{\hat{t}} pn_p - r - |V|\hat{t} \tag{3.3}$$

and from (3.3) we obtain the inequality

$$\sum_{p=-\hat{t}}^{-1} n_p \geq (\hat{t})^{-1} \sum_{p=1}^{\hat{t}} n_p - |r| - |V|\hat{t} \quad . \tag{3.4}$$

Combining (3.1) and (3.4) we see that the number of circuits in $\mathscr{C}^*$ with a negative transit time is at least $\hat{t}^2$. Using a symmetric argument, we can show that the number of circuits in $\mathscr{C}^*$ with a positive transit time is at least $\hat{t}^2$. By the pigeon hole principle, there are integers p and q such that $p < 0 < q$ and $n_p, n_q \geq t$. Let C^* be the sum of q circuits of $\mathscr{C}^*$ with transit time p plus another $-p$ circuits of $\mathscr{C}^*$ with transit time q. Then $t(C^*) = 0$, and we can devise the same contradictions as before.□

In order to translate Lemma 6 into a result concerning dynamic graphs, we first define $V_p = \{v^j : v \in V \text{ and } |j| \leq p\}$ and we let G^p be the subgraph of G^∞ induced by the vertex set V_p .

COROLLARY 1. Suppose that $G = (V,E,T)$ is a static graph. If there is a minimum cost path from u^0 to v^r in G^∞ then there is such a minimum cost path all of whose edges are in G^p for $p = t_{max} \cdot t^*$.

PROOF. By Lemma 6 we can restrict attention to paths with at most t^* edges and thus whose transit time is at most $t_{max} \cdot t^*$.□

4. Eulerian Paths

In what is usually credited with being the first paper in graph theory, Euler (1736) showed that there is a closed path that passes through each edge exactly once if and only if (1) the graph is connected and (2) every vertex has even degree. In this section we consider the problem of determining eulerian paths on dynamic graphs, i.e., paths that pass through each edge of the dynamic graph exactly once. It is easy to see that conditions (1) and (2) above are still necessary, but they are no longer sufficient. For instance, the dynamic graph of Figure 4.1 satisfies both conditions even though there is no eulerian path.

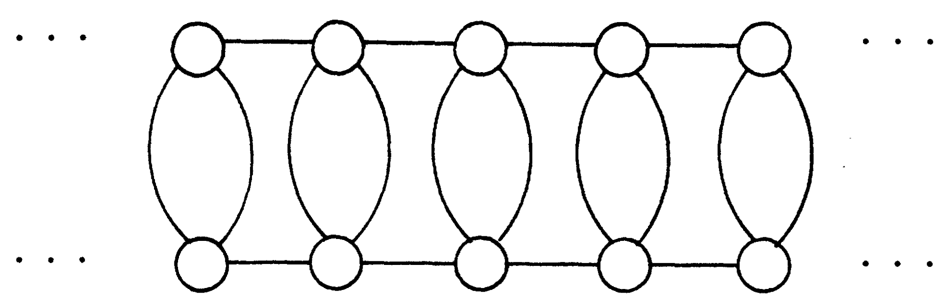


Figure 4.1 A dynamic graph in which all vertices have degree 4 and for which there is no (infinite) eulerian path.

Below we provide necessary and sufficient conditions for dynamic graphs to have either directed eulerian paths or undirected eulerian paths.

THEOREM 3. Suppose that $G = (V,E,T)$ is a static graph. Then G^∞

has a directed eulerian path if and only if each of the following conditions holds:

(1) The indegree of each vertex of G is equal to its outdegree,

(2) G is connected,

and (3) $\left| \sum_{(u,v)\in V} t_{uv} \right| = 1$.

PROOF. The necessity of conditions (1) and (2) are obvious. We see the necessity of condition (3) as follows. First, suppose that P is any infinite path in G^{∞} that passes through each edge at most once. We say that P crosses the origin from below (resp., from above) at edge $(u^p, v^r) \in P$ if $p < 0$ and $r \geq 0$ (resp., $p \geq 0$ and $r < 0$) . Let $b^+(P)$, (resp., $b^-(P)$) denote the number of crossings of P from below (resp., from above) . Since there is a crossing from above between every two crossings from below, and there is a crossing from below between every two crossings from above, and since $b^+(P)$, $b^-(P)$ are finite it follows that $|b^+(P) - b^-(P)| \leq 1$. Moreover, if P is eulerian then the number of crossings from below cannot equal the number of crossings from above. (If the first crossing is from below and if the last crossing is from above, then the path cannot pass through an infinite number of edges (u^p, v^r) with $p, r \geq 0$, contradicting that P is eulerian. We can derive a similar contradiction if the first crossing is from above and the second crossing is from below). Thus

$$|b^+(P) - b^-(P)| = 1 .$$

Furthermore, if P is eulerian then

$$b^+(P) = \sum_{(u,v)\in V} \max(0, t_{uv}) ,$$

and

$$b^-(P) = \sum_{(u,v)\in E} \max(0,-t_{uv})$$

so that

$$b^+(P) - b^-(P) = \sum_{(u,v)\in E} t_{uv} .$$

We see the sufficiency of conditions (i), (ii), and (iii) as follows. Let C be a directed eulerian cycle in C initiating and ending at vertex v. (Conditions (i) and (ii) ensure that such a cycle exists). Then the p^{th} copy of C is a directed path in G^∞ from v^p to v^{p+1} or from v^p to v^{p-1} according as $t(C) = +1$ or $t(C) = -1$. In either case, we can concatenate all of the copies of C so as to form an eulerian path.□

THEOREM 4. Suppose that $G = (V,E,T)$ is a static graph. Then G^∞ has an undirected eulerian path if and only if each of the following conditions holds:

(1) Each vertex of G has even degree,

(2) G^∞ is connected,

and (3) $\sum_{(u,v)\in V} t_{uv} \equiv 1 \pmod 2$.

PROOF OF NECESSITY OF CONDITIONS (1), (2) and (3). The necessity of conditions (1) and (2) are immediate. The necessity of condition (3) is proved analogously to the necessity of condition (3) of Theorem 3.

First, suppose that P is an undirected infinite path in G^∞. We say that P <u>crosses the origin from below</u> at edge (u^p, v^r) if $p < 0$ and $r \geq 0$ and (u^p, v^r) is a forward edge of P or else $p \geq 0$ and $r < 0$ and (u^p, v^r) is a backward edge of P. We say that P crosses the origin from above at edge

(u^p, v^r) if the reversal of P croses the origin from below at edge (u^p, v^r). As before, we let $b^+(P)$ (resp., $b^-(P)$) denote the number of crossings from below (resp., above). The necessity of condition (3) follows from the fact if P is an eulerian path then

$$|b^+(P) - b^-(P)| = 1$$

and

$$\sum_{(u,v)\in V} t_{uv} \equiv b^+(P) - b^-(P) \pmod 2. \square$$

In order to prove the sufficiency of (1), (2) and (3), we reduce the undirected eulerian path problem on dynamic graphs to the directed eulerian path problem. In order to perform the transformation, we will first consider a variant of dynamic graphs.

Let $G^* = (V^\infty, E^*)$ where $V^\infty = \{v^p : v\in V \text{ and } p\in Z\}$

We say that E^* is <u>periodic</u> with period q if the edges of E^* have the following property:

$$(u^r, v^p)\in E^* \text{ if and only if } (v^{r+q}, v^{p+q})\in E^* .$$

REMARK. Suppose that $G^* = (V^\infty, E^*)$ is an infinite graph such that E^* is periodic with period q and each vertex of V^∞ is incident to a finite number of other vertices. Then G^* is a dynamic graph induced by a static graph with $q|V|$ vertices.

PROOF. Let $\overline{G} = (\overline{V},\overline{E},\overline{T})$ where $\overline{V} = \{v^p : v\in V \text{ and } 1\le p\le q\}$. For each edge $(u^r, v^p)\in E^*$ with $1\le r\le q$ we choose k and t so that $p - r = k + tq$ and $1\le k\le q$ and we associate an edge $(u^r, v^k)\in\overline{E}$ with transit time t.□

PROOF OF SUFFICIENCY OF (1), (2) and (3).

Since G is eulerian we can express G as the union of undirected circuits $C_1, \ldots, C_k$, each with a non-negative transit time. Let us assume without loss of generality that each circuit is directed. Otherwise we could perform a series of edge reversals so as to obtain directed circuits. (An edge reversal is the replacing of edge (u, v) by an edge (v, u) whose transit time is $t_{vu} = -t_{uv}$. Edge reversals do not effect transit time of circuits, and they change arc directions in G^∞.)

Order the circuits so that $t(C_1) \le \ldots \le t(C_k)$ and let

$$t^* = \sum_{i=1}^{k} t(C_i) = \sum_{(u,v)\in E} t_{uv} .$$

Next choose index j and $k \in Z$ such that

$$k + \sum_{i=1}^{j-1} t(C_i) = (t^*+1)/2$$

and $$1 \le k \le t(C_j) .$$

Reorient the edges of G^∞ again so that

(i) all of the $\sum_{i=1}^{j-1} t(C_i)$ infinite paths in G^∞ induced by $C_1, \ldots, C_{j-1}$ are reversed (i.e., each edge of of these paths in G^∞ is reversed)

and (ii) k of the infinite paths induced by C_j are reversed.

and (iii) all other edges keep their current orientation.

It can be verified that the resulting infinite graph G^* is periodic with period $t(C_j)$ and that the static graph that induces G^* satisfies the conditions of Theorem 3. Hence G^∞ is eulerian.□

The condition that G^{∞} is connected is a necessary condition and cannot be replaced by the condition that G is connected as in the directed case. For example the graph $G = (V,E,T)$ with $V = \{v\}$, $E = \{(v, v)\}$ and $t_{vv} = 3$ satisfies conditions 1 and 3 and G is connected. However, G^{∞} fails to be connected.

5. 2-Colorability

As is well known, a graph G (infinite or not) is bipartite (2-colorable) if and only if there is no undirected circuit with an odd number of edges.

REMARK. Suppose that $G = (V,E,T)$ is a static graph. Then the induced dynamic graph G^{∞} is bipartite if and only if there is no closed path P in G with an odd number of edges and with $t(P) = 0$.

The above remark is an elementary consequence of Lemma 1. As an example, consider the static graph $G = (V,E,T)$ of Figure 5.1. Then G^{∞} is bipartite. However, if the transit time of (1, 2) were changed from 0 to 2p+1 then there would be an odd circuit in G^{∞} with 2p+9 edges. (If we take 2 copies of the triangle in G and 2p+3 copies of the loop then we obtain an odd closed path P in G with $t(P) = 0$).

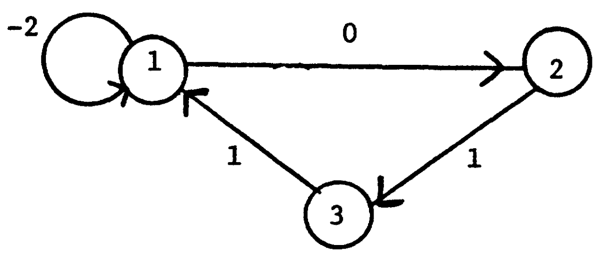


Figure 5.1. A graph G whose induced dynamic graph is bipartite.

We observe that the shortest odd circuit in G^{∞} would have $2p+9$ edges and thus has an exponential number of edges as p gets large. (Recall that the length of the input string is $O(\log p)$). It is fortunate that we can detect the odd circuit much faster than we can list the edges of the circuit.

Before proceeding to the algorithm we define the graph $G^2 = (V^2, E^2)$ as follows:

$$V^2 = \{v^1, v^2 : v \in V\} ,$$

$$E = \{(u^i, v^j) : (u, v) \in E, i,j \in \{1, 2\}, t_{uv} \equiv j - i \pmod 2\}$$

THEOREM 5. Suppose that $G = (V,E,T)$ is a static graph and that G^{∞} is connected. Then G^{∞} is bipartite if and only if G^2 is bipartite.

PROOF. Suppose first that G^{∞} is not bipartite. Then there is an odd length closed path P in G such that $t(P) = 0$. Then P induces an odd length closed path in G^2, and thus G^2 is not bipartite.

Suppose conversely that G^2 is not bipartate. Then there is an odd length closed path in G^2, and this path induces a closed path P in G such that $t(P) \equiv 0 \pmod 2$. Let $k = t(P)/2$; and let v be the initial vertex and terminal vertex of P. By assumption G^{∞} is connected and thus there is some closed walk P' in G from v to v with $t(P') = -k$. Then $P^* = P,P',P'$ is an odd closed walk in G and $t(P^*) = 0$, thus showing that G^{∞} is not bipartite □.

We observe that the conclusions of Theorem 5 would not be true if we dropped the assumption that G^{∞} is connected. In particular, the graph of Figure 5.1 induces a dynamic graph G^{∞} that is bipartite whereas G^2 is not bipartite.

6. Minimum Average Cost Spanning Trees

Here we consider the extension of the minimum cost spanning tree problem to dynamic graphs. Let $G = (V,E,T,C)$ be a static graph in which each edge $(u, v)\epsilon E$ has a cost c_{uv}. Moreover, the cost of each copy of edge (u, v) in the dynamic graph $G^{\infty} = (V^{\infty},E^{\infty},C^{\infty})$ is c_{uv}. A <u>dynamic spanning tree</u> refers to a spanning tree of G^{∞}.

Let $G^p = (V^p, E^p)$ be the subgraph of G^{∞} induced by the vertex set $V^p = \{v^r : v\epsilon V \text{ and } -r\leq p\leq r\}$. For each dynamic spanning tree S let $c^p(S)$ be the sum of the costs of the edges in $S \cap E^p$. The <u>dynamic spanning tree problem</u> is to determine a dynamic spanning tree S that minimizes the long run average cost per period, i.e., the value

$$\liminf_{p \to \infty} (2p+1)^{-1}c^p(S) .$$

To determine a minimum average cost spanning tree for G^{∞} it is possible to use the standard approaches of Kruskal (1956) on G^p and let p approach infinity. Instead we use Edmond's (1971) greedy algorithm to determine a minimum weight basis of the matroid M defined below. In fact, the greedy algorithm as applied to M may be interpreted as an efficient implementation of the greedy algorithm for trees as applied to E^{∞}.

For each static graph $G = (V,E,T)$ we define the <u>quasi-dynamic matroid</u> (or <u>Q-matroid</u>) to be the matroid $Q(G) = (E, I)$ on the edge set of G such that a subset $A \subseteq E$ is independent if (1) there is at most one circuit in A and (2) there is no simple circuit D in A with $t(D) = 0$. For example, the edges of the graph in Figure 6.1 are independent, but the edges in

Figures 6.2a and 6.2b are not. An equivalent definition is that the subset A is independent if there is no non-trivial closed path in A whose transit time is 0.

To verify that M defined above is a matroid, we will show that M is equivalent to another well known matroid. For a given network $G = (V,E,T)$, let us define another network $G' = (V,E,T')$ where $t'_{uv} = \exp(t_{uv})$. For each path P in G', let f(P) be the product of the forward edges of P divided by the product of the backward edges of P. It is easy to see that $t(P) = 0$ if and only if $f(P) = 1$.

Then G' may be interpreted as a generalized network. In such a network, if x_{uv} is the flow originating in edge (u,v), then the flow arriving at the head of (u,v) is $t'_{uv}x_{uv}$. Let A_{uv} be the column vector with a 1 in component u and a "$-t'_{uv}$" in component v. Then it is well known (see, for example, Dantzig (1963)) that a subset of columns of $A = (A_{uv})$ is linearly independent if and only if there is no non-trivial closed path P consisting of arcs of A such that $f(P) = 1$ or, equivalently, $t(P) = 0$.

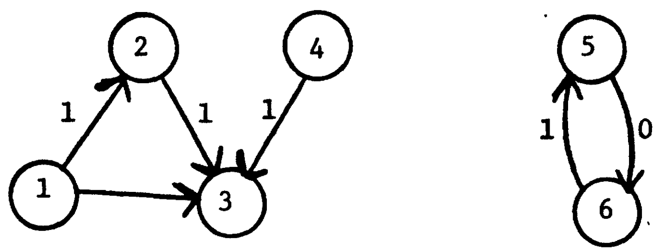
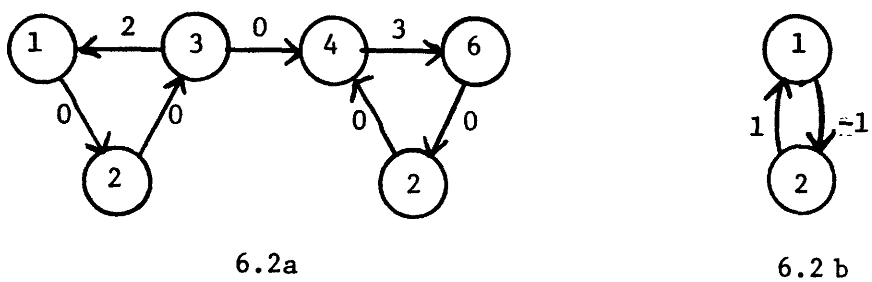


Figure 6.1. An independent set of edges with respect to the quasi-dynamic matroid of G.



Figures 6.2a and 6.2b. Dependent sets of edges with respect to the quasi-dynamic matroid of G.

THEOREM 6. Let $G = (V,E,T,C)$ be a static graph such that G^{∞} is connected and suppose that $A \subseteq E$ is a minimum cost basis for the Q-matroid Q(G). Then the forest F_A obtained as the union of the infinite number of copies of A may be extended to a minimum average cost dynamic tree via the addition of a finite number of edges.

PROOF. By Lemma 1, F_A has no simple circuit. We see that F_A may be extended to a dynamic tree through the addition of a finite number of edges as follows. First, G must have some non-zero length circuit and must be connected since G^{∞} is connected. Thus A must have $|V|$ edges, as every basis of Q(G) has $|V|$ edges. Since no component H of A can have two circuits, it follows that each component H of A has $|V(H)|$ edges and exactly one circuit. Thus the dynamic graph induced by such a component H induces k components of G^∞ where k is the transit time of the circuit of H. It follows that F_A has a finite number of components and can be extended to a spanning tree S_A by the addition of a finite number of edges. Also the average cost per period c' of S_A is the same as that of F_A.

To see that S_A is the minimum average cost dynamic spanning tree of G^∞, let S' be any other spanning tree. By the construction of the greedy algorithm it follows that $F_A \cap E^p$ is a subset of the minimum cost spanning tree of G^p. Also $S' \cap E^p$ may be extended to a spanning tree of G^p by the addition of at most $2t_{max}$ edges, and these edges have a total cost bounded above by $c^* = 2c_{max}t_{max}$. Thus

$$c^p(S') \geq c^p(F_A) - c^*,$$

and thus

$$\liminf_{p \to \infty} (2p+1)^{-1}c^p(S') \geq \liminf_{p \to \infty} (2p+1)^{-1}c^p(F_A) - c^* = c'.\square$$

7. Summary

In the previous sections we have shown how to solve various problems on dynamic graphs by reducing the problems to easily solved problems on finite graphs. In each case, the proof that the transformation is correct uses little more than Lemma 1 and some elementary graph theoretic analysis.

It may seem plausible that every problem on dynamic graphs reduces to a problem on the static graph G. While this reduction is indeed always possible in the trivial sense that G^∞ is itself defined in terms of G, in a very real sense there can be no such reduction in general. In particular, there is no such reduction if the reduced problem must be in the class NP, unless NP = PSPACE since the dynamic version of 3-colorability and many other graph problems is PSPACE-complete, as proved by Orlin (1981c).

There are no known general conditions which guarantee that the dynamic variant of a given problem is polynomially solvable. There are also no known general conditions which guarantee that the dynamic variant of an NP-complete problem is PSPACE-complete. There are general approaches for determining the complexity of dynamic problems, but it would be interesting if there were a broader, more encompassing theory of the complexity of these problems. It would be of theoretical interest if such a theory could be developed, and it is likely that such a theory would have much deeper ramifications into the structure of the classes NP and PSPACE.

ACKNOWLEDGMENTS

I wish to thank Rita Vachani for her careful reading of this manuscript and for her helpful suggestions.

References

Baker, K. R. (1976). Workforce Allocation in Cyclical Scheduling Problems. Operations Research Quarterly 27, 155-167.

Bartlett, T. E. (1957). An Algorithm for the Minimum Number of Transport Units to Maintain a Fixed Schedule. Naval Research Logistics Quarterly 4, 139-149.

Bartlett, T. E. and A. Charnes (1957). Cyclic Scheduling and Combinatorial Topology: Assignment and Routing of Motive Power to Meet Scheduling amd Maintenance Requirements. Part II: Generalization and Analysis, Naval Research Logistics Quarterly 4, 207-220.

Dantzig, G. B., W. Blattner, and M. R. Rao (1967). Finding a Cycle in a Graph with Minimum Cost to Times Ratio with Application to a Ship Routing Problem. In P. Rosenthiehl (ed.). Theory of Graphs. Dunod, Paris, Gordon and Breach, New York, 77-84.

Dantzig, G. B. and R. Simpson (1962). Consulting work for United Airlines.

Dantzig, G. B. (1963). Linear Programming and Extensions, Princeton University Press, Princeton, N.J.

Dhall, S. K. and C. L. Liu (1978). On a Real-Time Scheduling Problem. Operations Research 26, 127-140.

Edmonds, J. (1971). Matroids and the Greedy Algorithm. Math. Prog. 1, 127-136.

Euler, L. (1736). Solutio Probematis ad Geometriam Situs Pertinentis. Comentarii Academicae Petropolitanae 8, 128-140.

Ford, L. R. and D. R. Fulkerson (1958). Constructing Maximal Dynamic Flows from Static Flows. Operations Research 6, 419-433.

Gale, D. (1959). Transient Flows in Networks. Michigan Math Journal 6, 59-63.

Karp, R. M. (1972). Reducibility among Combinatorial Problems. In Miller and Thatcher (eds.). Complexity of Computer Computations. Plenum Press, New York.

Kruskal, J. B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proc. American Math. Soc. 7, 48-50.

Labetoulle, J. (1974). Some Theorems on Real-Time Scheduling. In E. Gelenbe and R. Mahl (eds.). Computer Architecture and Networks. North-Holland Publishing Company, 285-298.

Lawler, E. L. (1967). Optimal Cycles in Doubly Weighted Linear Graphs. In P. Rosentiehl (ed.), Theory of Graphs. Dunod, Paris, Gordon and Breach, New York, 209-214.

Lawler, E. L. and C. U. Martel (1981). Scheduling Periodically Occurring Tasks on Multiple Processors. Submitted for publication.

Liu, C. L. and J. W. Layland (1973). Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment. JACM 20 46-61.

Orlin, J. B. (1981a). Minimum Convex Cost Dynamic Network Flows. Chapter 4, Ph.D. Dissertation, Department of Operations Research, Stanford University, Stanford, California.

Orlin, J. B. (1981b). The Complexity of Dynamic Languages and Dynamic Optimization Problems. Transactions of the 13th Annual ACM Symposium on the Theory of Computing, Milwaukee, Wisconsin.

Orlin, J. B. (1982a). Minimizing the Number of Vehicles to Meet a Fixed Periodic Schedule: An Application of Periodic Posets. Operations Research 30, 760-776.

Orlin, J. B. (1982b). Maximum Throughput Dynamic Network Flows. Accepted for publication by Mathematical Programming.

Orlin, J. B. (1982c). Dynamic Matchings and Quasi-Dynamic Fractional Matchings I and II. Working Papers 1331-82 and 1332-82, Sloan School of Management, MIT.

Papadimitriou, C. and K. Steiglitz (1982). Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Englewood Cliffs, N.J.

Simpson, R. W. (1968). Scheduling and Routing Models for Airline Systems. Report FTL-R68-3, Department of Aeronautics and Astronautics, MIT, 100-107 and 128-134.

Tarjan, R. E. (1972). Depth First Search and Linear Graph Algorithms. SIAM J. Computing 1, 146-160.

Wollmer, R. D. (1980). An Airline Schedule Tail Routing Algorithm. Presented at the Fall 1980 ORSA/TIMS conference in Colorado Springs.