CHAPTER 33

Labelling to Obtain a Maximum Matching

M. L. BALINSKI[1], *The City University of New York*

1. INTRODUCTION

This paper describes a very simple labelling algorithm for solving the maximum matching problem on a graph. In contrast with the approach [4] which depends on " shrinking odd cycles " and, hence, on consideration of a hierarchy of reduced graphs this method permits all work to proceed on the given graph through use of purely " local " information stored only on the vertices of the graph. To contrast these approaches and enable the discussion to be self-sufficient Section 2 reviews basic theorems and the shrinking approach in a simple manner before proceeding to labelling. The success of labelling techniques in the closely related network flow problems together with the practical importance of integer programs defined on a graph [1], [2] has motivated this development which is now being extended to the general integer programming problem on a graph.

2. THE PROBLEM

A graph $G = \{V, E\}$ is taken to be a finite set of vertices

[1] This work was supported by the Army Research Office, Durham, under contract No. DA-31-124-ARO(D)-366.

V together with a set of distinct edges E which are unordered pairs of distinct vertices. A *matching* M of the graph is a subset of the edges E with the property that no two edges of M are incident at a vertex. The *maximum matching problem* is to find a matching having a maximum number of edges.

Given a graph G and a matching M an *alternating path* is a simple path (no vertices in common) whose successive edges alternately belong to and do not belong to M. An *augmenting path* is an alternating path connecting a pair of *exposed vertices,* vertices which are not incident to an edge of M. It is obvious that if a matching M admits an augmenting path, M cannot be maximum, since a simple reversal of assignment of edges in the augmenting path to M results in a new matching having one more edge than M. Not so obvious is the reverse statement which permits

Theorem 1. (Berge [3] for matching; Norman-Rabin [5] for covering). *A matching M is maximum if and only if G admits no augmenting paths relative to M.*

Proof. Suppose that M is a matching which admits no augmenting path but that $M^* \neq M$ is a maximum matching. Consider the subgraph G' of G containing all vertices of G and edges e of G which satisfy: $e \in M$ and $e \notin M^*$ or $e \notin M$ and $e \in M^*$. This is the set of edges where M and M^* differ.

Consider any connected component of the subgraph. Such a component can only be either a simple path or a cycle, for otherwise, the component would have a vertex with three incident edges. But this would mean that either at least two edges of M or at least two edges of M^* are incident, contradicting the fact that M and M^* are matchings. If the component is a simple path P we distinguish three cases: (i) the extreme edges of P are both in M and not in M^*; or (ii) both in M^* and not in M; or (iii) one is in M and not M^*, the other in M^* and not M.

In case (i) M^* admits an augmenting path in the component, contradicting its maximality. In case (ii) M admits an augmenting path, contradicting the hypothesis. In case (iii) the extreme vertices have no incident edges belonging to M or M^* other than those in the component itself. Therefore the component has an even number of edges and M^* may be changed by taking every edge $e \in M^*$ of the component out of M^*, and putting every edge $e \notin M^*$ of the component into M^*. This does not change the cardinality and hence gives a new M^* which is a maximum matching. But the change leads to a new G' containing fewer edges. We call such a change a *redefinition* of M^*.

If the component is a cycle, the cycle must contain an even number of edges. This permits the same redefinition of M^*. Thus, in a finite number of such redefinitions we must have $|M^*| = |M|$, completing the proof.

Our auguement yields

Corollary. *If M_1 and M_2 are both maximum matchings of G one may be obtained from the other through a finite number of redefinitions.*

Thus to solve the matching problem a method is needed to find augmenting paths or to show that none exist. Edmonds [4] correctly pointed out that the theorem itself does not indicate an obvious algorithm. For to have a "good" algorithm it is necessary to show that the number of steps required to obtain a solution is less than exponential in the size of the graph, i.e., the algorithm must be better than sheer exhaustive search and its difficulty should increase only algebraically with the size of the graph. Edmonds proposed a "good" method which is based on the idea of "shrinking" odd cycles of edges into pseudonodes and working on the reduced graph in searching for augmenting paths. Given a simple odd cycle of edges together with its vertices B in $G = (V, E)$, the *reduced graph* G/B, said to be obtained from G by *shrinking* B, is the graph consisting of vertices v_i in G but not in B, and a (pseudo) vertex v_B; and consisting of edges (v_i, v_j) for v_i, v_j in G but not B if $(v_i, v_j) \in E$, and edges (v_B, v_j) for $v_j \notin B$ if $(v_i, v_j) \in E$ for some $v_i \in B$.

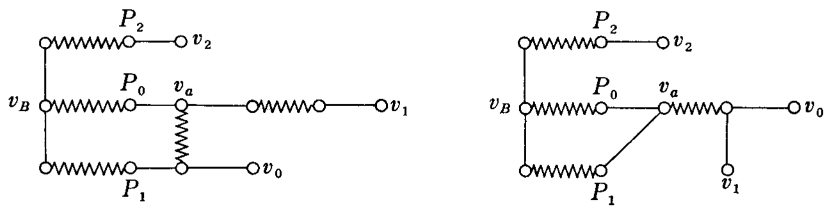
Lemma 1 ([4]). *Let B be a simple odd cycle of $2k + 1$ edges together with its vertices. Then, if M_1 is a matching in G/B there exists a maximum matching M_B of B, $|M_B| = k$, such that $M = M_1 \cup M_B$ is a matching for G.*

Proof. Since M_1 is a matching of G/B only one edge of M_1 is incident to v_B hence to some v_b in B. Thus, in B, choose for M_B the k unique edges which are not pairwise incident at a vertex and leave v_b exposed. The lemma implies that if an augmenting path can be found in G/B relative to some M_1 then an augmenting path obtains in G relative to the corresponding $M = M_1 \cup M_B$.

Theorem 2 (Edmonds [4]). *Let M be a matching leaving at least two vertices of G' exposed, B a simple odd cycle of $2k+1$ edges together with its vertices, and $M_B \subset M$, $|M_B| = k$, a maximum set of matching edges in B. Suppose v_b, the unique vertex not incident to an edge of M_B in B, is either exposed or connected to an exposed vertex v_0 along an alternating path beginning with an edge in M. Then M is a maximum matching for G if and only if $M \cap (G/B)$ is a maximum matching for G/B.*

Proof. If M is a maximum matching and $M_1 = M \cap (G/B)$ is not, then the latter admits an augmenting path which, by Lemma 1, implies M does as well. This is a contradiction by Theorem 1.

Suppose, then, that M_1 is a maximum matching but that M is not. Then there exists an augmenting path P in G relative to M connecting exposed vertices v_1 and v_2 which must include an edge of B. Let $P_i \subset P$ be the part of the path first joining v_i to a vertex of B $(i = 1, 2)$, and let P_0 be the alternating path joining v_b to v_0. If either P_1 or P_2 first hits a node of B along an edge in M, hence in M_1, then $(P_1 \cup P_2) \cap (G/B)$ is an augmenting path, contradicting the maximality of M_1. Thus we may assume P_1 and P_2 both first hit a vertex of B along an edge not in M, hence hit v_B along an edge not in M_1. If P_0 is distinct from either P_1 or P_2 the same contradiction results. So, suppose that P_0 first hits P_1 (and not P_2) from v_B to v_1 in G/B at v_a. Then, either the cycle of edges formed by P_0 and P_1 on G/B between v_B and v_a is even or it is odd.



If the cycle is even then define P_2^1 to be the alternating path going from v_2 to v_B along P_2 and v_B to v_a along P_0; define P_1^1 to be the alternating path going from v_1 to v_a along P_1; and P_0^1 to the alternating path going from v_0 to v_a along P_0. The paths P_0^1, P_1^1, and P_2^1 satisfy the same assumptions as P_0, P_1, P_2

did (where v_a takes the role of v_B), but the length of the path P_0^1 is strictly less than that of P_0. Repeat. In a finite number of redefinitions either P_0^k is distinct from P_1^k or P_2^k, a contradiction; or P_0^k is identical to P_1^k or P_2^k implying it is distinct from one or the other, again a contradiction; or, an odd cycle D is formed. In this last case shrink D to obtain the graph $G_2 = (G/B)/D$. In this graph the edge $P_1 \cap G_2$ incident to v_D in G_2 belongs to $M_1 \cap G_2$. Therefore $(P_1 \cup P_2) \cap G_2$ is an augmenting path in G_2 which implies, by Lemma 1, that $G_1 = G/B$ admits an augmenting path as well. This is a contradiction and establishes the theorem.

These observations lead to the following algorithm for finding augmenting paths or showing none exists [4]. Given a graph G and some matching M in which at least two nodes of G are exposed (otherwise no augmenting path is to be found) call one exposed node, say v_0, even. Then, iteratively define nodes to be *even* or *odd* by the rules below. We will say that a node which has been defined odd or even is *paired.* A node defined to be even (odd) has an alternating path joining it to v_0 with first edge in M (not in M).

1. If vertex v is even, w unpaired, w exposed, $(v, w) \in G$ an augmenting path from w to v_0 exists.

2. If vertex v is even, w unpaired, w not exposed, $(v, w) \in \tilde{M}$ $(= E - M)$, define w to be odd.

3. If vertex v is even, w is even, $(v, w) \in \tilde{M}$, or if v is odd, w is odd, and $(v, w) \in M$ an odd cycle B of $2k + 1$ edges containing k matching edges and the vertices v and w exists. Shrink B, to obtain a reduced graph, and define v_B to be even.

4. If vertex v is odd, w unpaired, $(v, w) \in M$, define w to be even.

At every stage the (reduced) graph of paired vertices is a tree with every even (odd) vertex v connected to v_0 along an alternating path beginning with an edge in M (not in M). These rules must result either in an augmenting path in some reduced graph which can be used to determine a larger matching M^1 in G, or in a reduced graph $G_k = G/B_1/\ldots/B_k$ where every even vertex is connected only to odd vertices. Suppose the subgraph of G corresponding to paired vertices in G_k together with all edges joining them is J, and that the matching edges M in J are M_J. M_J is a maximum matching for J. Eliminate from G all edges having exactly one end in J to obtain two disjoint subgraphs J and $G - J$. Use the algorithm only on $G - J$, and repeat. If no augmenting paths can be found in

$G - J$ the matching M is a maximum matching by virtue of

Theorem 3 [4]. *$M = M_1 \cup M_J$ is a maximum matching of G if and only if M_J and M_1 are maximum matchings for J and $G - J$, respectively.*

Proof. Clearly, if either M_J or M_1 is not a maximum matching then $M = M_1 \cup M_J$ is not. So suppose M_J and M_1 are maximum matchings but M is not. Then there exists an augmenting path P in G which must use one or more edges (v, w) with $v \in J$ and $w \in G - J$. Suppose only one such edge is in P. Then v_0 must be one of the exposed nodes of P. Therefore there must exist an alternating path beginning at v and terminating at v_0 in J, and the node corresponding to or including v in G_k must be odd, since otherwise w would be paired. $P \cap G_k$ is alternating in $J \cap G_k$, contains an even number of edges $2k$ and $k + 1$ nodes defined to be even since v_0 is even and every edge of $M_J \cap G_k$ has exactly one even incident node. This implies two even nodes are incident in G_k, a contradiction. If there is more than one edge $(v, w) \in P$, $v \in J$, $w \in G - J$, the same argument applies, completing the proof.

It should be noted why this is a "good" algorithm. If n is the number of edges in G then given any matching M the algorithm requires looking at each edge at most once to either prove M is a maximum or locate an augmenting path which permits improving M. Thus it is clear that in at most n^2 looks the problem must be solved, since at most n looks locates an augmenting path, if any exists.

The difficulty with this approach, however, is that the "memory" requirement in its implementation can be excessive. It would appear that shrinking odd circuits to obtain reduced graphs, and reduced-reduced graphs, etc., together with the necessity of expanding back to the original G makes it extremely difficult to keep track of and organize the information necessary to implement the algorithm in a completely prescribed manner, e.g., as a computer program. Building upon the philosophical discussion of Edmonds in which he discusses the importance of a "good" algorithm in terms of the number of steps necessary to find a solution, it seems important as well to consider the memory requirement necessary to carry out such a step. Users of modern day computing machinery are happy to testify to the fact that memory and speed, or number of steps, compete. Unfortunately, there seems to be no theory by which the nature of this competition can be analyzed. Nevertheless, motivated by this consideration, and by the success of labelling techniques in the closely related network flow problem, a labelling approach is developed below for solving the maximum matching problem. This approach is purely local in nature; that is, requires only information stored at vertices of the original graph G, and completely does away with shrinking odd cycles and reduced graphs.

3. LABELLING

Given a graph G and a matching M, let v_0 be an exposed vertex, and assume at least one other exposed vertex exists. Then we assign labels to vertices, according to stated rules, having the following meanings. If an arbitrary vertex x carries the label

(1) $[x', -]$, then there exists an alternating path, denoted $p_1(x)$, of labelled vertices beginning at x with edge $(x, x') \notin M$ and ending at v_0;

(2) $[-, x'']$, then there exists an alternating path, denoted $p_2(x)$, of labelled vertices beginning at x with edge $(x, x'') \in M$ and ending at v_0;

(3) $[x', x'']$, then both of the above simultaneously.

In the latter case x is said to be *doubly labelled*; in the former cases, *singly labelled*. The alternating path or paths $p_i(x)$, where i indicates that the path begins with edge (x, y), y the ith component of x's label, are defined inductively by "backtracking" from vertex to vertex as indicated by alternate components of the labels. We use the symbols $-, +, 0$ as components of the labels to mean, respectively, the component is empty, is nonempty, and is either empty or not. Finally, every labelled vertex x carries an *exponent*, $e(x)$, which is a nonnegative integer used in determining order in the application of the labelling rules.

Rule 0. *Label an exposed vertex* v_0 *with* $[v_0, v_0]$ *and let* $e(v_0) = 0$.

Then, at any stage, use any one of the following rules given a labelled vertex v^* (the "origin") with label $[0, +]$ and $e(v^*) = \max\{e(x): x$ labelled $[0, +]$ and admitting the application of at least one rule$\}$. If no such v^* exists, the current matching M gives a maximum matching M_J for the subgraph J consisting of all labelled vertices of G together with edges connecting them. Confine subsequent labelling to the subgraph $G - J$, and store or remember M_J.

Rule 1 ("breakthrough"). *There exists a vertex w, unlabelled, $(v, w) \notin M$ and w is exposed. This means an augmenting path has been found between exposed vertices w and v_0. Reverse the assignment of edges to M in $p_2(v^*)$ and adjoin (v^*, w) to M, to obtain a new matching. Erase all labels and exponents and return to rule* 0.

Rule 2 ("tandem label"). *There exist vertices w_1 and w_2, unlabelled, $(v^*, w_1) \notin M$ and $(w_1, w_2) \in M$. Label w_1 with $[v^*, -]$, w_2 with $[-, w_1]$ and let $e(w_1) = e(v^*) + 1$, $e(w_2) = e(v^*) + 2$.*

Rule 3 ("double label"). *There exists a vertex w with label $[0, +]$, $(v^*, w) \notin M$, $e(v^*) \geqq e(w)$. v^* and certain vertices on $p_2(v^*)$ should receive a second label since an alternating path from v^* to w via $p_2(w)$ to v_0 exists. Let $v^* = v_1, \ldots, v_k, v_{k+1}$ be successive vertices on $p_2(v^*)$ with $e(v_j) > e(w)$ for $j \leqq k$ and $e(v_{k+1}) \leqq e(w)$. Then give second labels to $v^* = v_1, \ldots, v_k$ (changing any labels if some are already doubly labelled) with v^* receiving w in the first component, v_2 receiving v in the second, v_3 receiving v_2 in the first, etc. Redefine exponents by setting $e(v_j) = e(w)$ for $1 \leqq j \leqq k$.*

These are the only possibilities we need consider. The other logical possibilities either cannot occur or lead to no additional or potential augmenting paths. The cases are these: (1) There exists a node w labelled with $[+, -]$ where $(v^*, w) \notin M$, which simply indicates the existence of more than one alternating path from w to v_0 beginning with an edge not in M. (2) There exists nodes w_1 unlabelled, w_2 labelled, $(v^*, w_1) \notin M$, $(w_1, w_2) \in M$. But this is impossible by the rules. For if w_2 has a label $[0, +]$ then this label must be $[0, w_1]$, since M is a matching, implying w_1 is labelled after all. So, suppose w_2 has a label $[+, -]$. This is impossible for no rule assigns a label of form $[+, -]$ without giving its neighbor along an edge of M a label of form $[-, +]$. This completes the description of the algorithm.

This algorithm is good in the sense of Edmonds. Let m be the number of vertices of G, n the number of edges of G. Usually one would expect $n > m$ for otherwise, if G is a connected graph, the problem is trivial. If we count the number of times a label is assigned (including changed) as a step, it takes at most m^2 steps before either an augmenting path is located or no further labelling is possible. For each vertex can be labelled at most m times, and there are m vertices. Since there are at most m possible origins for labelling, m^3 is as upper bound on the number of labellings.

4. STRUCTURE OF LABELLED GRAPHS AND VALIDITY OF ALGORITHM

The simplicity of the algorithm is not matched by a simplicity of structure. It seems that to achieve a simpler structure in labelled graphs it is necessary to make more stringent demands in the labelling process.

In order to be able to describe the pertinent structure of labels we introduce a number of terms. A vertex x is *usable* if it carries a label $[0, +]$ and either has an unlabelled neighbor or a neighbor $\bar{x}$ with label $[0, +]$ and $e(x) > e(\bar{x})$. A *branch* of a vertex x, denoted $B(x)$, is the set of vertices $B(x) = \{y : y$ labelled, $x \in p_1(y)$ or $x \in p_2(y)\}$. We take $x \in B(x)$.

Lemma 1. (i) *The exponents along any defined alternating path $p_i(x)$ are nonincreasing.*

(ii) *The exponents of two successive vertices along any $p_i(x)$ differ by exactly* 1 *if both are singly labelled, and differ by at most* 1 *along an edge of M.*

(iii) *Any vertex with label* $[0, +]$ *has an even exponent.*

Proofs are immediate by observing rules 2 and 3 preserve all properties.

Lemma 2. *If a node x is doubly labelled* $[+, +]$, *then the node $\bar{x}$ incident to x along $(x, \bar{x}) \in M$ is also doubly labelled.*

Double labelling can occur only by labelling according to rule 3, and the parity of exponents is sufficient to establish this fact.

Lemma 3. *Suppose v carries the label* $[-, +]$ *and $z \in B(v)$ is* *usable. Then every node receiving a label after v contains v in its paths unless v becomes doubly labelled.*

Proof. This simply says that a branch grown by labelling from a singly labelled v $[-, +]$ is grown until no further growth is possible before a distinct branch is considered. To prove the lemma assume it true at some stage of labelling. We show it must still be true after one labelling step. Suppose $z \in B(v)$ is usable and v has label $[-, +]$. Then $e(y) \geqq e(v)$ for $y \in B(v)$ and the origin of labelling $v^* \in B(v)$. For, if $v^* \notin B(v)$, $e(v^*) \geqq e(v)$ and hence v should not have been labelled when it was since it was labelled from a node with exponent $e(v) - 2$.

Suppose rule 2 is used from v^* to label w_1 and w_2 previously not labelled. Then, clearly, v belongs to the paths of w_1 and w_2.

Suppose rule 3 is used from v^* due to a neighbor w having label $[0, +]$, $e(v^*) > e(w)$. If $e(w) < e(v)$ then, since $v \in p_2(v^*)$, v would be doubly labelled. If $e(w) \geqq e(v)$ and $w \in B(v)$ then, since $v \in p_2(w)$, v would belong to the path of any newly labelled node. If $e(w) \geqq e(v)$ and $w \notin B(v)$ then, again, v should not have been labelled when it was.

Lemma 4. *Suppose v carries the label $[-, +]$, $e(v) \leqq e(u)$ and $B(u) \cap B(v) = \phi$ at some stage of labelling. Then if $y \in B(u)$ and $z \in B(v)$ have labels $[0, +]$ they do not both have unlabelled neighbors.*

Proof. Either v received the label $[-, +]$ before y, or y received the label $[0, +]$ before v was labelled. So, if the lemma is false, then in the first case v belongs to the paths of y, contradicting $B(u) \cap B(v) = \phi$; in the second case v should not have been labelled when it was since it was labelled from a node having exponent $e(v) - 2 < e(y)$.

Theorem 4. *Suppose v carries the label $[-, +]$, u is labelled, and $e(v) \leqq e(u)$ at some stage of labelling. Then, if $B(u) \cap B(v) = \phi$, and $y \in B(u)$ and $z \in B(v)$ have labels $[0, +]$*

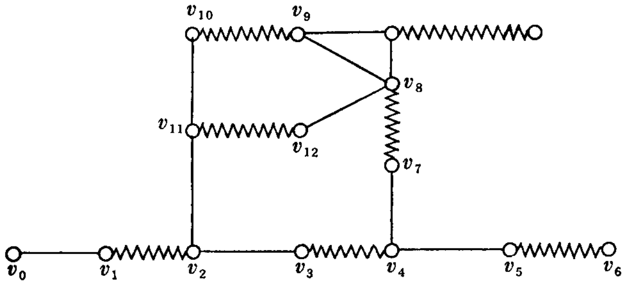
(i) *y (respectively, z) usable implies $z(y)$ has no neighbor in $B(u)$ (in $B(v)$); and*

(ii) *y and z are not neighbors.*

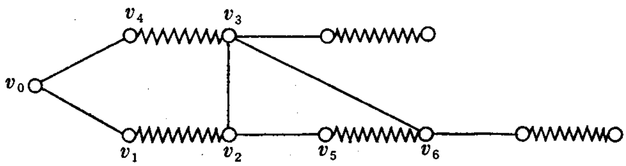
However, if $B(u) \cap B(v) \neq \phi$ then

(iii) *v belongs to all the paths of u.*

It should be pointed out that it is not possible to strengthen (i) and (ii) to say that $y(z)$ usable implies $z(y)$ is not usable; that is, a branch admitting of no growth at one stage of labelling may admit growth at a later stage. The following example illustrates this fact. Vertices v_i are indexed in the order in



which they are labelled and vertices without names are not labelled. Suppose v_0 through v_6 are labelled. $B(v_6) = v_6$ cannot be grown. Labelling continues from v_4 to through v_{12}, double labelling occurs due to v_{12} and v_8, then due to v_{11} and v_2. At this point $e(v_9) = 8$, $e(v_6) = 6$ yet both branches have usable vertices with $B(v_9) \cap B(v_6) = \phi$. The same example shows it is even impossible to assert that $y(z)$ usable implies $z(y)$ has no unlabelled neighbor. Also, the asymmetry in v and u is necessary as shown by the following example. For (iii) $B(v_2) \cap B(v_3) \neq \phi$ yet v_2 is on only one path of v_3. For (i) and (ii), $B(v_3) \cap B(v_6) = \phi$, yet both branches have vertices labelled $[0, +]$



having neighbors in the other branch (and unlabelled neighbors).

To prove the theorem we will use induction; namely, we show that if the structure described holds before use of labelling rules 2 or 3 then the same structure obtains after the labelling. However, as a preliminary, we establish

Lemma 5. *Suppose* $e(v) \leq e(u)$, *v carries the label* $[-, +]$, *and u is labelled immediately after a labelling by rule* 3. *Assume the structure of Theorem* 4 *obtains before this labelling. Then*

$B(u) \cap B(v) = \phi$ holds after the labelling if and only if it holds before.

Proof. Suppose $B(u) \cap B(v) = \phi$ before but $B(u) \cap B(v) \neq \phi$ after. Then there must exist a vertex $x \in B(u) \cap B(v)$ after labelling while before either (a) $x \notin B(v)$ or (b) $x \notin B(u)$ and $x \in B(v)$.

(a) There must exist a node y belonging to a path of x, with $v \notin p(y)$ whose label becomes changed by rule 3. Hence, $y \in p_2(v^*)$, where v^* is the origin of labelling, and either $v \in p_2(w)$ or $v \in p_2(v^*)$ between the vertices v^* and y, where w is the neighbor of v^* prompting the labelling. In the first instance, $e(v) \leqq e(w) < e(y)$ and $v^* \in B(y)$, $w \in B(v)$. But by Theorem 4 (ii) implies $B(v) \cap B(y) \neq \phi$, since v^* and w are labelled with $[0, +]$, and by (iii) this implies v belongs to the paths of y, contrary to hypothesis. In the second instance v would have to be doubly labelled by rule 3.

(b) There must be a node y in a path of x, $u \notin p(y)$ whose label becomes changed by rule 3. Thus, $y \in p_2(v^*)$ and $u \in p_2(w)$ or $u \in p_2(v^*)$ between v^* and y. In the first instance, $e(v) \leqq e(u) \leqq e(w) < e(y)$ and $x \in B(v) \cap B(y)$ implying v belongs to both paths of y and hence $v \in p_2(v^*)$. But this contradicts (ii) since $v^* \in B(v)$ and $w \in B(u)$ are neighbors carrying labels $[0, +]$. In the second instance we know (by (iii)) that v belongs to the paths of x. So, if $e(v) \leqq e(y)$, then v belongs to the paths of y hence to a path of u, a contradiction. Otherwise, $e(v) > e(y)$. But, then, after labelling $e(u) = e(w) < e(y) < e(v)$, again a contradiction.

The converse is easily established. Suppose $B(u) \cap B(v) \neq \phi$ before labelling but $B(u) \cap B(v) = \phi$ after labelling. By (iii) v belongs to the paths of u before but not after. This can only mean that a node y belonging to a path of u and containing v in its paths came to be doubly labelled. But this implies $y \in p_2(v^*)$ whence $v^* \in B(v)$ and is usable. By Lemma 3 this means all newly defined paths contain v. This establishes Lemma 5.

We turn to Theorem 4 and treat each part separately. It is trivial to verify that labelling rule 2 preserves the properties, so we consider only use of rule 3 and recall Lemma 5 applies.

(i) Suppose $v^* \in B(v)$ before applying rule 3. Then we show that if $y \in B(u)$ carries a label $[0, +]$ then it can have no neighbor in $B(v)$. For otherwise, before labelling either (a) $y \notin B(u)$ or (b) $y \in B(u)$ and has label $[+, -]$ or (c) $y \in B(u)$, has label $[0, +]$, but has no neighbor in $B(v)$. In case (a) this implies $w \in B(u)$ or $v^* \in B(u)$, both contradicting $B(v) \cap B(u) = \phi$. In case (b) $y \in p_2(v^*)$, hence v is in the paths of y, again contradicting $B(v) \cap B(u) = \phi$. Finally, in case (c) some neighbor t of y comes to belong to $B(v)$ due to double labelling a vertex $s \in p_2(v^*)$ belonging to a path of t. $e(s) \geqq e(v)$, since otherwise, v would be doubly labelled after rule 3. But then $v^* \in B(s)$ is usable, $t \in B(s)$, and $y \in B(u)$, with y labelled $[0, +]$, contradicting our inductive hypothesis (i).

Suppose $v^* \in B(u)$ before applying rule 3. We show that if $z \in B(w)$ carries a label $[0, +]$ then it can have no neighbor in $B(u)$ after labelling. For otherwise, before labelling either (a) $z \notin B(v)$ or (b) $z \in B(v)$ and has label $[+, -]$ or (c) $z \in B(v)$, has label $[0, +]$, but has no neighbor in $B(u)$. In case (a) either $w \in B(v)$ or $v^* \in B(v)$, contradicting $B(v) \cap B(u) = \phi$. In case (b) v belongs to the path of z, z to the $p_2(v^*)$, implying $v^* \in B(v)$, also a contradiction. In case (c) some neighbor t of z comes to belong to $B(u)$ due to double labelling some $s \in p_2(v^*)$. This means that either $w \in B(u)$ or $u \in p_2(v^*)$. If $w \in B(u)$ then $e(u) \leqq e(w) < e(s)$. If $u \in p_2(v^*)$, then the new exponent of s, which is strictly less than its old exponent $e(s)$, equals the new exponent of u which is assumed greater or equal to $e(v)$. So, $e(s) \geqq e(v)$ in either case. Consider $B(s)$ and $B(v)$. $v^* \in B(s)$, and is usable, $t \in B(s)$, $z \in B(v)$ with label $[0, +]$, and t and z are neighbors. This contradicts our hypothesis (i) before labelling.

So suppose $v^* \notin B(v)$ and $v^* \notin B(u)$. If $B(v)$ grows then either $w \in B(v)$ or $v \in p_2(v^*)$, i.e., $v^* \in B(v)$. The latter is a contradiction. So is the former since then $w \in B(v)$ and $v^* \in B(v^*)$ are neighbors with $e(v) \leqq e(w) < e(v^*)$ so that v belongs to the paths of v^*. So $B(v)$, by Lemma 3, has no usable nodes. So, if $y \in B(u)$ is usable then $z \in B(v)$ having label $[0, +]$ can have a neighbor $t \in B(u)$ after labelling only if $t \notin B(u)$ before. A contradiction is derived in precisely the same manner as case (c) immediately above. This establishes (i).

(ii) The arguments for (i) apply directly to (ii) except for minor modifications for the case $v^* \notin B(v)$ and $v^* \notin B(u)$. In this case $B(v)$ cannot, as shown above, grow. So the only question concerns a $z \in B(v)$ with label $[0, +]$ gaining a neighbor $t \in B(u)$ having label $[0, +]$. If $t \notin B(u)$ before labelling the same argument used above applies. So suppose $t \in B(u)$ and has the label $[+, -]$ before labelling but t has label $[+, +]$ after labelling. Then $t \in p_2(v^*)$, that is, $v^* \in B(t)$ or $v^* \in B(u)$, a contradiction.

portant practical applications [1] [2]; it is closely related to network flow problems, and an understanding of this class would throw light on the structure of more general integer programming problems. We take as the formulation for such problems

(4) *maximize* $\sum c_j x_j$ *when* $\sum \alpha_j x_j \mathrm{R} \alpha_0$, $x_j \geqq 0$, x_j *integer-valued where* α_j, $j \neq 0$, *is a column of* 0's *and* 1's *containing at most two* 1's, α_0 *is a column of nonnegative integers, and* R *represents relations which are either* $\geqq$, *or* $\leqq$ *or* $=$.

The simple graph matching problem treated in this paper is of form (4) with all R representing $\leqq$, α_0 a column of 1's, $c_j = 1$ and α_j containing exactly two 1's for all $j \neq 0$. The matrix $A = (\alpha_1, \ldots, \alpha_n)$ is the node-arc incidence matrix of the graph G. The network flow problems are of form (4) except that each α_j either contains one non-zero entry which can be ± 1, or two with one $+1$ and one -1. Note that the minors of A in (4) are all of form $\pm 2^k$, for some integer $k \geqq 0$.

In conclusion, we indicate one generalization to a restricted problem (4) which can itself be generalized further in various directions depending on the form of (4) which is considered (these are being developed in a separate paper). Suppose we consider a *weighted graph* G, a graph G in which every edge e is assigned a positive weight $w(e) > 0$. Problem: find a *maximum weighted matching* on G, that is, find a matching the sum of whose weights is a maximum. This is a problem (4) with $c_j > 0$ the weights, and remaining data as prescribed above for the simple matching problem.

Given a matching M in a weighted graph G define an *augmenting path* P relative to M to be an alternating path or alternating simple cycle having no edge of M incident to only one vertex of P and with the property $w_M(P) = \{\sum w(e) - \sum w(d) : e \in P \cap \tilde{M},\ d \in P \cap M\} > 0$.

Theorem 7. *M is a maximum weighted matching if and only if M admits no augmenting path.*

If M admits an augmenting path it is not of maximum weight.

Suppose that M is a matching which admits no augmenting path but that $M^* \neq M$ is a maximum weighted matching. Consider, as above, the subgraph G' of G containing all vertices of G and edges e of G which satisfy $e \in M$ and $e \notin M^*$ or $e \notin M$ and $e \in M^*$. Look at a connected component H of G'. H can only be a simple path or a simple cycle consisting of an even number of edges and having no edge of M or of M^* incident at only one node. In either case since M admits no augmenting path, $w_M(H) \leqq 0$. But since M^* is a maximum matching it can admit no augmenting path so that $0 \geqq w_{M^*}(H) = -w_M(H)$ or $w_{M^*}(H) = w_M(H) = 0$, proving that the weights of M and M^* must be identical.

References

1. Balinski, M. L. "Integer Programming: Methods, Uses, Computation," *Management Sci.*, **12** (1965), 253–313.
2. Balinski, M. L. and Quandt, R. E. "On an Integer Program for a Delivery Problem," *Operations Res.*, **12** (1964), 300–304.
3. Berge, Claude. "Two Theorems in Graph Theory," *Proc. Nat. Acad. Sci. U.S.A.*, **43** (1957), 842–844.
4. Edmonds, Jack. "Paths, Trees and Flowers," *Canad. J. Math.*, **17** (1965), 449–467.
5. Norman, R. Z. and Rabin, M. O. "An Algorithm for the Minimum Cover of a Graph," *Proc. Amer. Math. Soc.*, **10** (1959), 315–319.

Discussion on Professor Balinski's Paper

PROFESSOR B. B. BHATTACHARYYA: The labelling algorithm for the maximum matching problem may be viewed as a special method for solving a certain class of integer-programming problems. Linear programmers have been successful in solving transportation-model-type integer programming problems without any special difficulty, as the extreme points of the convex set of feasible solutions of the associated problem, with integral restrictions removed, are integer-valued. But a genuine difficulty may arise in solving the maximum matching problem in a non-bipartite graph as we may get fractional optimal solutions by removing integral restrictions. This difficulty was met by Edmonds by the device of "shrinking blossoms." It would be of help to understand the labelling algorithm if we set up a correspondence between the two algorithms.

Edmonds starts by rooting an alternating tree in a matching and partitions the vertices into outer and inner vertices.

The vertices labelled (0, +) in the labelling algorithm seem to be outer vertices of the rooted alternating tree. The breakthrough rule corresponds to the situation where the planted tree is augmenting. The tandem label rule enlarges the already existing tree while non-existence of a vertex v^* under Rule 0 indicates that the tree is hungarian. The "double label rule" deals with a blossoming tree, and this rule provides a method of keeping the matching in the blossom in view; and hence it becomes possible to avoid the problem of resurrecting shrinking blossoms at the end of the calculations.

It seems plausible that one should be able to devise a labelling algorithm for the maximum matching problem by solving the associated minimum cover problem by using the concept of "reducing path" introduced by Norman and Rabin. In this procedure labelling should start naturally from a vertex which has more than one edge incident on it.

It has been noted by Edmonds that the maximum matching problem may be solved as an ordinary linear programming problem, by substituting for the zero-one conditions the additional constraints that the variables are non-negative and that for any set R of $2K + 1$ vertices, the sum of the variables that correspond to the edges with both end points in R is not greater than K. It may be of interest to study the dual of this problem to see whether it leads to some simple algorithm.

The possibility of extending the algorithm to the weighted maximum matching problem is of great interest. This would be very helpful to an integer programmer as otherwise he has to use more laborious cutting plane techniques to deal with the situation.

COMBINATORIAL MATHEMATICS AND ITS APPLICATIONS

Proceedings of the Conference Held at the University of North Carolina at Chapel Hill April 10–14, 1967

Edited by
R. C. BOSE
and
T. A. DOWLING