

DYNAMIC MATCHINGS AND QUASI-DYNAMIC
FRACTIONAL MATCHINGS, PART I

James B. Orlin*

July 1982

Working Paper #1331-82

* Work supported in part under ECS-8205022, National Science Foundation,
entitled "Research Initiation: Dynamic/Periodic Optimization Models".

Abstract

This paper presents and solves in polynomial time the dynamic matching problem, an integer programming problem which involves matchings in a time expanded infinite network. The initial model is a finite directed graph $G = (V, E)$ in which each edge has an associated real-valued weight and an integral distance. We wish to "match" vertices over an infinite horizon, and we permit vertex i in period p to be matched to vertex j in period r if and only if there is an edge $e = (i, j)$ of E with distance $r-p$ or else an edge $e = (j, i)$ of E with distance $p-r$. Equivalently, we construct a "dynamic graph" in which there is an edge incident to vertex $i-p$ and to vertex $j-r$ in the above cases. The weight of this matched edge in the dynamic (time expanded) graph is the weight of e .

The dynamic matching problem is to determine a matching M in the dynamic graph such that M has a maximum long run average weight per period. We show that the infinite horizon dynamic matching problem is linearly transformable to the finite horizon Q -matching problem, which is shown to be solvable in polynomial time in Part II of this paper.

I. Introduction

In this paper we consider the dynamic matching problem, a matching problem defined on infinite dynamic (periodic, time-expanded) graphs in which the objective is to maximize the average weight per period of a (periodic) matching. We show that the infinite-horizon dynamic matching problem is linearly transformable to the finite horizon quasi-dynamic fractional matching problem, which is shown to be solvable by a polynomial time algorithm in Part II of this paper.

One can pose other problems on dynamic graphs such as network flow problems, connectivity problems, independence set problems, etc. Orlin (1981a and b) has shown that the dynamic network flow problem is polynomially solvable, whereas the hamiltonian path problem, the maximum independent set problem, the 3-colorability problem and others are all P-Space hard. (See Garey and Johnson [1979] for definitions). Thus the polynomial solvability of the dynamic matching problem highlights the difference between polynomially-solvable graph problems and NP-hard graph problems as generalized to dynamic graphs.

2. Finite Graphs and Dynamic Graphs

Let $G = (V(G), E(G))$ be a finite directed graph such that $V(G) = \{1, \dots, n\}$, and each edge $e \in E(G)$ has an associated distance $d(e)$ and an associated weight $w(e)$. The distance $d(e)$ may be interpreted as the number of periods separating the head of e from the tail of e . From G we form an infinite undirected graph $G^\infty = (V(G^\infty), E(G^\infty))$, where

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

and v^p is called the p^{th} copy of vertex v . (Here Z denotes the set of integers.) Moreover, each edge $e=(u,v) \in E(G)$ induces an infinite set of edges $\{(u^p, v^{p+d(e)}); p \in Z\}$ each with weight $w(e)$, and $(u^p, v^{p+d(e)})$ is the p^{th} copy of edge e . The set $E(G^\infty)$ consists of all copies of all edges of $E(G)$. Any infinite graph that is isomorphic to G^∞ for some finite graph G is called a dynamic (or time expanded) graph. In Figure 2.1 b, we illustrate a dynamic graph, which is induced from the graph in Figure 2.1a.

Ford and Fulkerson [1958] used (finite horizon) time-expanded networks in order to model dynamic flows evolving over a fixed number of periods. Subsequently a number of authors have investigated dynamic network flows (either finite or infinite horizon) including Gale [1959], Minieka [1973], Orlin [1981a], Halpern [1979], Jarvis and Ratliff [1981], White [1972], and Wilkinson [1971] and [1973]. The author [1981a and b] has analyzed other graph theoretic problems on the infinite dynamic networks described above, and has shown that a number of NP-hard problems on finite graphs become P-space hard when generalized to dynamic graphs. (See Garey and Johnson [1979] for the definitions with respect to P-space hardness.)

Dynamic Matchings

A matching is a set of edges no two of which are incident with a common vertex. We will refer to a matching in a dynamic graph as a

dynamic matching. Suppose M is a dynamic matching. For each integer p , let

$$w^p(M) = \sum_{\substack{e=(u^r, v^s) \in M \\ \text{and } -p \leq r, s \leq p}} w(e)$$

Thus $w^p(M)$ is the sum of the weights of all edges in M whose endpoints are in the set $\{v^r: v \in V(G), -p \leq r \leq p\}$. A dynamic matching M is called average weight optimal if for all other dynamic matchings M' we have

$$\liminf_{p \rightarrow \infty} p^{-1}(w^p(M) - w^p(M')) \geq 0.$$

The dynamic matching problem is to determine a dynamic matching that is average weight optimal.

A dynamic matching is called periodic with period t if the following is true: the p^{th} copy of edge $e \in E(G)$ is in M if and only if the $(p+t)^{\text{th}}$ copy of edge e is in M for all $e \in E(G)$ and $p \in \mathbb{Z}$. The following elementary result is a specialization of a result proved in Orlin [1981b].

Remark 1. For each instance of the dynamic matching problem, there is an optimal average weight matching that is periodic.

Henceforth, we restrict attention only to those dynamic matchings that are periodic. Unfortunately, the periodicity of the matching does not in and of itself lead to an efficient algorithm because the period length is not, in general, polynomial in the problem size. Nevertheless, since we may place an a priori finite upper bound on the period length, and since the dynamic matching problem reduces to an ordinary matching problem if the period length is fixed, we may solve the dynamic matching problem as a finite sequence of matching problems.

As is well known, Edmonds [1965a and b] was the first to provide a polynomial time algorithm for the non-bipartite matching problem. Since that time, many authors including Cunningham and Marsh [1978], Even and Kariv [1975], Gabow [1976] and Lawler [1976] have provided increasingly more efficient algorithms for various types of matching algorithms.

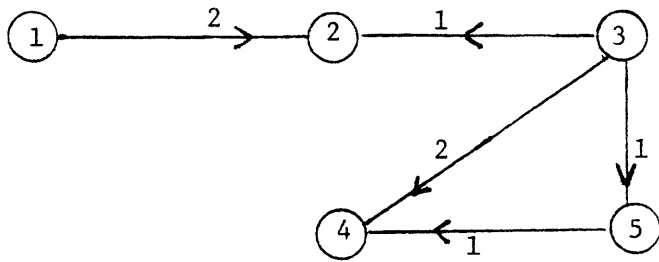


Figure 2.1a A 1-period digraph with associated edge distances.

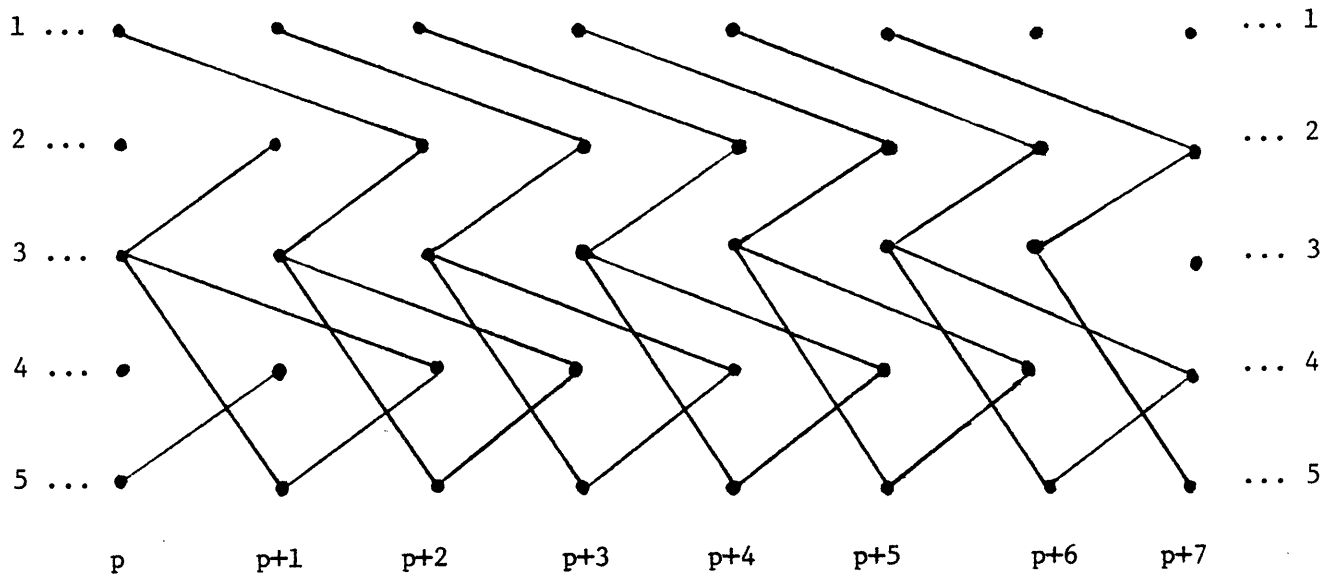


Figure 2.1b The dynamic graph induced by the digraph of Figure 2.1a.

3. Quasi-Dynamic Fractional Matchings

Our method for solving the dynamic matching problem is as follows. First we define the quasi-dynamic fractional matching problem, a finite graph theoretic problem that is closely related to the dynamic matching problem. We then show that a quasi-dynamic fractional matching of weight W "induces" a dynamic matching whose average weight per period is $W/2$. Finally, we use the quasi-dynamic fractional matching duality theory proved in Part II of this paper in order to prove that a maximum weight quasi-dynamic fractional matching induces a dynamic matching of maximum average weight per period.

Graph Theoretical Preliminaries

Below we introduce some graph theoretic definitions and notation most of which is common to the literature although not universal. A path in a directed graph G is a sequence $P = v_0, e_1, v_1, \dots, e_k, v_k$ such that $V(P) = \{v_0, \dots, v_k\}$ is a subset of $V(G)$ (although we do not assume that the vertices of P are distinct); $E(P) = \{e_1, \dots, e_k\}$ is a subset of $E(G)$; for $1 \leq j \leq k$, either v_{j-1} is the tail of e_j and v_j is the head of e_j or else v_j is the tail and v_{j-1} is the head; in the former case e_j is called a forward edge of p ; in the latter case e_j is called a backward edge. The path P is simple if $|V(P)| = k+1$. The weight of a path P is the sum of the weights of the edges of $E(P)$, and we denote it as $w(P)$. The distance of a path P is the sum of the distances of the forward edges of P minus the sum of the distances of the backward edges of P and we denote the distance of P as $d(P)$. The path P is a circuit if $|V(P)| = k$, $v_0 = v_k$, and $k \geq 1$. A circuit C is called trivial if it is a 2-edge circuit in which the same edge appears twice. A circuit C is called neutral if $d(C) = 0$, and it is called aneutral if $d(C) \neq 0$. Figures 3.1 and 3.2 display a neutral and an aneutral circuit respectively.

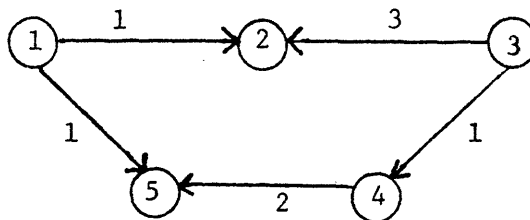


Figure 3.1 A neutral circuit.

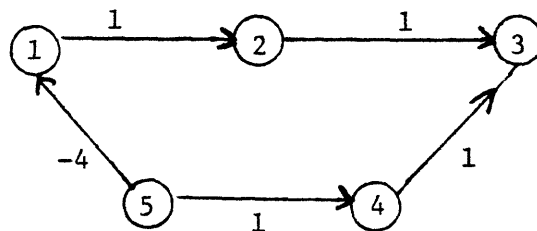


Figure 3.2 An aneutral circuit. Its distance is +4 or -4 depending on its orientation.

For each set $S \subseteq V(G)$, we let $\delta_G(S)$ denote the set of edges having exactly one endpoint in S , and we let $\gamma_G(S)$ denote the set of edges having both endpoints in S . (Henceforth, in cases in which there is no ambiguity we will drop the subscript G from δ and γ). We will let $\delta(v)$ (resp., $\gamma(v)$) denote the set $\delta(\{v\})$ (resp., $\gamma(\{v\})$). Thus $\gamma(v)$ is the set of loops in $E(G)$ both of whose endpoints are vertex v .

A fractional matching is a collection \mathcal{C} of vertex disjoint circuits such that each circuit in \mathcal{C} is either trivial (2 edges) or else is odd. The weight of a fractional matching is the sum of the weights of the circuits in \mathcal{C} . (Fractional matchings are often referred to as 2-matchings, for example in Cornuejols and Pulleyblank [1980].)

A quasi-dynamic fractional matching is a fractional matching with the additional property that all of the odd circuits in the collection are aneutral. (Of course, all of the trivial circuits are by definition neutral.) Henceforth we will refer to a quasi-dynamic fractional matching by the abbreviated name Q-matching. We will often denote a Q-matching as an ordered pair (M, Q) where M is the subset of edges of $E(G)$ corresponding to the trivial circuits of the Q-matching and Q is a collection of odd aneutral circuits. A Q-matching (M, Q) is portrayed in Figure 3.3. The set $M = \{(6,4)\}$ and Q contains the two odd aneutral circuits in boldface.

A subgraph H of G is said to be rooted if there is a designated vertex v_H of H that is the root. A subgraph H of G is said to be odd (resp., even) if $|V(H)|$ is odd (resp., even).

The close relation of Q-matchings to dynamic matchings is a consequence of the following lemma proved by the author (1981a).

Lemma 1. Let $G = (V(G), E(G))$ be a finite directed graph with edge distances $d = (d(e))$, and let $G^\infty = (V(G^\infty), E(G^\infty))$ be the induced dynamic

graph. If C is a cycle in G of distance $d(C) = s \neq 0$, then the infinite number of copies of the edges of C comprise $|s|$ vertex disjoint infinite paths in G^∞ .

The above Lemma is illustrated in Figures 3.4, the dynamic graph induced by the subgraph $G[\{1,2,3,5\}]$ of G as restricted to vertices 1,2,3 and 5.

Corollary 1. Let G be a finite directed graph, and let G^∞ be the induced dynamic graph. Then any Q -matching (M,Q) of G induces a dynamic matching M^* of G^∞ consisting of all copies of edges e for which $e \in E(M)$ together with every other edge of the vertex disjoint infinite paths induced by the odd circuits of Q . Moreover, the weight of (M,Q) is twice the average weight per period of M^* .

Proof. The proof is direct and is left to the reader.

Corollary 1 is illustrated in Figure 3.5 and is the dynamic matching induced by the Q -matching of Figure 3.3.

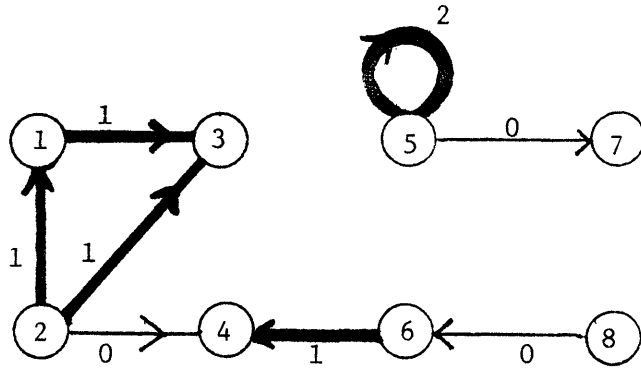


Figure 3.3. A Q-matching with one matched edge and two matched odd neutral circuits.

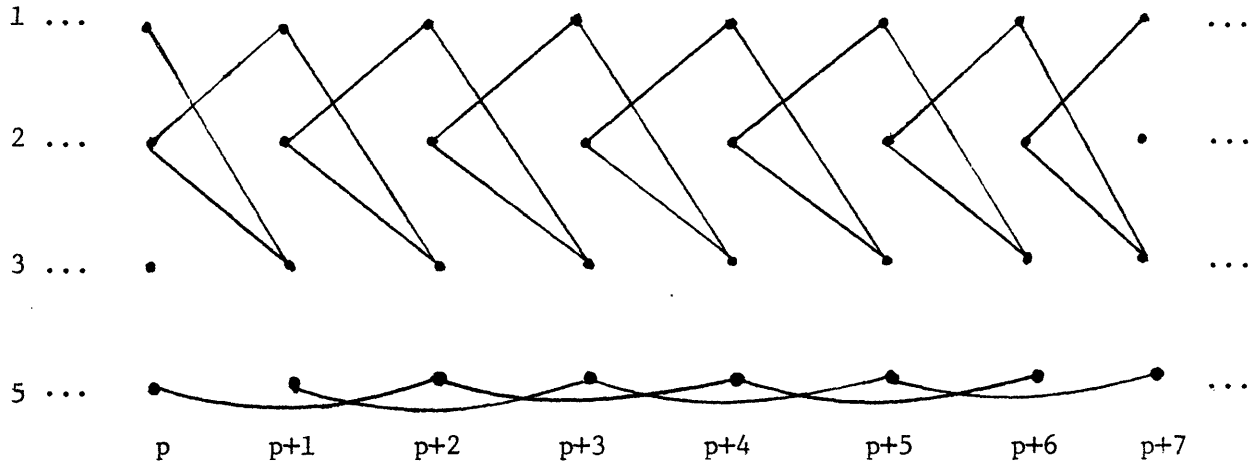


Figure 3.4. The three infinite length paths in G^∞ induced by the two odd circuits in the Q-matching of G in Figure 3.3.

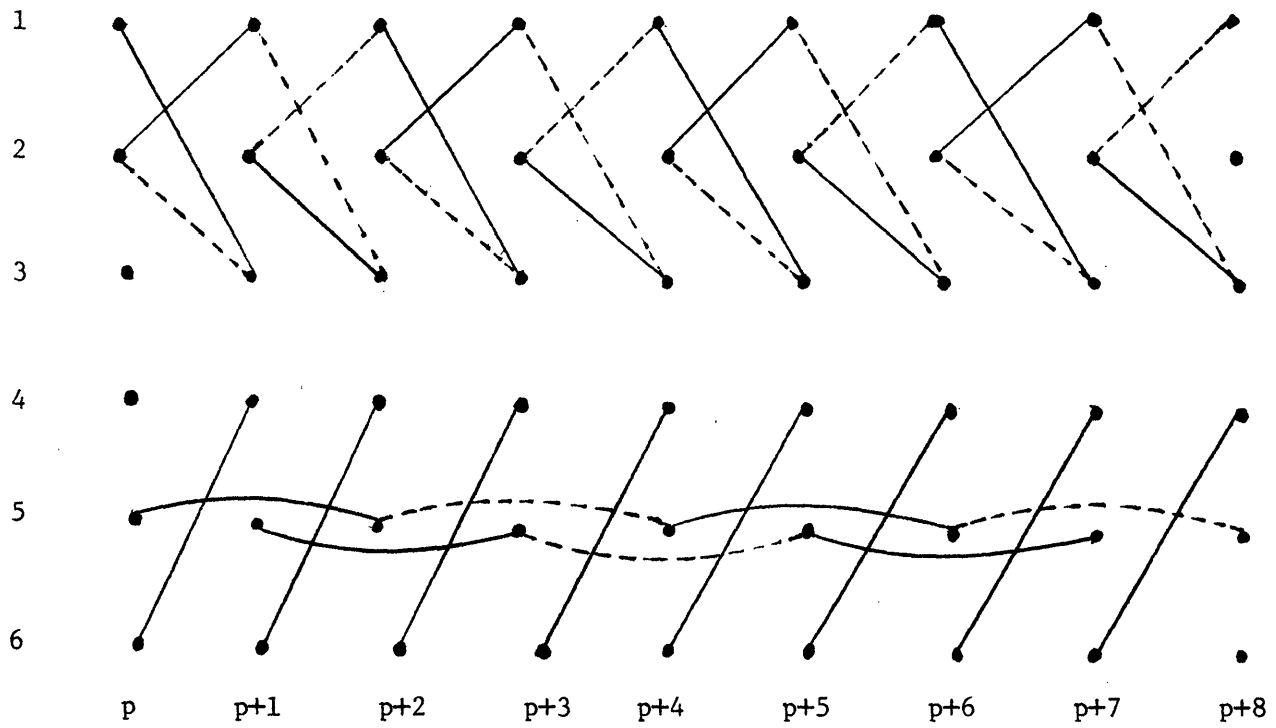


Figure 3.5. A dynamic matching induced by the Q-matching of Figure 3.3. (The copies of vertices 7 and 8 are not portrayed since they are unmatched in the Q-matching).

4. Maximum Weight Q-Matchings Induce Average Weight Optimal Dynamic Matchings

We have just seen that a maximum weight Q-matching (M,Q) induces a dynamic matching whose average weight per period is one-half of the weight of (M,Q). In this section we show that the induced dynamic matching is indeed average weight optimal. The proof technique relies on some elementary properties of dynamic integer programs described below.

Dynamic Integer Programs

The dynamic integer programming problem consists of all instances of (4.1)

$$\text{Maximize } \liminf_{t \rightarrow \infty} (2t+1)^{-1} \sum_{p=-t}^t wx^p \quad (4.1a)$$

$$\text{Subject to } A^0 x^p + \dots + A^q x^{p+q} \leq b \quad \text{for all } p \in Z \quad (4.1b)$$

$$x^p \geq 0 \text{ integer} \quad \text{for all } p \in Z \quad (4.1c)$$

where A^0, \dots, A^q are $m \times n$ integer valued matrices, w and b are vectors, and x^p is the vector of decision variables for period p .

We note that the dynamic matching problem is the special case of the dynamic integer programming problem in which b is a vector of all 1's and each column of the 2-way infinite constraint matrix is a 0-1 vector with exactly two 1's.

The dynamic linear programming problem is the problem obtained from the dynamic integer programming problem by relaxing the integrality constraints. A sequence $x = (x^p)$ of decision vectors is said to be stationary if $x^p = x^{p+1}$ for all $p \in Z$. The author [1981a] proved that if there is an optimum average-weight sequence $x = (x^p)$ for a dynamic linear programming problem, then there is such a sequence that is stationary. Here we state and prove a weaker result (which is strong enough for our purposes).

Lemma 2. Let $x = (x^P)$ be an optimal periodic solution for a dynamic integer programming problem. Then there is a stationary continuous solution y that is feasible for the continuous relaxation and whose average weight per period is at least as great as that of x .

Proof. Suppose x is periodic with period t . Let $y^P = t^{-1}(x^1 + \dots + x^t)$ for all $p \in Z$. It is easily verified that y is a stationary solution that is feasible for the continuous relaxation. Moreover, the average weight per period of y is equal to the average weight per period of x .

Neutral Subgraph

In the case of finite matching problems, Edmonds [1965b] defined a polyhedral set of constraints that he showed to be the convex hull of the matching polyhedron. Unfortunately, the convex hull of the dynamic matching polyhedron cannot be expressed as a dynamic linear program. (The set of facets contains the odd set constraints defined by Edmonds [1965b], and there are a countably infinite number of finite odd sets containing any specified edge of $E(G^\infty)$. Thus any dynamic linear programming formulation would involve infinite matrices A^0, \dots, A^q , which is forbidden.) Nevertheless, we can solve the dynamic matching problem through a minor modification of Edmonds technique. It suffices to add a subset of polyhedral facets to the dynamic integer programming formulation of the dynamic matching problem with the property that the resulting linear relaxation is "optimally solved by a Q-matching." Then by Lemmas 1 and 2, the induced dynamic matching is average weight optimal. The subset of polyhedral facets that we consider may be referred to as "neutral set constraints", as defined below.

Let H be a subgraph of directed graph G . We say that H is neutral if H contains no aneutral circuit. The following characterization of connected neutral subgraphs is elementary and is stated without proof.

Lemma 3. A connected subgraph H of the digraph G is neutral if and only if for every two vertices $u, v \in V(H)$ it follows that the distance of a path from u to v in H is independent of the path.

Corollary 2. Let H be a rooted connected neutral subgraph of the digraph G . For each vertex $v \in V(H)$, let $d_H(v)$ be the distance of a path in H from the root to v . Then for each edge $e = (u, v) \in E(H)$ it follows that $d(e) = d(v) - d(u)$. \square

Let H be a rooted neutral connected subgraph of the digraph G and let $d(v) = d_H(v)$ be the distance in H from the root to vertex v . Then the p^{th} copy of H is the unique subgraph H^p of G^∞ such that

$$(1) \quad V(H^p) = \{v^{p+d(v)} : v \in V(H)\},$$

$$(2) \quad E(H^p) = \{\text{all copies of } E(H) \text{ with both endpoints in } V(H^p)\}.$$

By Lemma 3 and its Corollary, the p^{th} copy of H is well defined and has exactly one copy of each vertex and edge of H . (Note that this latter property would not be true if H has an aneutral circuit.) Therefore, the following result is immediate.

Corollary 3. Let H be an odd rooted connected neutral subgraph of the directed graph G such that $|V(H)| = 2k+1$. Suppose M is a dynamic matching of the dynamic graph G^∞ induced from G . Then for each integer p , the number of edges of $E(H^p)$ in M is at most k .

The Dynamic Matching Problem with The Neutral Subgraph Facets

We present below a dynamic integer programming formulation of the dynamic matching problem in which we incorporate the implied constraints (or cuts) described in the previous subsection.

Let $G = (V(G), E(G))$ be a directed graph, and suppose that $E = \{e_1, \dots, e_m\}$. Let \mathcal{N} be the set of all of the odd connected neutral subgraphs of G (a set bounded in cardinality by the number of subsets of edges of G). Moreover, let us assume that each neutral subgraph H in \mathcal{N} is rooted (for example, at the lowest index vertex of $V(H)$).

We let $x_j^p = 1$ or 0 accordingly as edge e_j^p of $V(G^\infty)$ is or is not in the matching. We let $w_j = w(e_j)$. We then formulate the dynamic matching problem as the dynamic integer program (5.1).

$$\text{Maximize } \liminf_{t \rightarrow \infty} (2t+1)^{-1} \sum_{p=-t}^t \sum_{e_j \in E(G)} w_j x_j^p \quad (5.1a)$$

$$\text{Subject to } \sum_{e_j^p \in \delta(v^r)} x_j^p \leq 1 \quad \text{for all } v \in V(G) \quad (5.1b)$$

and all $r \in \mathbb{Z}$

$$\sum_{e_j^p \in E(H^r)} x_j^p \leq (1/2)(|V(H)|-1) \quad \text{for all } H \in \mathcal{N} \quad (5.1c)$$

and all $r \in \mathbb{Z}$

$$x_j^r \geq 0 \text{ integer} \quad \text{for all } e_j \in E(G) \quad (5.1d)$$

and all $r \in \mathbb{Z}$

Theorem 1. Let G be a directed graph and let G^∞ be the induced dynamic graph. Then a maximum weight Q -matching for G induces a dynamic matching for G^∞ that is average weight maximum.

Proof. Let (M, Q) be a maximum weight Q -matching of G . By Corollary 1, (M, Q) induces a dynamic matching M' of G^∞ whose average weight W' per period is half of the weight of (M, Q) . Thus it suffices to show that there cannot be a dynamic matching whose average weight per period exceeds W' .

By Lemma 2, it suffices to show that the maximum weight per period of a stationary continuous-valued solution to (5.1) does not exceed W' . We can determine such a maximum average weight stationary solution $y = (y_i^p)$ by observing first that the value of y_1^p is independent of the superscript p ,

and then substituting into the dynamic integer program (5.1) to obtain the linear program (5.2).

$$\text{Maximize } \sum_{e_j \in E(G)} w_j y_j \quad (5.2a)$$

$$\text{Subject to } \sum_{e_j \in \delta(v)} y_j \leq 1 \quad \text{for all } v \in V(G) \quad (5.2b)$$

$$\sum_{e_j \in E(H)} y_j \leq (1/2)(|V(H)|-1) \quad \text{for all } H \in \mathcal{N} \quad (5.2c)$$

$$y_j \geq 0 \quad \text{for all } e_j \in E(G) \quad (5.2d)$$

The stationary solution y^∞ can be obtained by repeating the solution y in each period, and thus the average weight per period of y^∞ is the same as the weight of wy .

We observe that any Q -matching (M, Q) with weight W^* induces a solution $y = (y_e)$ for (5.2) with $wy = (1/2)W^*$ given by

$$y_j = \begin{cases} 1 & \text{if } e_j \text{ is an edge of } M \\ 1/2 & \text{if } e_j \text{ is an edge of } Q \\ 0 & \text{otherwise .} \end{cases}$$

The proof will be complete if we demonstrate that (5.2) is optimized at a Q -matching. This final result is Theorem 3 in Part II of this paper, and an algorithmic proof is provided there. \square

We note that the time to solve the dynamic matching problem is thus equal to the time to solve the Q -matching problem. In Part II of this paper we provide an $O(|V(G)|^3)$ time algorithm to solve both problems.

6. Remarks on The Methodology

The above solution technique for dynamic matchings is an example of an integer program solved via rounding. The dynamic integer programming problem is equivalent to (5.1). The solution technique may be viewed as rounding an optimal continuous stationary solution. (Each edge e_j in a circuit of Q is such that $x_j^p = 1/2$ for all p . Half of these variables are rounded up whereas the other half are rounded down.)

The idea of rounding may seem incidental to the solution technique because in a 0-1 integer program any change in a fractional variable may be viewed as rounding. However, the structure of dynamic integer programs suggests that the rounding depends on the basis structure of the linear relaxation (5.2). Moreover, the rounding technique here is quite similar to the one developed by the author [1981a] for the dynamic network flow problem. We conjecture that a similar rounding technique will be successful if we consider b -dynamic matchings, i.e., if we allow the right-hand side of the dynamic matching integer programming problem to be any positive integer vector b .

7. Applications

A number of applications of matchings may be extended quite directly to dynamic matchings if the scheduling process is both dynamic and periodic. Below we provide one such application. Although the example is somewhat artificial, we hope that it illustrates the modeling of the dynamic/periodic aspect of scheduling.

An Application to Dynamic Pairings of Workers

Consider a firm that wishes to pair its workers on projects on each day over an infinite horizon. The projects may be split so that worker i can complete half of any project and the remaining half may be completed s days later by worker j at a value w to the firm. In this case we have vertices i^p representing worker i on day p , and there is an edge joining i^p to j^{p+s} with weight w . The objective is to pair workers so as to maximize the per-day value to the firm.

A Potential Application to Dynamic Integer Programming

Not only is dynamic integer programming P-space hard, but most special cases of it are also P-space hard, as proved by the author [1981b]. For instance, the special case of determining whether there is a feasible solution to (7.1) is P-space hard.

$$\begin{aligned}
Ax^i + Bx^{i+1} &= 1 && \text{for } i \in Z \\
0 \leq x^i \leq 1; x^i &\text{ integer} && \text{for } i \in Z
\end{aligned}
\tag{7.1}$$

where A and B are $m \times n$ matrices of 0's and 1's such that each column of $\begin{bmatrix} B \\ A \end{bmatrix}$ has at most three 1's.

As an example of a dynamic integer programming set cover problem, consider the following periodic single processor scheduling problem which was formulated and communicated to the author by C. L. Liu [1979]. Let

$\{J_1, \dots, J_n\}$ be a set of n jobs each of which has a unit processing time and must be scheduled "periodically". The objective is to determine an infinite horizon schedule such that exactly one job is processed in every period and such that the maximum number of periods between successive proce of job J_i is at most t_i .

We can formulate the periodic processor scheduling problem as a dynamic integer program as follows:

$$\text{Let } x_i^p = \begin{cases} 1 & \text{if job } J_i \text{ is processed in period } p \\ 0 & \text{if otherwise} \end{cases}$$

Then determine a sequence $x = (x_i^p)$ such that

$$\sum_{i=1}^n x_i^p = 1 \quad \text{for } p = 1, 2, 3, \dots$$

$$\sum_{j=p}^{p+t_i-1} x_i^j \geq 1 \quad \text{for } p = 1, 2, 3, \dots \text{ and } i = 1, \dots, n$$

$$x_i^p \in \{0, 1\} \quad \text{for all } i, p.$$

A promising application of dynamic matchings may be as subroutines in heuristics for solving dynamic combinatorial optimization problems such as the above processor scheduling problem. In particular, Nemhauser and Weber [1975] showed how to solve the covering problem using a Lagrangean relaxation in which each subproblem is a matching problem. It is easy to extend their technique so as to transform dynamic set covering problems into dynamic matching problems with side conditions. It is an interesting open question as to whether the resulting Lagrangean relaxation can be exploited so as to provide a more efficient solution technique for the covering problem.

ACKNOWLEDGEMENTS

I wish to thank Professor Arthur F. Veinott for his help during the initial stages of this research. I also wish to thank John VandeVate for his careful reading of the manuscript and for his suggestions that led to improvements in the exposition.

References

- Cornuejols, G. and W. Pulleyblank. 1980. A Matching Problem with Side Constraints. Discrete Mathematics 29, 135-159.
- Cunningham, W. and A. Marsh III. 1978. A Primal Algorithm for Optimum Matching. Mathematical Programming Study 8, 50-72.
- Edmonds, J. 1965a. Paths Trees and Flowers. Canadian Journal of Mathematics 17, 449-467.
- Edmonds, J. 1965b. Maximum Matching and a Polyhedron of (0,1) Vertices. Journal of Research of the National Bureau of Standards. 6913, 125-130.
- Even, S. and Q. Kariv. 1975. An $O(n^{5/2})$ Algorithm for Maximum Matching in General Graphs. In 16th Annual Symposium on the Found. of Comp. Science, IEEE, New York, 100-112.
- Ford, L. R. and D. R. Fulkerson. 1958. Constructing Maximal Dynamic Flows from Static Flows. Operations Research 6, 419-433.
- Gabow, H. 1976. An Efficient Implementation of Edmonds' Algorithm for Maximum Matchings on Graphs. Journal of the Association of Computing Machinery, 23, 221-234.
- Gale, D. 1959. Transient Flows in Networks. Michigan Math Journal 6, 59-63.
- Garey, M. R. and D. S. Johnson. 1979. Computers and Intractability: A Guide to The Theory of NP-Completeness. W. H. Freeman and Company, San Francisco.
- Halpern, J. 1979. A Generalized Dynamic Flow Problem. Networks 9, 133-167.
- Jarvis, J. and H. D. Ratliff. 1981. Some Equivalent Objectives for Dynamic Network Flow Problems. Management Science 28, 106-109.
- Lawler, E. L. 1976. Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston, New York.
- Minieka, E. 1973. Maximal Lexicographic and Dynamic Network Flows. Operations Research 21, 517-527.
- Nemhauser, G. and G. Weber. 1979. Optimal Set Partitioning, Matching, and Lagrangian Duality. Naval Research Logistics Quarterly 26, 553-563.
- Orlin, J. 1981a. Dynamic Network Flows. Ph.D. dissertation. Department of Operations Research, Stanford University, Stanford, CA.
- Orlin, J. 1981b. The Complexity of Dynamic Languages and Dynamic Optimization Problems. 13th Annual Symposium on the Theory of Computing, 218-227.

Orlin, J. 1982. Dynamic Matchings and Quasi-Dynamic Fractional Matchings II.

White, W. 1972. Dynamic Transshipment Network: An Algorithm and its Application to The Distribution of Empty Containers. Networks 2, 211-236.

Wilkenson, W. 1971. An Algorighm for Universal Dynamic Flows in A Network. Operations Research 19, 1602-1612.

Wilkenson, W. 1973. Min/Max Bounds for Dynamic Network Flows. Naval Research Logistics Quarterly 20, 505-516.