Matching: A Well-Solved Class of Integer Linear Programs

Jack Edmonds[1] and Ellis L. Johnson[2]

[1] National Bureau of Standards, Washington, D.C., U.S.A.
[2] I.B.M. Research Center, Yorktown Heights, NY, U.S.A.

A main purpose of this work is to give a good algorithm for a certain well-described class of integer linear programming problems, called *matching problems* (or the *matching problem*). Methods developed for simple matching [2,3], a special case to which these problems can be reduced [4], are applied directly to the larger class. In the process, we derive a description of a system of linear inequalities whose polyhedron is the convex hull of the admissible solution vectors to the given matching problem. At the same time, various combinatorial results about matchings are derived and discussed in terms of graphs.

The *general integer linear programming problem* can be stated as:

(1) Minimize $z = \sum_{j \in E} c_j x_j$, where c_j is a given real number, subject to

(2) x_j an integer for each $j \in E$;

(3) $0 \leq x_j \leq \alpha_j$, $j \in E$, where α_j is a given positive integer or $+\infty$;

(4) $\sum_{j \in E} a_{ij} x_j = b_i$, $i \in V$, where a_{ij} and b_i are given integers;

V and E are index sets having cardinalities $|V|$ and $|E|$.

(5) The integer program (1) is called a *matching problem* whenever

$$\sum_{i \in V} |a_{ij}| \leq 2$$

holds for all $j \in E$.

(6) A *solution* to the integer program (1) is a vector $[x_j]$, $j \in E$, satisfying (2), (3), and (4), and an *optimum solution* is a solution which minimizes z among all solutions. When the integer program is a matching problem, a solution is called a *matching* and an optimum solution is an *optimum matching*.

If the integer restriction (2) is omitted, the problem becomes a linear program. An optimum solution to that linear program will typically have fractional values. There is an important class of linear programs, called transportation or network flow problems, which have the property that for any integer right-hand side b_i, $i \in V$, and any cost vector c_j, $j \in E$, there is an optimum solution which has all integer x_j, $j \in E$. The class of matching probems includes that class of linear programs, but, in addition, includes problems for which omitting

M. Jünger et al. (Eds.): Combinatorial Optimization (Edmonds Festschrift), LNCS 2570, pp. 27–30, 2003.

the integer restriction (2) results in a linear program with no optimum solution which is all integer.

Many interesting and practical combinatorial problems can be formulated as integer linear programs. However, limitations in the known methods for treating general integer linear programs have made such formulations of limited value. By contrast with general integer linear programming, the matching problem is *well-solved*.

(7) **Theorem.** *There is an algorithm for the general matching problem such that an upper bound on the amount of work which it requires for any input is on the order of the product of (8), (9), (10), and (11). An upper bound on the memory required is on the order of (8) times (11) plus (9) times (11).*

(8) $|V|^2$, the number of nodes squared;

(9) $|E|$, the number of edges;

(10) $\sum_{i\in V} |b_i| + 2\sum_{\alpha_j<\infty} \alpha_j$;

(11) $\log(|V| \max |b_i| + |E| \max_{\alpha_j<\infty} \alpha_j) + \log(\sum_{j\in E} |c_j|)$.

(12) **Theorem.** *For any matching problem,* (1)*, the convex hull* P *of the matchings, i.e., of the solutions to* [(2), (3)*, and* (4)]*, is the polyhedron of solutions to the linear constraints* (3) *and* (4) *together with additional inequalities:*

(13) $\sum_{j\in W} x_j - \sum_{j\in U} x_j \geq 1 - \sum_{j\in U} \alpha_j$.

There is an inequality (13) *for every pair* (T, U) *where* T *is a subset of* V *and* U *is a subset of* E *such that*

(14) $\sum_{i\in T} |a_{ij}| = 1$ for each $j \in U$;

(15) $\sum_{i\in T} b_i + \sum_{j\in U} \alpha_j$ is an odd integer.

The W *in* (13) *is given by*

(16) $W = \{j \in E \,:\, \sum_{i\in T} |a_{ij}| = 1\} - U$.

(17) Let Q denote the set of pairs (T, U). The inequalities (13), one for each $(T, U) \in Q$, are called the *blossom inequalities* of the matching problem (1).

By Theorem (12), the matching problem is the linear program:

(18) Minimize $z = \sum_{j\in E} c_j x_j$ subject to (3), (4), and (13).

(19) **Theorem.** *If* c_j *is an integral multiple of* $\sum_{i\in V} |a_{ij}|$*, and if the l.p. dual of* (18) *has an optimum solution, then it has an optimum solution which is integer-valued.*

Using *l.p.* duality, theorems (12) and (19) yield a variety of combinatorial existence and optimality theorems.

To treat matching more graphically, we use what we will call *bidirected* graphs. All of our graphs are bidirected, so *graph* is used to mean bidirected graph.

(20) A *graph* G consists of a set $V = V(G)$ of *nodes* and a set $E = E(G)$ of *edges*. Each edge has one or two *ends* and each end *meets* one node. Each end of an edge is either a *head* or a *tail*.

(21) If an edge has two ends which meet the same node it is called a *loop*. If it has two ends which meet different nodes, it is called a *link*. If it has only one end it is called a *lobe*. An edge is called *directed* if it has one head and one tail. Otherwise it is called *undirected*, *all-head* or *all-tail* accordingly.

(22) The node-edge incidence matrix of a graph is a matrix $A = [a_{ij}]$ with a row for each node $i \in V$ and a column for each edge $j \in E$, such that $a_{ij} = +2$, $+1$, 0, -1, or -2, according to whether edge j has two tails, one tail, no end, one head, or two heads meeting node i. (Directed loops are not needed for the matching problem.)

(23) If we interpret the capacity α_j to mean that α_j copies of edge j are present in graph G^α, then x_j copies of j for each $j \in E$, where $x = [x_j]$ is a solution of [(2), (3), (4)], gives a subgraph G^x of G^α. The *degree* of node i in G^x is b_i, the number of tails of G^x which meet i minus the number of heads of G^x which meet i. Thus, where x is an optimum matching, G^x can be regarded as an "optimum degree-constrained subgraph" of G^α, where the b_i's are the degree constraints.

(24) A Fortran code of the algorithm is available from either author. It was written in large part by Scott C. Lockhart, who also wrote many comments interspersed through the deck to make it understandable. Several random problem generators are included.

It has been run on a variety of problems on a Univac 1108, IBM 7094, and IBM 360. On the latter, problems of 300 nodes, 1500 edges, $b = 1$ or 2, $\alpha = 1$, and randon c_j's from 1 to 10, take about 30 seconds. Running times fit rather closely a formula which is an order of magnitude better than our theoretical upper bound.

References

1. Berge, C., *Théorie des graphes et ses applications*, Dunod, Paris, 1958.
2. Edmonds, J., Paths, trees, and flowers, *Canad. J. Math.* 17 (1965), 449–467.
3. Edmonds, J., Maximum matching and a polyhedron with 0,1-vertices, *J. Res. Nat. Bur. Standards* 69B (1965), 125–130.
4. Edmonds, J., An introduction to matching, preprinted lectures, Univ. of Mich. Summer Engineering Conf. 1967.
5. Johnson, E.L., Programming in networks and graphs, Operation Research Center Report 65-1, Etchavary Hall, Univ. of Calif., Berkeley.
6. Tutte, W.T., The factorization of linear graphs, *J. London Math. Soc.* 22 (1947), 107–111.
7. Tutte, W.T., The factors of graphs, *Canad. J. Math.* 4 (1952), 314–328.
8. Witzgall, C. and Zahn, C.T. Jr., Modification of Edmonds' matching algorithm, *J. Res. Nat. Bur. Standards* 69B (1965), 91–98.

9. White, L.J., A parametric study of matchings, Ph.D. Thesis, Dept. of Elec. Engineering, Univ. of Mich., 1967.
10. Balinski, M., A labelling method for matching, Combinatorics Conference, Univ. of North Carolina, 1967.
11. Balinski, M., Establishing the matching poloytope, preprint, City Univ. of New York, 1969.