A $n^{5/2}$ ALGORITHM FOR MAXIMUM MATCHINGS IN BIPARTITE GRAPHS

John E. Hopcroft
Cornell University
Ithaca, N.Y.

Richard M. Karp
University of California
Berkeley, California

Abstract

The present paper shows how to construct a maximum matching in a bipartite graph with n vertices and m edges in a number of computation steps proportional to $(m+n)\ n$.

1. Introduction

Suppose we are given a rectangular array in which each cell is designated as "occupied" or "unoccupied". A set of cells is <u>independent</u> if no two of the cells lie in the same row or column. Our object is to construct an independent set of occupied cells having maximum cardinality.

In one interpretation, the rows of the array represent boys, and the columns, girls. Cell i,j is occupied if boy i and girl j are compatible, and we wish to match a maximum number of compatible couples.

An alternative statement of the problem is obtained by representing the rows and columns of the array as the vertices of a bipartite graph. The vertices corresponding to row i and column j are joined by an edge if and only if cell i,j is occupied. We then seek a maximum matching; i.e., a maximum number of edges, no two of which meet at a common vertex.

This problem has a wide variety of applications [3,4,5]. These include the determination of chain decompositions in partially ordered sets, of coset representatives in groups, of systems of distinct representatives, and of block-triangular decompositions of sparse matrices. The problem also occurs as a subroutine in the solution of the Hitchcock transportation problem, and in the determination of whether one given tree is isomorphic to a subtree of another.

In view of this variety of applications, the computational complexity of the problem of finding a maximum matching in a bipartite graph is of interest. The best previous methods ([1,3,4,5]) seem to require $O(mn)$ steps, where m is the number of edges, and n the number of vertices. The present method requires only $O((m+n)\sqrt{n})$ steps.

We hope to extend our results to the non-bipartite case (cf.[2]). With this in mind, all the results in Section 2 are derived for general graphs. The specialization to the bipartite case occurs in Section 3.

* This research was supported in part by National Science Foundation grants NSF GJ96 and GP-25081.

2. Matchings and Augmenting Paths

Let $G = (V,E)$ be a finite undirected graph (without loops or multiple edges) having the vertex set V and the edge set E. An edge incident with vertices v and w is written $\{v,w\}$. A set $M \subseteq E$ is a <u>matching</u> if no vertex $v \in V$ is incident with more than one edge in M. A matching of maximum cardinality is called a <u>maximum matching</u>.

We make the following definitions relative to a matching M. A vertex v is <u>free</u> if it is incident with no edge in M.

A path (without repeated vertices)

$$P = (v_1,v_2)(v_2,v_3),\ldots,(v_{2k-1},v_{2k})$$

is called an <u>augmenting path</u> if its end points v_1 and v_{2k} are both free, and its edges are alternately in $E-M$ and in M; i.e.,

$$P \cap M = \{(v_2,v_3),(v_4,v_5),(v_6,v_7),\ldots(v_{2k-2},v_{2k-1}$$

When no ambiguity is possible we let P denote the set of edges in an augmenting path P, as well as the sequence of edges which is the path itself. If S and T are sets then $S \oplus T$ denotes the symmetric difference of S and T, and $S-T$ denotes the set of elements in S which are not in T. If S is a finite set then $|S|$ denotes the cardinality of S.

<u>Lemma</u> 1 If M is a matching and P is an augmenting path relative to M, then $M \oplus P$ is a matching, and $|M \oplus P| = |M|+1$.

<u>Theorem 1</u> Let M and N be matchings. If $|M| = r$, $|N| = s$ and $s > r$, then $M \oplus N$ contains at least $s-r$ vertex-disjoint augmenting paths relative to M.

<u>Proof</u> Consider the graph $\bar{G} = (V, M \oplus N)$ with vertex set V and edge set $M \oplus N$. Since M and N are matchings, each vertex is incident with at most one edge from $N-M$ and at most one edge from $M-N$; hence each (connected) component of $\bar{G}$ is either:

i) an isolated vertex,
ii) a cycle of even length, with edges alternately in $M-N$ and in $N-M$,

or

iii) a path whose edges are alternately in $M-N$ and in $N-M$.

Let the components of $\bar{G}$ be $C_1, C_2, \ldots, C_g$, where $C_i = (V_i, E_i)$

Let $\delta(C_i) = |E_i \cap N| - |E_i \cap M|$

Then $\delta(C_i) \in \{-1,0,1\}$, and $\delta(C_i) = 1$ if and only if C_i is an augmenting path relative to M.

$$\sum_i \delta(C_i) = |N-M| - |M-N| = |N| - |M| = s-r.$$

Hence there are at least $s-r$ components C_i of $\bar{G}$ such that $\delta(C_i)=1$. These components are vertex-disjoint, and each is an augmenting path relative to M.

Corollary 1 (Berge) M is a maximum matching if and only if there is no augmenting path relative to M.

Corollary 2 Let M be a matching. Suppose $|M| = r$, and suppose that the cardinality of a maximum matching is s, $s>r$. Then there exists an augmenting path relative to M of length $\leq 2\left\lfloor \frac{r}{s-r} \right\rfloor + 1$.[+]

Proof Let N be a maximum matching. Then $M \oplus N$ contains $s-r$ vertex-disjoint (and hence edge-disjoint) augmenting paths relative to M. Altogether these contain at most r edges from M, so one of them must contain at most $\left\lfloor \frac{r}{s-r} \right\rfloor$ edges from M, and hence at most $2\left\lfloor \frac{r}{s-r} \right\rfloor + 1$ edges altogether.

Let M be a matching. The augmenting path P is called shortest relative to M if P is of least cardinality among augmenting paths relative to M.

Theorem 2 Let M be a matching, P a shortest augmenting path relative to M, and P' an augmenting path relative to $M+P$. Then

$$|P'| \geq |P| + |P \cap P'|.$$

Proof Let $N = M \oplus P \oplus P'$. Then N is a matching and $|N| = |M| + 2$, so $M \oplus N$ contains two vertex-disjoint augmenting paths relative to M; call them P_1 and P_2.

Since $M \oplus N = P \oplus P'$

$$|P \oplus P'| \geq |P_1| + |P_2| .$$

But $|P_1| \geq |P|$ and $|P_2| \geq |P|$, since P is a shortest augmenting path. So

$$|P \oplus P'| \geq |P_1| + |P_2| \geq 2|P|.$$

and also we have the identity

$$|P \oplus P'| = |P| + |P'| - |P \cap P'|.$$

Hence $|P'| \geq |P| + |P \cap P'|$.

We envisage the following scheme of computation: Starting with a matching M_0, compute a sequence $M_0, M_1, M_2, \ldots, M_i, \ldots$ where P_i is a shortest augmenting path relative to M_i, and $M_{i+1} = M_i \oplus P_i$.

Corollary 3 $|P_i| \leq |P_{i+1}|$

Corollary 4 For all i and j such that $|P_i| = |P_j|$, P_i and P_j are vertex-disjoint.

Proof Suppose for contradiction that $|P_i| = |P_j|$, $i<j$, and P_i and P_j are not vertex-disjoint. Then there exist k and ℓ such that $i \leq k < \ell \leq j$, P_k and P_ℓ are not vertex-disjoint and for each m, $k<m<\ell$, P_m is vertex-disjoint from P_k and P_ℓ. Then P_ℓ is an augmenting path relative to M_k+P_k , so

$$|P_\ell| \geq |P_k| + |P_k \cap P_\ell|.$$

But $|P_\ell| = |P_k|$, so $|P_k \cap P_\ell| = 0$. Thus P_k and P_ℓ have no edges in common. But if P_k and P_ℓ had a vertex v in common they would have in common that edge incident with v which is in $M_k \oplus P_k$. Hence P_k and P_ℓ are vertex-disjoint, and a contradiction is obtained.

Theorem 3 Let s be the cardinality of a maximum matching. The number of distinct integers in the sequence

$$|P_0|, |P_1|, \ldots, |P_i|, \ldots$$

is less than on equal to $2\lfloor \sqrt{s} \rfloor + 2$.

Proof Let $r = \lfloor s - \sqrt{s} \rfloor$ then $|M_r| = r$ and, by Corollary 2,

$$|P_r| \leq 2 \frac{\lfloor s-\sqrt{s} \rfloor}{s - \lfloor s - \sqrt{s} \rfloor} + 1 \leq 2\lfloor \sqrt{s} \rfloor + 1.$$

Thus, for each $i<r$, $|P_i|$ is one of the $\lfloor \sqrt{s} \rfloor + 1$ positive odd integers less then or equal to $2\lfloor \sqrt{s} \rfloor + 1$. Also $|P_{r+1}|, \ldots, |P_s|$ contribute at most $s-r = \lceil \sqrt{s} \rceil$ distinct integers, and the total number of distinct integers is less than or equal to

$$\lfloor \sqrt{s} \rfloor + 1 + \lceil \sqrt{s} \rceil \leq 2 \lfloor \sqrt{s} \rfloor + 2.$$

+ The symbol $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x, and $\lceil x \rceil$ denotes the least integer greater than or equal to x.

In view of Corollaries 3 and 4 and Theorem 3, the computation of the sequence $\{M_i\}$ breaks into at most $2\lfloor\sqrt{s}\rfloor + 2$ phases, within each of which all the augmenting paths found are vertex-disjoint and of the same length. Since these paths are vertex-disjoint, they are all augmenting paths relative to the matching with which the phase is begun. This gives us an alternative way of describing the computation of a maximum matching.

<u>Algorithm A</u> <u>Maximum Matching Algorithm</u>

0. $M \leftarrow Q$

1. Let $\ell(M)$ be the length of a shortest augmenting path relative to M. Find a maximal set of paths $\{Q_1, Q_2, \ldots, Q_t\}$ with the properties that:

   a) for each i, Q_i is an augmenting path relative to M and $|Q_i| = \ell(M)$;

   b) the Q_i are vertex-disjoint.

2. $M \leftarrow M \oplus Q_1 \oplus Q_2 \oplus \ldots \oplus Q_t$; go to 1.

<u>Corollary 5</u> If the cardinality of a maximum matching is s then Algorithm A constructs a maximum matching within $\lfloor 2\sqrt{s}\rfloor + 2$ executions of Step 1.

This way of describing the construction of a maximum matching suggests that we should not regard successive augmentation steps as independent computations, but should instead concentrate on the efficient implementation of an entire phase (i.e., the execution of Step 1 in Algorithm A). The next section shows the advantage of this approach in the case where G is a bipartite graph.

3. The Bipartite Case

The graph $G = (V,E)$ is <u>bipartite</u> if the set of vertices V can be partitioned into two sets, X and Y, such that each edge of G joins a vertex in X with a vertex in Y. An element of X will be called a <u>boy</u>, and an element of Y, a <u>girl</u>.

Let M be a matching in a bipartite graph G. We discuss the implementation of Step 1 of Algorithm A, in which a maximal vertex-disjoint set of shortest augmenting paths relative to M is found.

We construct a directed graph $\hat{G} = (\hat{V},\hat{E})$ which exhibits conveniently all shortest augmenting paths relative to M. Define:

a) $L_0 = \{x \in X \mid x \text{ is free relative to } M\}$

b) for $i = 0, 1, 2, \ldots$

$$E_{2i} = \{[y,x] \mid x \in L_{2i},\ y \notin L_1 \cup L_3 \cup \ldots \cup L_{2i-1},\ \text{and } \{x,y\} \in E - M\}$$

$$L_{2i+1} = \{y \mid \text{for some } x,\ [y,x] \in E_{2i}\}$$

c) for $i = 1, 2, 3, \ldots$.

$$E_{2i-1} = \{[x,y] \mid y \in L_{2i-1} \text{ and } \{x,y\} \in M\}$$

$$L_{2i} = \{x \mid \text{for some } y,\ [x,y] \in E_{2i-1}\}.$$

The construction continues until for the first time a set L_{2i^*+1} is constructed which includes a free girl. Then

$$\hat{V} = \bigcup_{j=0}^{2i^*+1} L_j \quad \text{and} \quad \hat{E} = \bigcup_{j=0}^{2i^*} E_j$$

The following properties of $\hat{G}$ are immediate:

i) The vertices at even levels are boys, and those at odd levels are girls;

ii) If $[u,v]$ is an edge of $\hat{G}$, then v is at the level preceding the level of u;

iii) Each boy in $L_{2i}, i>0$ has out-degree 1, and each girl has out-degree ≥ 1.

iv) The shortest augmenting paths relative to M are in one-to-one correspondence with the paths of $\hat{G}$ which begin at a free girl and end at a free boy.

It is convenient to associate with each vertex u the sets

$$PRED(u) = \{v \mid [v,u] \in \hat{E}\}$$

and $SUCC(u) = \{v \mid [u,v] \in \hat{E}\}$. Also, let $\#(u) = |SUCC(u)|$. By iii), $\#(u) > 0$ if $u \notin L_0$.

Given $\hat{G}$ it is easy to construct the sequence of vertices of a shortest augmenting path relative to M. Simply choose any free girl y_0 in L_{2i^*+1}, and construct a sequence $u_1, u_2, \ldots, u_{2i^*+1}$, where $y_0 = u_0$ and u_{i+1} is the first element in $SUCC(u_i)$. No back-tracking is necessary since, for each vertex $u \notin L_0$, $SUCC(u) \neq \phi$.

As successive shortest augmenting paths are found it is necessary to delete their vertices, since all the paths found must be vertex-disjoint. The deletion of such vertices may render certain other vertices useless, in the sense that they cannot reach any vertex in L_0. These vertices must also be deleted; for, if they were kept, the search for an augmenting path would be complicated by the possibility of dead-ends. Hence we need an algorithm which, given $\hat{G}$ and a path P, deletes the vertices of P, together with all other vertices rendered useless by this deletion. This is achieved as follows:

<u>Deletion Algorithm</u>

0) $U \leftarrow$ the set of vertices of P.
   $T \leftarrow U$

1) If $U = \phi$, stop. Else choose any $u \in U$.
   $U \leftarrow U - \{u\}$

   For each $v \in PRED(u) - T$

      $SUCC(v) \leftarrow SUCC(v) - \{u\}$
      $\#(v) \leftarrow \#(v) - 1$
      If $\#(v) = 0$,
         $U \leftarrow U \cup \{v\}$
         $T \leftarrow T \cup \{v\}$

For each $v \in SUCC(u) - T$

$PRED(v) \leftarrow PRED(v) - \{u\}$

Go to 1.

The effect of the deletion algorithm is to remove from $\hat{G}$ those vertices which enter the set T, as well as the edges incident with those vertices. Let the resulting directed graph be denoted $\hat{G}\Delta P$.

Theorem 4 Let $P_1, P_2, \ldots, P_k$ be vertex-disjoint paths from a free girl to a free boy in $\hat{G}$. Let $G_0 = \hat{G}$ and $G_{i+1} = G_i \Delta P_i$. Then the paths from a free girl to a free boy in G_{k+1} are precisely the paths from a free girl to a free boy in $\hat{G}$ which are vertex-disjoint from all of $P_1, P_2, \ldots, P_k$.

Theorem 5 Every vertex of G_{k+1} which is not a free boy has out-degree greater than or equal to one.

The implementation of a phase consists of the construction of $\hat{G}$, followed by an iteration which alternately finds an augmenting path and executes the deletion algorithm. The iteration stops when all free girls have been deleted. It is easy to see that each edge of G is examined once when $\hat{G}$ is first constructed, and once more if an end point of that edge is deleted. With appropriate choice of data structures, an implementation is possible in which each edge inspection requires the execution of only a few machine language instructions. Indeed, such an implementation has been carried out, using Algol W.

The execution time of a phase is $O(m+n)$, where m is the number of edges in G, and n is the number of vertices. Hence the execution time of the entire algorithm is $O((m+n)\, s)$, where s is the cardinality of a maximum matching.

If G has n vertices then $m \leq \frac{n^2}{4}$ and $s \leq \frac{n}{2}$, so that the execution time is bounded by $O(n^{5/2})$.

References

1. Berge, C., "Two Theorems in Graph Theory", Proc. Nat. Acad. Sci. 43, pp. 842-844 (1957).

2. Edmonds, J., "Paths, Trees and Flowers", Candadian J. Math., XVII, pp. 449-467 (1965)

3. Ford, L.R. and D.R. Fulkerson, Flows in Networks, Princeton University Press (1965).

4. Hall, M., "Distinct Representatives of Subsets", Bull. Amer. Math. Soc. 54, pp. 922-926 (1948).

5. Kuhn, H.W., "The Hungarian Method for the Assignment Problem", Naval Res. Log. Quart. 2 , pp 83-97 (1955).