Matching, Matroids, and Extensions

William H. Cunningham*
Department of Combinatorics & Optimization
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

August 2, 2001

Abstract

Perhaps the two most fundamental well-solved models in combinatorial optimization are the optimal matching problem and the optimal matroid intersection problem. We review the basic results for both, and describe some more recent advances. Then we discuss extensions of these models, in particular, two recent ones—jump systems and path-matchings.

1 Introduction

In the 1960's Edmonds called attention to the class of (polynomially) solvable problems and to certain other problems (such as the Travelling Salesman Problem), which he conjectured to be unsolvable. He also proved that several important models associated with graph matching and matroids are solvable, and observed an apparent connection between solvability and the existence of certain of these structures. In the ensuing years there has been increasing recognition, both of the importance of his results and the validity of his hypothesis.

Two of those original models, the optimal matching problem and the optimal matroid intersection problem, remain the most important well-solved models in combinatorial optimization. Some of the foundations for them already appeared in the 1940's, in the purely combinatorial work of Tutte and Rado. However, it was Edmonds in the 1960's who made these subjects part of combinatorial optimization. He was the first to provide efficient algorithms and polyhedral descriptions. In addition, he found beautiful connections, analogies, generalizations, and applications.

Since then, our understanding of both models has grown. Edmonds' original proofs were mainly algorithmic and polyhedral. Now several different proofs are known, and solution algorithms that are essentially different from his augmenting path algorithms exist. There have also been a number of important successes in extending these models in new directions. I want to emphasize that these two themes—alternate approaches to the classical results, and extensions of those results—are not unrelated. Very often the newer, more general results have been established by methods that were already introduced for matching, but were not the original methods.

*Research partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

In many cases it is simply not known whether the original methods can be generalized. Having an arsenal of other techniques, including ones introduced for "already solved problems", is important.

The fundamental results on matching and matroid intersection have already found their way into textbooks. We review some of the main points, and try to emphasize a few important aspects, such as the Tutte matrix and the Gallai-Edmonds partition, that are less well-known to optimizers. Work on extensions has already exceeded what can be surveyed adequately in a paper of this length, so we have made some choices. The classical extensions found by Edmonds—degree-constrained subgraphs and polymatroid intersection—are very important, but have already appeared in earlier surveys. Of the many other important extensions, several are related to the fundamental notion of submodular functions, and are rather independent of matching theory. Here I refer especially to the submodular flow models of Edmonds and Giles [16] and others, the bisupermodular covering model of Frank and Jordan [18], and, of course, the fundamental problem of minimization of submodular functions, on which there have been exciting developments recently. We continue to be motivated somewhat by the very old question of finding a satisfactory common generalization of optimal matching and optimal matroid intersection. We describe in some detail two recent attempts, jump systems and path-matching.

This paper is mainly expository. However, Theorems 5.3 and 6.5 are new. Also, we summarize in Section 6 some results from [8], which has not yet appeared.

We introduce here a bit of notation. For sets S, T, we denote their symmetric difference (the set of elements in exactly one of them) by $S \triangle T$. We denote by $\mathbf{R}^S$ ($\mathbf{Z}^S$) the set of all real (integral) vectors having components indexed by S. If we have $T \subseteq S$ and $x \in \mathbf{R}^S$, we denote $\sum(x_j : j \in T)$ by $x(T)$. For an undirected graph $G = (V, E)$ and $S \subseteq V$, we denote by $\delta(S)$ (respectively, $\gamma(S)$) the set of edges having one end (respectively, both ends) in S. The subgraph of G having vertex set S and edge set $\gamma(S)$ is denoted by $G[S]$. For a path P of a graph, we use $E(P)$ to denote the set of edges of P.

2 Matching

A *matching* in a graph $G = (V, E)$ is a set M of edges such that no two elements of M have a common end. A matching is said to *cover* the vertices that it is incident with. If B is the set of vertices covered by some matching, then B is called a *matchable set* of G. A matching is *perfect* if it covers all of the elements of V. The number of components of G having an odd number of vertices is denoted by $\mathrm{odd}(G)$. Tutte [41] proved the fundamental theorem about the existence of perfect matchings.

Theorem 2.1 (Tutte's Matching Theorem) *G has a perfect matching if and only if $\mathrm{odd}(G[V \backslash S]) \le |S|$ for all $S \subseteq V$.*

By now there are many proofs of this theorem, and we will mention some below. Tutte's original proof is not one of the better known ones, but it has begun to attract more attention recently. It uses the following idea. Let z_e, $e \in E$, be distinct commuting variables. Let $A(G)$ be a V by V skew-symmetric matrix whose entries satisfy $a_{ij} = \pm z_e$ if $ij = e \in E$, and $a_{ij} = 0$ otherwise. We call $A(G)$ the *Tutte matrix* of G (even though it is not quite unique). Tutte observed, as a direct consequence of nineteenth century theory of determinants, the following unexpected fact.

Theorem 2.2 *G has a perfect matching if and only if $A(G)$ is nonsingular.*

(Note what it means for such a matrix to be nonsingular—that its determinant, viewed as a multivariate polynomial, is not identically zero.) It is a nice exercise to prove Theorem 2.2 directly. Here is an outline of a proof. It is easily checked that each non-zero term in the determinant expansion arises from a set of vertex-disjoint paths covering all the vertices, such that each path is closed, but is otherwise vertex-simple. (Note that the path v, w, v arising from a single edge is such a path.) A perfect matching obviously determines such a set of paths, and moreover, one whose corresponding term in the expansion cannot be cancelled by another term. If such a set of paths has the property that each path is of even length, it is easy to see that G has a perfect matching. On the other hand, if one or more of the paths has odd length, then there is another term that cancels this term. Thus there is a perfect matching if and only if the determinant is not zero.

It follows from Theorem 2.2 that a set B of vertices is matchable if and only if the principal submatrix of $A(G)$ with rows and columns indexed by B, is nonsingular. Further, since the rank of a skew-symmetric matrix is equal to the size of a maximum-size nonsingular principal submatrix, this implies that the rank of the Tutte matrix is equal to the size of a maximum matchable set.

Tutte's Matching Theorem is equivalent to the following result of Berge, now often referred to with both of their names. Note that the size of a largest matchable set is twice the size of a largest matching, so one can write the formula a little more cleanly in terms of matchable sets. However, we have used the more traditional statement.

Theorem 2.3 (Tutte-Berge Formula) *The maximum size of a matching of G is the minimum, over subsets S of V of*

$$\frac{1}{2}\left(|V| - \mathrm{odd}(G[V \backslash S]) + |S|\right).$$

In his classic paper [13] Edmonds gave the first efficient algorithm to find a matching of maximum size (and hence to decide whether a graph has a perfect matching). The algorithm is an augmenting path method—if the current matching M is not maximum, it finds a path P such that $M \bigtriangleup E(P)$ is a matching of size larger by one. If M is maximum it finds a set S such that $|M| = (|V| - \mathrm{odd}(G[V \backslash S]) + |S|)/2$. Thus it also provides another proof of the Tutte-Berge Formula. In fact, the algorithm finds a certain canonical minimizing set S. Namely, define the partition A, C, D of V by

$$\begin{aligned} D = &\ \{v \in V : \text{ there is a maximum matching not covering } v\}; \\ A = &\ \{u \in V \backslash D : uv \in E \text{ for some } v \in D\}; \\ C = &\ V \backslash (D \cup A). \end{aligned}$$

The partition (A, C, D) is called the *Gallai-Edmonds partition* of G. It is clearly well-defined. As can be seen from the following result, this partition tells a great deal about matching properties of G. Its statement uses one more notion—a graph H is *critical* if, deleting any vertex from H, the resulting graph has a perfect matching; such a matching is called a *near-perfect* matching of H.

Theorem 2.4 (Gallai-Edmonds Structure Theorem) *If (A, C, D) is the Gallai-Edmonds partition of G, then every component of $G[D]$ is critical, and every maximum matching of G consists of*

- *a perfect matching of $G[C]$,*
- *a near-perfect matching of H for each component H of $G[D]$, and*
- *an edge joining v to some vertex in D for each $v \in A$.*

It is an easy consequence that A is a minimizing choice for S in the Tutte-Berge Formula. Edmonds' algorithm finds the Gallai-Edmonds partition, in particular, finds as a minimizing set S, the set A.

Weighted matching and matching polyhedra

Edmonds [14] also considered the problem of finding a (perfect) matching of maximum weight, subject to given weights on the edges. (The easier special case in which the graph is bipartite had been solved in the 1950's by Kuhn.) He was able to find a description by linear inequalities of the "Matching Polytope", the convex hull of incidence vectors of matchings.

Theorem 2.5 (Matching Polytope Theorem) *The convex hull of the set of incidence vectors of matchings of G is the set of solutions of*

$$\begin{aligned} x(\delta(v)) &\leq 1 && (v \in V) \\ x(\gamma(S)) &\leq (|S|-1)/2 && (S \subseteq V,\ |S| \ \textit{odd}) \\ x &\geq 0. \end{aligned}$$

Edmonds gave an efficient algorithm that not only finds a maximum weight matching, but verifies that the incidence vector of the matching optimizes $c^T x$ over all x satisfying the system, thus proving the Matching Polytope Theorem. It essentially applies his maximum cardinality matching algorithm to the subgraph consisting of the edges whose dual constraints are satisfied with equality for the current dual solution, and then changes the dual solution (if necessary). Other proofs of the theorem have been given by Lovász and Schrijver. Both of these newer proofs introduced techniques that have been useful in proofs of other important theorems of polyhedral combinatorics.

Cunningham and Marsh [9] proved that the system of linear inequalities in the Matching Polytope Theorem is totally dual integral—for any $c \in \mathbf{Z}^E$, such that the maximum of $c^T x$ subject to the system exists, the dual linear program has an integral optimal solution. This result implies the Tutte-Berge Formula. The proof of [9] is algorithmic. Schrijver [39] has given a very short inductive proof, again introducing a technique that has been used elsewhere.

A version of the Matching Polytope Theorem provides a description of the perfect matchings by linear inequalities. This theorem and Theorem 2.5 can each be proved from the other.

Theorem 2.6 (Perfect Matching Polytope Theorem) *The convex hull of the set of incidence vectors of perfect matchings of G is the set of solutions of*

$$\begin{array}{rcll} x(\delta(v)) & = & 1 & (v \in V) \\ x(\delta(S)) & \geq & 1 & (S \subseteq V,\ |S|\ \mathit{odd}) \\ x & \geq & 0. & \end{array} \tag{1}$$

There are important extensions of matching theory that are nevertheless essentially equivalent. A *T-join* is a subset of edges having the property that the number of them incident to each vertex of G is odd precisely if the vertex is in T. The *Chinese postman problem* in a connected graph is to find a minimum-cost closed path covering all edges. It can be reduced to the problem of finding a minimum-weight T-join where T is the set of odd degree vertices of G. Given G, $b \in \mathbf{Z}_+^V$, $u \in (\mathbf{Z}_+ \cup \{\infty\})^E$, and $c \in \mathbf{R}^E$, the *u-capacitated b-matching problem* is to find $x \in \mathbf{Z}^E$ such that $x(\delta(v)) = b_v$ for all $v \in V$, $0 \leq x \leq u$, and $c^T x$ is minimized. These problems can be solved using algorithms for optimal matching as the essential ingredient. Moreover, the corresponding polyhedral descriptions can be proved from those for matching. See Cook, *et al.* [5].

Matching Algorithms

It seems still to be true that implementations of Edmonds' algorithm provide the fastest way, in theory and practice, to solve matching problems. However, several other algorithms, each interesting from some point of view, have been introduced. Often they have provided techniques that could be generalized when the augmenting path approach seemed difficult to extend.

An algorithm for finding a maximum matching, or more generally, a maximum-weight matching, can be based on the equivalence of separation and optimization [24]. It is most convenient to describe this for the problem of finding an optimal perfect matching. It is enough to show that the problem: "Given $x \in \mathbf{R}^E$, find if one exists, an inequality from (1) violated by x" can be solved in polynomial time. The only inequalities for which testing violation is nontrivial, are the ones of the form $x(\delta(S)) \geq 1$. It is enough, therefore, to have an efficient algorithm to find a minimum weight "odd" cut with respect to given non-negative weights. Padberg and Rao [35] showed that one can be found by first finding a minimum cut $\delta(S)$. If $|S|$ is odd then clearly $\delta(S)$ is a minimum odd cut, and we we are done. If not, one can show that the problem reduces to one of finding a minimum odd cut in each of two smaller graphs (obtained by shrinking S and its complement to single vertices). We remark that the resulting algorithm for finding an optimal perfect matching is based on the ellipsoid method, and is not at all practical.

Lovász and Plummer [33] gave a new algorithm, based on the Gallai-Edmonds partition, for finding a matching of maximum cardinality. (It is actually a specialization of Lovász's algorithm for the linear matroid parity problem, which is introduced below.) It works as follows. At step i we have a list L_i of at most $|V|$ matchings of size k_i. Based on that list we define the partition A, C, D as above. (That is, we use L_i instead of the set of all maximum matchings to define D.) If the set $S = A$ and a matching M of size k_i satisfy the equation of the Tutte-Berge Formula, then M is a maximum matching. If not, then one of the conclusions of the Gallai-Edmonds Structure Theorem is violated. Any such violation leads to the discovery of a new matching M', of cardinality $k_i + 1$ or k_i. In the former case, we proceed to step $i + 1$ with $k_{i+1} = k_i + 1$, and

$L_{i+1} = \{M'\}$. In the latter case, we proceed to step $i+1$ with $k_{i+1} = k_i$, and $L_{i+1} = L_i \cup \{M'\}$. (In this case, the set D of the new partition will be larger.) Of course, we have omitted some details. These details do involve the use of augmenting-like paths, but, unlike in Edmonds' algorithm, one does not have to search for the paths—they just appear.

Lovász and Plummer also gave a variant of the Edmonds algorithm for finding a maximum weight matching. Edmonds' algorithm, while the current dual solution remains fixed, is essentially working on the subgraph consisting of edges whose corresponding dual constraints hold with equality. The dual change that it makes can depend, at least in many versions of the algorithm, on some arbitrary choices. Lovász and Plummer's variant uses only the Gallai-Edmonds structure of this subgraph and the old dual solution to define the new dual solution. It thus separates the primal-dual phase from the augmenting path phase of the Edmonds algorithm. (Another way to say it, is that any algorithm capable of finding the Gallai-Edmonds partition of a graph can be used as a subroutine in the primal-dual algorithm.) Although this idea does not lead to a more efficent algorithm, it does provide more insight. In addition, it turns out to be very important in generalizations.

Matching algorithms from the Tutte matrix

The Tutte matrix lends itself to algorithmic approaches to the maximum matching problem. We know that it is enough to compute its rank. (This will find the size of a maximum matching; actually finding such a matching takes a bit more work, which we will ignore here.) However, a straightforward approach fails. Gaussian elimination, applied directly, leads to intermediate matrices having entries that are exponentially long. Rather, we use an approach based on the idea of an *evaluation* of the Tutte matrix. This is a rational matrix obtained from the Tutte matrix by assigning a rational value to each variable z_e. It is easy to see that the rank of $A(G)$ is at most the rank (over the rationals) of any evaluation of $A(G)$, and that there exists an evaluation whose rank is equal to the rank of $A(G)$. Lovász showed that a randomly chosen evaluation of the Tutte matrix has a significant probability of being such a maximum-rank evaluation. Choosing a few such evaluations independently, one can find in polynomial time the size of a maximum matching with high probability.

Geelen [23] recently showed how the Tutte matrix could be used to give a deterministic polynomial-time algorithm. This is a beautiful result, and shows that matching theory itself remains a vital area of research. His algorithm begins with an arbitrary evaluation, and then searches locally for an improvement. One obvious idea for a local improvement is to find a variable z_e whose value can be changed to increase the rank of the evaluation. It is not quite true that an evaluation that is locally optimal in this sense will have maximum rank. However, it turns out to be sufficient to amend the definition of "improvement" just slightly. Say that a row or column of a matrix is *dependent* if its deletion does not decrease the rank. (In view of Theorem 2.2, the dependent rows of the Tutte matrix correspond exactly to the set D of the Gallai-Edmonds partition of G.) We define an evaluation to be *locally optimal* if no change to the value of a single variable either increases the rank of the evaluation, or keeps the rank the same while increasing the number of dependent rows.

Theorem 2.7 *A locally optimal evaluation of the Tutte matrix of G is a maximum-rank evaluation.*

Geelen also shows that one can restrict values for the z_e in an evaluation to the integers $\{1, 2, \ldots, |V|\}$. It is now immediate that there is a polynomial-time algorithm to find the evaluation. Namely, we try changing the value of a variable to one of $|V| - 1$ other possible values. If we get an improvement, we continue, and otherwise we have an optimal evaluation. Clearly, we need to try at most $|V|^3$ evaluations to get an improvement, at most $|V|^5$ throughout the algorithm, and the work at each step is simple linear algebra. Thus there is an almost trivial algorithm for finding the size of a maximum matching. (On the face of it, it is very inefficient. There are a number of observations that make it possible to improve the running time. Moreover, the algorithm can be refined to actually find a maximum matching. See [23].)

It is interesting to observe that three of the algorithms for finding a maximum matching that we have mentioned have a common structure. The Lovász-Plummer algorithm keeps at each step a list of matchable sets of the same size. (These are the sets induced by the matchings in the list L_i.) Geelen's algorithm also keeps such a list, implicitly; its elements correspond to the maximum-rank principal submatrices of the current evaluation. Edmonds' algorithm keeps a forest of "alternating trees" rooted at the unmatched vertices of a "shrunken graph", which also encodes such a list implicitly. The matchable sets correspond to the matchings that can be obtained by taking a maximum matching of each tree and the current matching of the vertices not in the forest, and then extending the matching to the original graph by repeatedly "expanding". Each of the algorithms tests whether optimality is reached, and if not, makes a new list with the following property. Either the new list consists of matchable sets of larger size, or the list consists of matchable sets of the same size, the union of whose complements is larger.

3 Matroids

A *matroid* on T can usually be thought of as a matrix with columns indexed by elements of T, where available knowledge of the matrix is limited to knowledge of the subsets of T that index linearly independent sets of columns. Although there do exist matroids that do not arise from matrices in this way, this fact is not very important for our purposes. However, the fact that we cannot necessarily access the matrix itself *is* important. A matroid may be defined in a number of ways, for example, via its set of independent sets, or its set of bases (maximum size independent sets), its set of circuits (minimal dependent sets), or its rank function (giving, for any subset A of T, the size $r(A)$ of a largest subset of A that is independent). Here is an axiomatic definition in terms of the set of bases—for any two bases B, B' and any element $e \in B' \backslash B$ there exists $f \in B \backslash B'$ such that $(B \cup \{e\}) \backslash \{f\}$ is also a basis. Algorithms operating on matroids typically access the matroid only through an "oracle" that, given a subset A of T, tells whether or not A is independent.

A well-known class of matroids arises from graphs—take T to be the edge set of a (connected) graph, and define a set of edges to be independent if they do not contain the edges of a simple circuit. Then the bases of the matroid are the (edge sets of) spanning trees of the graph. There is a well-known "greedy" procedure for finding a spanning tree of maximum (or minimum) weight.

Perhaps the earliest connection of matroid theory with combinatorial optimization is a result of Rado [37], that such a procedure works in general for matroids.

Greedy Algorithm

```
Order $T$ as $\{e_1, \ldots, e_m\}$, where $c_{e_1} \geq c_{e_2} \geq \ldots \geq c_{e_m}$;
Initialize $B = \emptyset$;
For $i = 1$ to $m$
    If $B \cup \{e_i\}$ is independent
        Add $e_i$ to $B$.
```

Theorem 3.1 *At termination of the Greedy Algorithm, B is a basis of maximum weight.*

Edmonds rediscovered this fact, and proved the stronger result that the incidence vector of B maximizes $c^T x$ over all x satisfying a natural set of inequalities. Thus he established a description for the convex hull of incidence vectors of bases of the matroid.

Theorem 3.2 (Matroid Basis Polytope Theorem) *The convex hull of the set of incidence vectors of bases of the matroid M on T is the set of solutions of*

$$\begin{aligned} x(A) &\leq r(A) \qquad (A \subseteq T) \\ x(T) &= r(T) \\ x &\geq 0. \end{aligned}$$

Matroid Intersection

The most important model related to matroids, is matroid intersection. We are given two matroids on the same set T, and we are interested in the existence of a common basis. So we may assume that the two matroids have the same rank, say k. Let $r_i(A)$ denote the rank of A in matroid M_i, for $i = 1$ and 2. If B is a common basis and A is any subset of T, we have

$$k = |B| = |B \cap A| + |B \cap (T \backslash A)| \leq r_1(A) + r_2(T \backslash A).$$

Edmonds [15] proved that this necessary condition is also sufficient.

Theorem 3.3 (Matroid Intersection Theorem) *If M_1, M_2 are matroids on T of rank k, they admit a common basis if and only if for every set $A \subseteq T$*

$$r_1(A) + r_2(T \backslash A) \geq k.$$

One special case of the existence problem for a common basis is the existence problem for a perfect matching in a bipartite graph G. We may assume that the parts, P_1, P_2 of the bipartition both have size k. Where T is the edge-set of the graph, we say that a set of edges is a basis in M_1 if it has exactly one edge incident with each vertex of P_1, and similarly for M_2 and P_2. Then the common bases are indeed the perfect matchings of G. Now let us apply the Matroid Intersection

Theorem to this special case. Let $A \subseteq E$, and consider the set C consisting of those vertices of P_1 incident with at least one edge in A, together with those vertices of P_2 incident with at least one edge of $E \backslash A$. Notice $|C| = r_1(A) + r_2(T \backslash A)$, and that every edge of G is incident to at least one vertex in C. So Theorem 3.3 implies that, if G has no perfect matching, then it has a "vertex cover" of size less than k. This is a form of the Kőnig-Hall Theorem for bipartite matching.

There are a number of attractive theorems that are equivalent to the Matroid Intersection Theorem. One of them was actually found earlier by Rado [36]. There are also a number of elegant proofs. Edmonds' original proof was constructive, providing an efficient augmenting path algorithm, generalizing such algorithms for bipartite matching.

Edmonds also considered the more general problem of finding a maximum weight common basis. He proved that the convex hull of common bases of two matroids is the intersection of the convex hulls of the two basis polyhedra. Moreover, he showed that the system consisting of the linear descriptions for the two basis polyhedra, is totally dual integral. (This can be used to prove Theorem 3.3.) He gave an elegant nonconstructive proof of the polyhedral theorem and the total dual integrality. He also gave an algorithmic proof, based on a primal-dual approach. Like the maximum-weight matching algorithm, it uses the linear description and the augmenting-path algorithm for the unweighted case.

4 Extensions

In the early years the striking similarity of the results—existence theorems, efficient algorithms, polyhedral descriptions—for matching and matroid intersection suggested to many the existence of a nice, solvable common generalization. Since one could define a common generalization to be simply the union of the two models, the "nice" qualifier is important.

Many of the most useful extensions of one of matching or matroid intersection, do not seem to bear any relation to the other one. However, we will describe in this paper three that do, and describe some of their advantages and limitations. The reader (or time) will decide how "nice" each of them is. The generalizations are matroid parity, jump systems, and path-matching. Since the first one is relatively old, we discuss it only briefly here. Then we devote the next two sections to the other two models.

We are given a matroid M on T and a pairing of the elements of T. A *parity set* is a subset of T that contains either both or neither of the elements of each pair. We are interested in parity bases—bases that are also parity sets. Consider the special case in which we are given a graph $G = (V, E)$; it is convenient to assume that G has no isolated vertices. We take $T = \{(v, e) : v \in V, e \in E, v \text{ incident with } e\}$. The pairing simply pairs (v, e) and (w, e), where e joins v to w. The matroid M has as bases the subsets of T of size $|V|$ such that every vertex occurs exactly once. Then the parity bases correspond to the perfect matchings of G. As a second special case, suppose that we have matroids M_1, M_2 on S. We make disjoint copies S', S'' of S and form the matroid on $T = S' \cup S''$ whose bases consist of the union of a subset of S' corresponding to a basis of M_1 with a subset of S'' corresponding to a basis of M_2. The pairing simply pairs two elements of T that are copies of the same element of S. Then a parity basis corresponds to a common basis of M_1, M_2. So matroid parity is indeed a common generalization of matching and matroid intersection.

The problem of determining whether a parity basis exists is unsolvable in general [30]. (That is, there is no polynomial-time algorithm to solve it, assuming that the algorithm is allowed to access the matroid only by calling an independence-testing oracle.) In addition, it contains $\mathcal{NP}$-hard special cases. Nevertheless, there are some deep results. First, Lovász gave an existence theorem and an efficient algorithm for the case where M arises from a (given) matrix. This "linear matroid parity problem" does not quite give a solvable common generalization of matching and matroid intersection—it includes the case of matroid intersection only where the two matroids arise from (given) matrices over the same field. (The weighted version of the linear matroid parity problem remains open.) Second, there are results for the general problem that lead to solutions for other important special cases [30].

5 Jump Systems

In this section we let $V = \{1, \ldots, n\}$. For $x, y \in \mathbf{Z}^V$ we define $[x, y]$ to be $\{x' \in \mathbf{Z}^V : \min(x_i, y_i) \leq x'_i \leq \max(x_i, y_i), i \in V\}$. We call $[x, y]$ a (bounded) *box*. (More generally, a *box* is a Cartesian product of intervals in Z, where the intervals are possibly infinite.)

We define $d(x, y)$ to be $\sum(|x_i - y_i| : i \in V)$, and, for subsets A, B of $\mathbf{Z}^V$, $d(A, B)$ to be $\min(d(x, y) : x \in A, y \in B)$. A point $x' \in \mathbf{Z}^V$ is a *step from x to y* (or an *(x, y)-step*) if $x' \in [x, y]$ and $d(x, x') = 1$. Let $\mathcal{J}$ be a nonempty subset of $\mathbf{Z}^V$; a point $x \in \mathcal{J}$ is called a *feasible* point. The set $\mathcal{J}$ is a *jump system* if it satisfies the following axiom:

two-step axiom Given feasible points x, y and a step x' from x to y, then either x' is feasible, or there exists a feasible step x'' from x' to y.

For convenience in this paper, we will assume that the jump systems are also bounded. However, the results—sometimes with slight modifications—hold without this assumption.

Examples in $\mathbf{Z}^2$ are useful for understanding the definition; see Figure 1, where solid points are the feasible ones. The example on the left is a jump system, and that on the right is not—the two-step axiom fails for the indicated points.

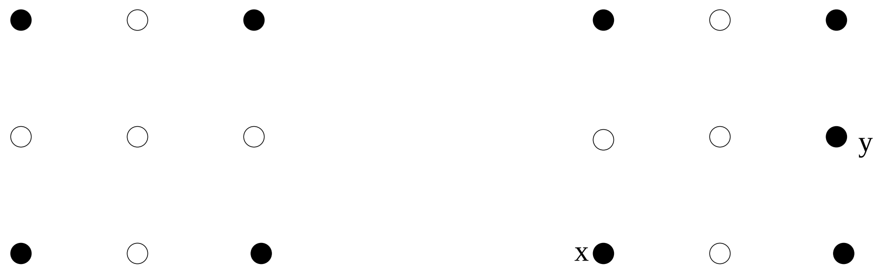

Figure 1: A jump system and a set that is not a jump system

Examples and Constructions

Jump systems were introduced by Bouchet and Cunningham [3]. Except where mentioned explicitly, the following examples and basic results are from that paper. Here are some examples of jump systems.

- **Low-dimensional jump systems.** If $n = 1$, $\mathcal{J}$ is a jump system if and only if there do not exist two feasible points such that between them, there is no feasible point and two or more nonfeasible points. If $n = 2$, a characterization is more involved. See [22].

- **Matroids and delta-matroids**. The jump systems contained in $\{0,1\}^V$ are called *delta-matroids.* These were introduced earlier [1], [4], [10], and have many attractive properties. However, we will not go into them here. Those that are also *constant sum*, that is, have the property that each feasible point has the same coordinate sum, are equivalent to matroids. That is, they are exactly the ones whose feasible points are the incidence vectors of bases of a matroid.

- **Degree systems.** This is perhaps the most fundamental example. Let $G = (V, E)$ be a graph. For a spanning subgraph H of G, we define the *degree sequence of* H, to be the vector $\deg_H \in \mathbf{Z}^V$ such that, for $v \in V$, $\deg_H(v)$ is the degree of vertex v in H. The set of degree sequences of spanning subgraphs of G is called the *degree system* of G.

We list a few ways to construct jump systems from others. In the first few examples, it is obvious that we get a jump system, but in some of the later ones it is not.

◇ **Translation.** For an integral vector b, add b to every feasible point.

◇ **Reflection.** For a coordinate i, replace x_i by $-x_i$ for every feasible point x.

◇ **Intersection with a Box.** Given a box B, $\mathcal{J} \cap B$ is a jump system, if it is non-empty.

◇ **Projection.** Given a set $S \subseteq V$, replace each feasible point by its restriction to S (and delete duplicates).

◇ **Sum.** This is perhaps the most important operation on jump systems. If $\mathcal{J}_1$ and $\mathcal{J}_2$ are jump systems, then $\mathcal{J}_1 + \mathcal{J}_2 = \{x + y : x \in \mathcal{J}_1,\ y \in \mathcal{J}_2\}$ is also a jump system.

◇ **Closest Points to a Box.** Given a box B, $\mathcal{J}_B = \{x \in \mathcal{J} : d(x, B) = d(\mathcal{J}, B)\}$ is a jump system [32].

◇ **Faces.** Let F be a non-empty face of $conv(\mathcal{J})$. Then $\mathcal{J} \cap F$ is a jump system [32].

Here are some examples using the above ideas.

- If we intersect the degree system of a graph G with the unit cube, the feasible points are the incidence vectors of matchable sets of G. This is a fundamental class of delta-matroids.

- The degree system of a graph is the sum of the degree systems of its one-edge spanning subgraphs. Since these are trivially jump systems, this proves that degree systems are indeed jump systems.

- If we begin with a matroid, reflect it in all coordinates, and then translate it by the vector of all 1's, we get another jump system in the unit cube that is constant sum, that is, we get another matroid. Its bases are the complements of the bases of the given matroid M; that is, it is the *dual* of M.

- A *minor* of $\mathcal{J}$ is obtained by taking, for some $S \subseteq V$ and some $y \in \mathbf{Z}^{V \setminus S}$, $\mathcal{J}' = \{x \in \mathbf{Z}^S : (x, y) \in \mathcal{J}\}$. It is a jump system, because it is a projection followed by intersection with a box.

- **(Homomorphism)** Given $\mathcal{J} \subseteq \mathbf{Z}^V$, form $\mathcal{J}' \subseteq \mathbf{Z}^{(V \setminus \{n\})}$ by $\mathcal{J}' = \{(x_1, \ldots, x_{n-2}, x_{n-1} + x_n) : x \in \mathcal{J}\}$. We can see that $\mathcal{J}'$ is a jump system as follows. First, extend $\mathcal{J}$ to $\mathbf{Z}^{n+1}$ by defining $\mathcal{J}_1 = \{(x, 0) : x \in \mathcal{J}\}$. Now define $\mathcal{J}_2 \subseteq \mathbf{Z}^{n+1}$ by $\mathcal{J}_2 = \{(0, \ldots, 0, -y_{n-1}, -y_n, y_{n-1} + y_n) : y_{n-1}, y_n \in \mathbf{Z}\}$. It is easy to check that $\mathcal{J}_2$ is a jump system. Now take $\mathcal{J}'' = \mathcal{J}_1 + \mathcal{J}_2$, and then take the minor consisting of those points $z \in \mathcal{J}''$ such that $z_{n-1} = z_n = 0$. This is $\mathcal{J}'$ (essentially).

Basic Results

The following *nested box lemma* of Lovász [32], is an important tool in the proof of many results in [32]. (Actually, as observed by Geelen [22], one can take this as the definition of a jump system.)

Theorem 5.1 *If $\mathcal{J}$ is a a jump system and $B^1 \subseteq \ldots \subseteq B^r$ are boxes, then $\mathcal{J}_{B^1} \cap \ldots \cap \mathcal{J}_{B^r} \neq \emptyset$.*

One application of this lemma is the validity of a greedy algorithm. We state the algorithm and refer to Geelen [22] for the proof. Suppose that we are interested in maximizing $c^T x$ over feasible points x of a bounded jump system $\mathcal{J}$. By reflection, we may assume that $c \geq 0$.

Greedy Algorithm for Jump Systems

```
Order $V$ as $\{j_1, \ldots, j_n\}$, where $c_{j_1} \geq c_{j_2} \geq \ldots \geq c_{j_k} > 0 = c_{j_{k+1}} = \cdots = c_{j_n}$;
Initialize $\mathcal{J}^0 = \mathcal{J}$;
For $i = 1$ to $k$
    Set $\alpha = \max(x_{e_i} : x \in \mathcal{J}^{i-1})$;
    Set $\mathcal{J}^i = \{x \in \mathcal{J}^{i-1} : x_{e_i} = \alpha\}$.
```

Theorem 5.2 *Each point $x \in \mathcal{J}^k$ maximizes $c^T x$ over $\mathcal{J}$.*

Note that the set $\mathcal{J}^k$ is a face of $\mathcal{J}$; we call it a *greedy face* of $\mathcal{J}$. (It is also a minor of $\mathcal{J}$.)

It is interesting to consider the relationship between this greedy algorithm and the one for matroids. If the c_j were initially non-negative, then this algorithm does indeed yield the one for matroids. And of course, we could (taking advantage of the fact that matroids are constant-sum) make the c_j non-negative by adding a constant to all of them, and again the above algorithm would reduce to the one for matroids. However, if we use the reflection method before applying the jump system greedy algorithm, the resulting matroid algorithm is different—it involves considering the elements in decreasing order of their absolute value.

Jump systems and restricted ≤ 2-factors

Many of the known algorithms for matching and generalizations are based on augmenting paths. Often, however, an augmenting path theorem—roughly, the fact that any non-maximum matching X can be transformed into a larger one using a path—can be proved more easily in a nonconstructive way than by formulating an algorithm. (We are using the term "matching" here, though these remarks also apply to more general objects.) Indeed, there are settings where augmenting path theorems have been found, but augmenting path algorithms are not known. Nonconstructive proofs use a particular larger matching Y together with X to prove the existence of a path. The path transforms X into a matching X'' of size $|X|+1$. (In general $X'' \neq Y$.) Its degree sequence differs from that of X only at the ends, u and v, of the path. Suppose that we consider the degree sequences x of X and y of Y and take x' to be obtained from x by increasing x_u by 1, and x'' to be the degree sequence of X''. Then there is a good deal of similarity to the two-step axiom. For a class $\mathcal{F}$ of subgraphs of G, we are interested in algorithmic solvability of problems involving $\mathcal{F}$. Possibly this issue is related to two others—the existence of an augmenting path theorem for $\mathcal{F}$, and whether the set of degree sequences of members of $\mathcal{F}$ is a jump system. Here we explore this idea in some detail for some choices of $\mathcal{F}$ related to 2-factors.

A *2-factor* (respectively, $\leq$ *2-factor*) of a graph $G=(V,E)$ is a set $X \subseteq E$ such that every vertex of G is incident with exactly (respectively, at most) two edges of X. For this subsection, we will use the term *factor* to refer to ≤ 2-factors. For a positive integer k, a factor X is *k-restricted*, or simply *restricted*, if every circuit formed by edges of X has length at least $k+1$. If $k \leq 2$, then this is no restriction at all. It is well-known that questions about factors can be solved by reduction to ordinary matchings—see [5]. (They can also be solved directly, by generalizing matching techniques.) On the other hand, if $|V|-1 \geq k \geq \frac{|V|}{2}$, then a restricted factor can contain only circuits that are hamiltonian, so deciding the existence of a k-restricted 2-factor is hard in general. In fact, for most other values of k, questions about restricted factors are also hard. For $k \geq 5$ the problem of finding a largest restricted factor has been proved to be $\mathcal{NP}$-hard by Papadimitriou; see [6]. Hell, Kirkpatrick, Kratchovil, and Kriz [27] proved a stronger result, that if the set of circuit lengths to be excluded is not a subset of $\{3,4\}$, then the problem is $\mathcal{NP}$-hard.

On the positive side, Hartvigsen [25] gave an efficient (but very complicated) algorithm for the case when $k=3$. As yet, no clean statement of a min-max theorem for this case is known. The case $k=4$ remains open, but there are some reasons for optimism. One reason is that there exists an augmenting path theorem—see Russell [38]. (In fact, [38] identifies exactly the sets of excluded circuit lengths for which there exists an augmenting path theorem. One warning—there do exist $\mathcal{NP}$-hard problems for which augmenting-path theorems exist; an example is the problem where the only *allowed* circuit length is 3.) Another reason is that there is a nice min-max theorem for the case when G is bipartite, due to Kiraly [28]. Also Frank [17] observed that this theorem could be derived from the very general theory of Frank and Jordan [18]. Neither of these proofs is algorithmic, but Hartvigsen [26] has outlined an algorithm.

There is also the more general problem, where there are weights on the edges and the goal is to find a maximum-weight factor. Of course, this problem is efficiently solvable for $k \leq 2$ and $\mathcal{NP}$-hard for $k \geq 5$, so the dividing line between tractability and hardness is near that for the unweighted version. However, the case $k=4$ of the weighted problem is already $\mathcal{NP}$-hard, even when G is bipartite. (See Vornberger [42].) Moreover, the case $k = 3$ remains open (although it is solved in the unweighted case), and there is evidence that the corresponding polytope is complicated.

To make the link with jump systems, we ask the question, "For what values of k does the set $G(k)$ of degree sequences of restricted factors of any graph G form a jump system?" The following result gives some evidence of a connection between the solvability of questions about restricted factors and the existence of associated jump systems.

Theorem 5.3 *For any graph G and any $k \leq 3$, $G(k)$ is a jump system. For any $k > 4$ there exists a graph G such that $G(k)$ is not a jump system.*

Proof For $k \leq 2$, since the restricted factors are just the factors, $G(k)$ is the intersection of the degree system of G with a box, and so is indeed a jump system.

For $k = 5$, consider the graph G of Figure 2. (This example is derived from a gadget used in an $\mathcal{NP}$-completeness proof of [27].) It is easy to see that there is a circuit of G of length 9 missing vertex u, and also such a circuit missing v. Therefore, the vectors x and y are both in $G(k)$, where $x_u = 0 = y_v$ and $x_w = 2$ for all $w \neq u$ and $y_w = 2$ for all $w \neq v$. Suppose that $G(k)$ is a jump system. Then one of the vectors z defined by $z_w = 2$ for all w, or z' defined by $z'_u = z'_v = 1$ and $z'_w = 2$ for all other w, is in $G(k)$. But we are not allowed to use circuits of length 5 or less, and G has just 10 vertices. Therefore, $z \in G(k)$ implies that G has a hamiltonian circuit, whereas $z' \in G(k)$ implies that there is a path from u to v, either of length 9 or less than 4. It is easy to see that none of these possibilities holds, so $G(k)$ is not a jump system. An easy modification to this example (inserting some additional vertices of degree 2) shows that for any larger value of k, we again do not get a jump system.

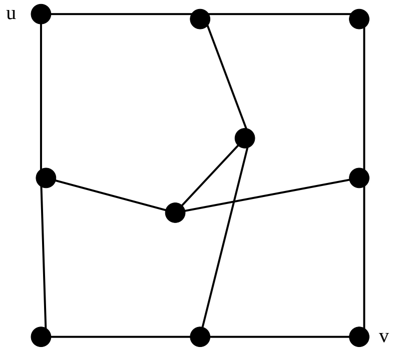

Figure 2: G(5) is not a jump system

Now consider the case when $k = 3$. We refer to circuits of length three as *triangles*. We denote $G(3)$ by $\mathcal{J}$. Let $x, y \in \mathcal{J}$, let x' be a step from x to y, and let u be the component on which x' differs from x. Suppose first that $x_u < y_u$. Then x' is obtained from x by increasing x_u by 1. Of course, $x' \notin \mathcal{J}$, since it has odd component sum. Therefore, we seek a step x'' from x' to y such that $x'' \in \mathcal{J}$. Choose restricted factors X, Y having degree sequence x, y, respectively. We show that there exists an edge-simple path P from u such that $X'' = X \triangle E(P)$ satisfies the properties required of x'' above.

Consider a path $v_0, v_1, \ldots, v_m$ from u to some vertex v. For $i = 0, 1, \ldots, m$, we use P_i to denote the path $v_0, v_1, \ldots, v_i$, and we use X_i to denote $X \triangle E(P_i)$. We require that the path P_m satisfy:

(a) $v_i v_{i+1} \in Y \backslash (X \cup E(P_i))$ for i even

(b) $v_i v_{i+1} \in X \backslash (Y \cup E(P_i))$ for i odd

(c) X_m is triangle-free.

Obviously, the path P_0 satisfies these conditions. We will show how any such path P_m that does not satisfy the requirements for P above, can be extended to a longer path satisfying (a), (b), and (c). Let us write the degree of a vertex w in the subgraph with edge-set X_m as $\deg'(w)$.

Suppose first that m is odd. If the degree sequence of X_m is a step from x' to y, then P_m is the required path P. If not, then $\deg'(v) = x_v + 1 > y_v$, and we conclude that there exists an edge $vq \in X \backslash (Y \cup E(P_m))$. We extend P_m by putting $v_{m+1} = q$.

Now suppose that m is even. If the degree sequence of X_m is a step from x' to y, then P_m is the required path P. If not, then $\deg'(v) = x_v - 1 < y_v$, and we conclude that there exists an edge $vq \in Y \backslash (X \cup E(P_m))$. If $X_m \cup \{vq\}$ is triangle-free, then we can extend P_m by putting $v_{m+1} = q$. So suppose that X_m contains edges qw, wv for some vertex w. If $qw \in X \backslash E(P_m)$, then we can extend P_m by putting $v_{m+1} = q$ and $v_{m+2} = w$. Otherwise, we have $qw \in E(P_m) \backslash X$ (and therefore $qw \in Y$). Then since Y is triangle-free, we must have $vw \in X \backslash (E(P_m) \cup Y)$. Now $\deg'(v) = x_v - 1 = 1$, whereas $y_v = 2$. Therefore, there exists an edge $vp \neq vq$ in $Y \backslash X$. Suppose that $X_m \cup \{vp\}$ contains a triangle. Then the triangle must have vertices v, p, w. But this would imply that $\deg'(w) = 3$, a contradiction. Therefore, we can extend P_m by putting $v_{m+1} = p$.

Since the path P_m is edge-simple, and can be extended as long as it does not have the properties required of P, we must eventually find such a path P.

This completes the proof for $k = 3$ when $x_u < y_u$. However, the situation where $x_u > y_u$ can be reduced to this one. Add a new vertex u' and a new edge uu' to G and put the edge into Y. Now $x_{u'} < y_{u'}$. Apply the previous result, to get X''. Necessarily, $uu' \in X''$. Deleting it, we get a restricted factor of the original graph with the required properties. ∎

The above theorem does not address the case $k = 4$. It seems that a similar, but more complicated, approach can be used to prove that $G(4)$ is a jump system for all graphs G. If so, then the values of k not having the jump system property are precisely those for which the existence problem for restricted factors is known to be $\mathcal{NP}$-hard. I also expect (but do not know how to prove) that the latter problem is solvable in polynomial time for $k = 4$. Note that the paths that arise in this case (either to prove that $G(4)$ is a jump system or to use as augmenting paths in a possible algorithm) need not be edge-simple. An example to show that it may be necessary to traverse an edge twice is shown in Figure 3. Here X is the factor consisting of the thick edges, and Y is any larger restricted factor.

The Membership Problem

The membership problem for a jump system $\mathcal{J}$ is "Given $x \in \mathbf{Z}^V$, is x feasible?" Here are some examples. Suppose that $\mathcal{J}$ is the degree system of a graph G. Then the question amounts

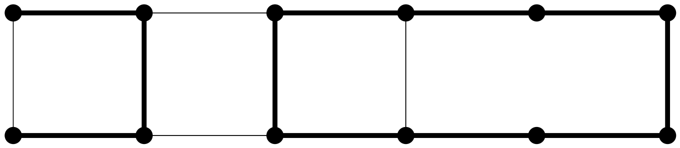

Figure 3: Edge-simple paths are not enough

to whether there exists a subgraph of G with prescribed degrees at the vertices, the existence problem for 'degree-constrained subgraphs'. Of course, the perfect matching existence problem is a special case.

Our next two examples use the fact that, for jump systems $\mathcal{J}_1, \mathcal{J}_2$, the set $\mathcal{J}_1 - \mathcal{J}_2 = \{x - y : x \in \mathcal{J}_1, y \in \mathcal{J}_2\}$ is also a jump system. (It is the sum of two jump systems, the second one obtained by reflection of $\mathcal{J}_2$ in all coordinates.) Then $0 \in \mathcal{J}_1 - \mathcal{J}_2$ if and only if $\mathcal{J}_1, \mathcal{J}_2$ have a common point. So the "intersection problem" for two jump systems reduces to the membership problem for their "difference".

This implies, since matroids are jump systems, that the membership problem includes the matroid intersection (existence) problem as a special case. On the other hand, suppose that we take $\mathcal{J}_1$ to be a matroid and $\mathcal{J}_2$ to consist of the set of incidence vectors of the matchable sets of G, where the edges of G consist of a perfect matching. Then the points of $\mathcal{J}_2$ are the parity sets of V with respect to the pairing determined by the edges of G. Therefore, a common point of $\mathcal{J}_1, \mathcal{J}_2$, will correspond to a parity basis.

These examples suggest some remarks. One is that the membership problem is, in general, hard, since it includes matroid parity as a special case. Another is that the device used to reduce intersection to membership should be used with caution. For one thing, the difficulty of a problem on jump systems may depend on how the system is given to us. There seems as yet to be no agreement on standard assumptions about how this is to be done. Moreover, if one considers, not just the existence of a common point of two jump systems, but the problem of finding a best common feasible point, the reduction to their difference seems useless. So, for example, there are as yet no results on generalizing the weighted matroid intersection problem to jump systems.

If a point $\hat{x}$ is not feasible, we need to be able to demonstrate this fact. It may be that $\hat{x}$ is not in the convex hull of $\mathcal{J}$, in which case there is an inequality $a^T x \leq b$ satisfied by every point of $\mathcal{J}$ but violated by $\hat{x}$. On the other hand, it may be that $\hat{x}$ is a *gap* of $\mathcal{J}$—it is an integral but not feasible point in the convex hull of $\mathcal{J}$. We call a jump system *convex* if it has no gaps. Many important classes of jump systems are convex (more on this below), so the first case is important by itself. Note that the degree system of a graph has gaps—take any integral vector with odd component-sum that is in the convex hull of the system. (In fact, a theorem of Koren [29] implies that these are the only gaps. However, if we intersect the degree system with a box—that is, put upper and lower bounds on the degrees—the resulting jump system has more interesting gaps.)

To understand the situation when $\hat{x}$ is not in the convex hull of $\mathcal{J}$, it is useful to have a linear-inequality description of the convex hull of a jump system. Let us define, for disjoint subsets A, B of V, $f(A, B)$ to be $\max(x(A) - x(B) : x \in \mathcal{J})$. Then f has the following property, called *bisubmodularity*:

$$f(A, B) + f(A', B') \geq f(A \cap A', B \cap B') + f((A \cup A') \backslash (B \cup B'), (B \cup B') \backslash (A \cup A')) \qquad (2)$$

An *integral bisubmodular polyhedron* is a polyhedron of the form $\{x \in \mathbf{R}^V : x(A) - x(B) \leq f(A,B)$ for all disjoint $A, B \subseteq V\}$, where f is an integral bisubmodular function. Bisubmodular polyhedra generalize polymatroids, submodular polyhedra, generalized polymatroids, and other classes. This class of polyhedra was first considered by Dunstan and Welsh [12], but without the connection to bisubmodularity. The connection with jump systems is given by the following result from [3].

Theorem 5.4 *The convex hull of a jump system is an integral bisubmodular polyhedron. The set of integral points in any integral bisubmodular polyhedron is a (convex) jump system.*

Corollary 5.5 *Let $\mathcal{J}$ be a jump system. Then the convex hull of $\mathcal{J}$ is the solution set of a set of inequalities with* $0, 1, -1$ *coefficients.*

We summarize the approach of Lovász [32] to the membership problem. First, we generalize the problem to the following. For a jump system $\mathcal{J}$ and a box B, find $d(\mathcal{J}, B)$. (In the special case where $B = \{x\}$, this minimum distance is 0 if and only if $x \in \mathcal{J}$). Here is a rather trivial lower bound. Let $w \in \{0, 1, -1\}^V$. Then

$$d(\mathcal{J}, B) \geq \min_{x \in \mathcal{J}} w^T x - \max_{x \in B} w^T x. \tag{3}$$

Cases where there exists a w for which equality holds in (3), provide most of our positive results on the membership problem. One case in which this is true, is when B is “fat”.

Theorem 5.6 *Let $B = [a, b]$ be a box with $a_i \neq b_i$ for all i. Then there exists $w \in \{0, 1, -1\}^V$ giving equality in (3).*

Corollary 5.7 *If $B = [a, b]$ is a box with $a_i \neq b_i$ for all i and $\mathcal{J}$ is a jump system such that $B \cap \mathcal{J} = \emptyset$, then there exists $w \in \{0, 1, -1\}^V$, and $w_0 \in \mathbf{Z}$ such that $w^T x \leq w_0$ for all $x \in \mathcal{J}$, and $w^T x > w_0$ for all $x \in B$.*

Notice that, while the convex hull of a jump system can contain an integral point that is not feasible, the corollary implies that it cannot contain a box with no feasible point that has non-zero width in every direction. In fact, it cannot even intersect such a box.

This last result seems to say nothing about the original motivating case, that is, when B is a singleton. But, by a trick, we can apply it to the membership problem for constant-sum jump systems. Suppose that $\mathcal{J}$ is a jump system for which every feasible point x satisfies $\sum(x_j : j \in V) = \alpha$. If we wish to determine whether y is feasible, we first check that $\sum(y_j : j \in V) = \alpha$. Now y is feasible if and only if $B \cap \mathcal{J} \neq \emptyset$, where $B = \{x \in \mathbf{Z}^V : y \leq x\}$. Therefore, if y is not feasible, we get w and w_0 as in the corollary. Clearly, we cannot have any $w_j = -1$, so there is a set $A \subseteq V$ such that every feasible point x satisfies $x(A) \leq w_0$, while $y(A) > w_0$.

The latter result is already strong enough to imply the Matroid Intersection Theorem. Namely, if $\mathcal{J}$ is the difference of two matroids, each having rank k, and having rank functions r_1, r_2, then $\mathcal{J}$ has constant sum with $\alpha = 0$, and the question whether there is a common basis, is the question whether $y = 0$ is in $\mathcal{J}$. Therefore, if there is no common basis, we have A as above, and since we can choose $w_0 = -1$, we have, for all $x \in \mathcal{J}$, $x(A) < 0$. That is, for all bases B_1 of the first matroid and B_2 of the second,

$$|B_1 \cap A| < |B_2 \cap A| = k - |B_2 \cap (V \backslash A)|.$$

This gives the Matroid Intersection Theorem.

Lovász provides a deeper application of (3), as follows. Let us consider the problem of finding the minimum distance from $\mathcal{J}$ to 0. Define $\mathcal{J}$ to be *critical* if for every $j \in V$ and every $\sigma \in \{1, -1\}$, $d(\sigma e_j, \mathcal{J}) < d(0, \mathcal{J})$. ($e_j$ is the integral unit vector corresponding to $j \in V$.) In other words, *every* step from 0 is closer to $\mathcal{J}$ than is 0. Before describing the result, we provide a bit of motivation, by showing that, if $\mathcal{J}$ is the degree system of a connected graph G, translated by subtracting the vector of 1's, then $\mathcal{J}$ is critical if and only if G is critical in the sense defined in Section 2. Suppose first that G is critical. Then clearly the distance from 0 to $\mathcal{J}$ is 1, so we need to show that, for every vertex j of G and every $\sigma \in \{1, -1\}$, $\sigma e_j \in \mathcal{J}$, that is, that $(1, \ldots, 1) + \sigma e_j$ is the degree sequence of a subgraph of G. Since G has a matching missing only j, the case when $\sigma = -1$ is immediate. Now choose an edge jk of G. There is a matching of G missing only k, and adding jk to the matching gives a subgraph with degree-sequence equal to e_j plus the vector of 1's, handling the other case. Now suppose that $\mathcal{J}$ is critical. Fix $j \in V$. Since $-e_j$ is closer to $\mathcal{J}$ than is 0, there is a subgraph H of G whose degree-sequence is as close to the all 1's vector as possible and such that j has degree 0. By repeatedly deleting from H an edge at a vertex of degree more than 2, we can convert H to a maximum matching leaving j exposed. Therefore, in the Gallai-Edmonds partition, $D = V$, and, since G is connected, G is critical.

Theorem 5.8 *Let $\mathcal{J}$ be a jump system, let $w \in \{0, 1, -1\}^V$, and let $S \subseteq V$ with $w_j = 0$ for all $j \in S$. Let F be the greedy face of $\mathcal{J}$ maximizing $w^T x$, let w_0 be the optimal value, and let F_S be the projection of F onto S. Then*

$$d(\mathcal{J}, 0) \geq d(F_S, 0) - w_0.$$

Moreover, there exist S and w such that equality holds, and F_S is critical.

This result provides a reduction of the minimum distance problem to the special case of critical jump systems. In some cases the reduction is well behaved, and it is possible to obtain a good characterization of the minimum distance. For example, in the case of degree systems, the critical systems that arise are also degree systems, and they are well understood. Lovász gives a min-max theorem for a class of jump systems that includes degree systems and differences of matroids. (In particular, both the existence theorem for degree-constrained subgraphs and the Matroid Intersection Theorem can be derived.) There may be room to go further in this direction.

6 Path-Matchings

Let $G = (V, E)$ be a graph and T_1, T_2 disjoint stable sets of G, that is, sets of mutually nonadjacent vertices, with $|T_1| = |T_2| = k$. We denote $V \backslash (T_1 \cup T_2)$ by R. A *perfect path-matching* is a collection of k vertex-disjoint paths, all of whose internal vertices are in R, linking T_1 to T_2, together with a perfect matching of the vertices of R not in any of the paths. Figure 4 shows an example—the

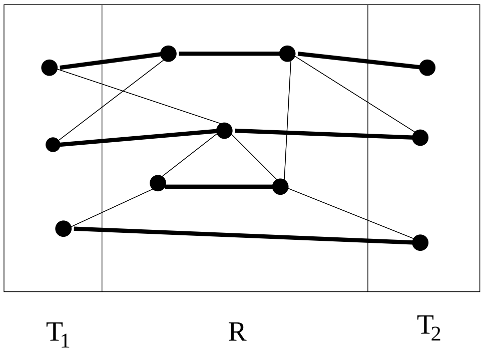

Figure 4: A perfect path-matching

thick edges form a perfect path-matching. In the special case when $R = V$, a perfect path-matching is nothing but a perfect matching of G. In the special case when $R = \emptyset$, then G is bipartite, and again, a perfect path-matching is a perfect matching of G. Therefore, this model contains bipartite matching in two different ways. As we shall see, this fact is related to the existence of a further generalization that includes matroid intersection.

We first describe a condition for the existence of a perfect path-matching. A pair of subsets $D_1 \subseteq T_1 \cup R$, $D_2 \subseteq T_2 \cup R$ is called *stable* if no edge of G joins a vertex in $D_1 \backslash D_2$ to a vertex in D_2 or a vertex in $D_2 \backslash D_1$ to a vertex in D_1. The vertices contained in the ellipses of Figure 5 form a stable pair.

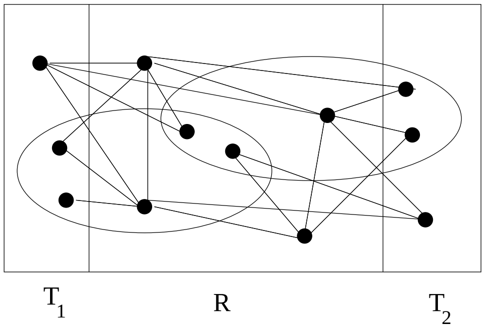

Figure 5: A stable pair

Suppose that there exists a perfect path-matching K, and let (D_1, D_2) be a stable pair. We think of the paths of K as being *from* T_1 *to* T_2. There are at least $k - |T_1 \backslash D_1|$ paths of K beginning in $D_1 \cap T_1$. Each of them eventually leaves D_1; consider its first vertex not in D_1. Since (D_1, D_2) is stable, that vertex must be in $(R \cup T_2) \backslash (D_1 \cup D_2)$. Also, for each odd component H of $G[D_1 \cap D_2]$, either an edge of a path of K leaves H or a matching edge of K leaves H. In either case the other end of this edge is again in $(R \cup T_2) \backslash (D_1 \cup D_2)$. Now we have identified at least $k - |T_1 \backslash D_1| + \text{odd}(G[D_1 \cap D_2])$ vertices of $(R \cup T_2) \backslash (D_1 \cup D_2)$, and all of them must be distinct. Therefore,

$$k - |T_1 \backslash D_1| + \text{odd}(G[D_1 \cap D_2]) \leq |(R \cup T_2) \backslash (D_1 \cup D_2)|. \tag{4}$$

The stable pair of Figure 5 violates (4), and hence no perfect path-matching exists in that example. The condition (4) is also sufficient, as the following existence theorem [7] shows.

Theorem 6.1 *(G, T_1, T_2) admits a perfect path-matching if and only if for every stable pair (D_1, D_2) we have*

$$\mathrm{odd}(G[D_1 \cap D_2]) \leq |V \backslash (D_1 \cup D_2)| - k.$$

Let us show how Tutte's Matching Theorem follows from this result. The nontrivial part is to show that, if G has no perfect matching, then there is a set S violating the condition of the theorem. Applying Theorem 6.1 with $R = V$ (and so $k = 0$), there exists a stable pair (D_1, D_2) such that

$$|V \backslash (D_1 \cup D_2)| < \mathrm{odd}(G[D_1 \cap D_2]).$$

Now observe that, because (D_1, D_2) is stable, every odd component of $G[D_1 \cap D_2]$ is also an odd component of $G[D_1 \cup D_2]$. Therefore, $\mathrm{odd}(G[D_1 \cap D_2]) \leq \mathrm{odd}(G[D_1 \cup D_2])$. If we take $S = V \backslash (D_1 \cup D_2)$, it follows that $\mathrm{odd}(G[V \backslash S]) > |S|$, as required.

Optimal path-matchings and polyhedral descriptions

An attempt to generalize the problem of finding a perfect path-matching to that of finding one of maximum weight leads to the following difficulty. If we consider the weight of a perfect path-matching to be (as usual), the sum of the weights of its edges, then the problem is $\mathcal{NP}$-hard. For suppose that all edge-weights are 1, and $|T_1| = |T_2| = 1$. Then there exists a perfect path-matching of weight $|V| - 1$ if and only if G has a hamiltonian path joining T_1 to T_2. Instead, we define the weight of a perfect path-matching to be the sum of the weights of the edges of the paths plus *twice* the weights of its other edges. Notice that this choice has the nice property that it does not favour putting edges into paths over putting them into the matching, and the resulting maximum-weight problem still contains the weighted version of the perfect matching problem.

We define the *vector* of a perfect path-matching K with the above in mind, that is, we assign a component of 1 for each path edge of K, a component of 2 for each matching edge of K, and a component of 0 for each non-edge of K. Then the weight of an optimal perfect path-matching is equal to the optimal value of $c^T x$ over the convex hull of such vectors. This convex hull can be nicely described.

Theorem 6.2 (Perfect Path-Matching Polytope Theorem) *The convex hull of vectors of perfect path-matchings with respect to G, T_1, T_2 is equal to the set of all $x \in \mathbf{R}^E$ satisfying*

$$\begin{array}{rcll} x(\delta(v)) & = & 1 & (v \in V \backslash R) \\ x(\delta(v)) & = & 2 & (v \in R) \\ x(\delta(S)) & \geq & k & (T_1 \subseteq S \subseteq T_1 \cup R) \\ x(\delta(S)) & \geq & 2 & (S \subseteq R,\ |S| \ \textit{odd}) \\ x & \geq & 0. & \end{array}$$

The Perfect Matching Polytope Theorem is an easy consequence. It is also quite easy to check that the separation problem for the inequalities of Theorem 6.2 is solvable in polynomial time, and therefore that there is a polynomial-time algorithm for finding an optimal perfect path-matching. This is the way in which polynomial-time solvability of path-matching problems was first established [21].

We have found it convenient to describe the above results in terms of perfect path-matchings, which generalize perfect matchings in graphs. However, there is also a notion that generalizes matchings. We take a partition T_1, T_2, R of V as before, but we do not require that $|T_1| = |T_2|$. A *path-matching* is a subset K of edges such that every component of $G(V, K)$ is a simple path from $T_1 \cup R$ to $T_2 \cup R$, all of whose internal vertices are in R. Let us refer to the "matching edges" of K as the edges of paths of length 1 having both ends in R, and refer to the other edges of K as "path edges". If we define the "value" of path-matching K to be $|K|$ plus the number of matching edges of K, then the maximum value of a path-matching can be characterized as follows.

Theorem 6.3 *The maximum value of a path-matching with respect to* (G, T_1, T_2) *is the minimum, over stable pairs* (D_1, D_2) *of*

$$|V \backslash (D_1 \cup D_2)| + |R| - \mathrm{odd}(G[D_1 \cap D_2]).$$

This theorem generalizes the Tutte-Berge Formula and also implies the existence theorem for perfect path-matchings. (It has been given a nice restatement and inductive proof by Frank and Szegő [19].) Morever, the convex hull of path-matching vectors (defined again to have a 2 for each matching edge) has a totally dual integral polyhedral description. The proof in Geelen [21] of the description follows by a construction from Theorem 6.2; that construction was introduced for matching by Schrijver [40]. The total dual integrality of the description is proved in [21] using another idea introduced for matching by Schrijver [39]. The min-max theorem above is proved in [21] from the total dual integrality of the system. The proofs in the published version [7] follow the same lines, except that they apply to a more general model (outlined below). Finally, we remark that algorithmic results for path-matching follow from those for perfect path-matchings by straightforward reductions.

Combinatorial algorithms for path-matching

The path-matching algorithms derived above depend on the ellipsoid method, and so are somewhat unsatisfactory. Similarly, the proofs of the existence theorems are non-constructive. One natural route to filling these gaps, would be an augmenting-path algorithm, generalizing Edmonds' matching algorithm. No such algorithm is yet known; we will say a bit more about this later.

A combinatorial method to find a path-matching of maximum value has been found [8]; it is an extension of Geelen's algorithm for matching. This approach is based on the following result, which generalizes Theorem 2.2; it can be proved by the same methods. In fact, this observation was the original motivation for the notion of path-matching.

Theorem 6.4 G, T_1, T_2 *admits a perfect path-matching if and only if the submatrix of* $A(G)$ *having rows indexed by* $V \backslash T_2$ *and columns indexed by* $V \backslash T_1$ *is nonsingular.*

In one direction, this implies that there exists a polynomial-time (ellipsoid) algorithm to determine whether any square submatrix of the Tutte matrix is nonsingular. (More generally, by elementary linear algebra, there is an efficient algorithm to find the rank of any submatrix of the Tutte matrix.) In the other direction, it allows the application of linear algebra to path-matching. One such application is an argument of Lovász [31] that gave the first characterization of existence for perfect path-matchings, and leads to a proof of Theorem 6.1. A second application is the extension of Geelen's matching algorithm to path-matching. Theorem 2.7 generalizes to submatrices of the Tutte matrix, thus yielding such an extension. (One small difference is that the values of the variables can be restricted to the set of integers i such that $-(|V|+1) \leq i \leq |V|+1$.)

It is also desirable to have a combinatorial algorithm to find a minimum-weight perfect path-matching. A primal-dual approach to this, with the idea of using the matrix algorithm as a subroutine, runs into the difficulty that the subproblems generated by the primal-dual approach are more general than path-matching. Here is a description of the new problem. Suppose that G' is a directed graph. An *even factor* is a collection of vertex-disjoint dipaths and even dicircuits. From a perfect path-matching problem, one can make such a problem by replacing each edge in $\gamma(R)$ by a pair of oppositely directed arcs, and replacing each other edge by a single arc, such that vertices in T_1 are not heads of arcs and vertices in T_2 are not tails of arcs. It is easy to see that there exists an even factor in G' having $|V| - k$ arcs if and only if G, T_1, T_2 has a perfect path-matching. The problem of finding an even factor of maximum cardinality is shown in [8] to be $\mathcal{NP}$-hard in general, but solvable in polynomial time for digraphs that are *weakly symmetric*, meaning that every strong component is symmetric. Since G' above is weakly symmetric, this result implies the solvability of the problem for path-matchings. The algorithm for weakly symmetric digraphs depends on a generalization of the Tutte matrix—if $e = ij$ is a directed edge then we put z_e into position (i, j), and put $-z_e$ or 0 into position (j, i), according to whether ji is a directed edge or not. These results do provide combinatorial algorithms for weighted perfect path-matchings, and some generalizations, and also constructive proofs of the polyhedral theorems.

Of course, the problem of finding an even factor of minimum weight is $\mathcal{NP}$-hard. Are there other special cases that are solvable? Notice that requiring only that the digraph be weakly symmetric will not be enough. Namely, we could reduce the problem of finding a maximum cardinality even factor in a digraph to the problem of finding a minimum-weight even factor in a weakly connected digraph, by giving each arc weight -1 and adding appropriate zero-weight arcs. So one must restrict the choice of weights, too. It turns out that one can solve the problem for digraphs G and weights c such that G is weakly symmetric and any two oppositely directed arcs have equal weights. Of course, this class of problems does include the optimal path-matching problem as a special case.

Path-matchings and jump systems

As indicated in the section on jump systems, an augmenting-path result for a family of subgraphs implies that the two-step axiom holds (at least for certain cases) for the degree sequences of those subgraphs. We will show that the degree sequences of path-matchings do satisfy the two-step axiom, which suggests that there may be some hope for finding an augmenting path result for path-matchings. (But as yet, no such result has been found.) Suppose that we define the degree sequence of a path-matching K by defining component v to be 2 if v is incident to a matching edge of K, and to be the number of path edges of K to which v is incident, otherwise.

Theorem 6.5 *The set of degree sequences of path-matchings with respect to G, T_1, T_2 is a jump system.*

The proof uses the following (known) lemma.

Lemma 6.6 *Let A be a matrix with rows indexed by I and columns indexed by J, where I and J are disjoint, and let $\mathcal{J}$ denote the incidence vectors of sets of the form $P \cup Q$ where $P \subseteq I$, $Q \subseteq J$ and the (P, Q)-submatrix of A is nonsingular. Then $\mathcal{J}$ is a jump system.*

Proof The result can be proved by elementary linear algebra, but here is another proof that uses some ideas from the previous section. Append an $|I|$ by $|I|$ identity matrix to A to form a matrix A', whose columns are indexed by $I \cup J$. Then the (P, Q) submatrix of A is nonsingular if and only if $(I \backslash P) \cup Q$ indexes a (column) basis of A'. Thus $\mathcal{J}$ can be obtained from the jump system $\mathcal{J}'$ of incidence vectors of these bases by negating the components for elements of I and translating by the incidence vector of I. ▮

Proof of Theorem 6.5 Let K be a path-matching. We choose a submatrix of $A(G)$ with rows I' and columns J' corresponding to K as follows. If a vertex v is incident with a matching edge of K or is an internal vertex of a path of K, put v into both I' and J'. For a path of K, put one end into I' and the other end into J', being careful always to put elements of T_1 only into I' and elements of T_2 only into J'. Now the (I', J')-submatrix of the Tutte matrix is nonsingular, since K is a perfect path-matching with respect to H, T_1', T_2', where $T_1' = I' \backslash J'$, $T_2' = J' \backslash I'$, and H is the subgraph obtained from G by deleting the vertices not in $I' \cup J'$ and the edges in $\gamma(T_1') \cup \gamma(T_2')$. Conversely, given a nonsingular square submatrix with row indices $I' \subseteq I = R \cup T_1$ and column indices $J' \subseteq J = R \cup T_2$, we form the subgraph H and terminal sets T_1', T_2' as above. Then (H, T_1', T_2') admits a perfect path-matching K, which is a path-matching with respect to G, T_1, T_2.

If we now think of I and J as disjoint sets (say, by making copies R_1, R_2 of R to index the rows and columns, respectively), then we get a jump system $\mathcal{J}$ from Lemma 6.6. We can now form a new jump system $\mathcal{J}'$ from $\mathcal{J}$ using homomorphism, getting the component for $v \in R$ by adding the components corresponding to its two copies. The feasible points will be precisely the degree sequences of path-matchings, as required. ▮

A matroid generalization

Now we describe a further extension [7] of the path-matching model, which also includes matroid intersection as a special case. We consider as before a graph $G = (V, E)$ and a partition of V into T_1, T_2, R with T_1, T_2 stable sets, but we drop the condition that T_1 and T_2 have the same size. On the other hand, now we are given matroids M_1 on T_1 and M_2 on T_2. We assume that the two matroids have the same rank k. A *basic path-matching* is now a set of vertex-disjoint paths joining a basis of M_1 to a basis of M_2 together with a perfect matching of the vertices of R not in any path.

In the special case in which T_i is a basis of M_i for $i = 1$ and 2, a basic path-matching is precisely a perfect path-matching. A second special case occurs when $R = \emptyset$ and G consists of a perfect matching joining T_1 to T_2. In this case, suppose we think of M_1 and M_2 as matroids on the same set T that have been copied onto the sets T_1, T_2, respectively, with edges joining corresponding copies. Then a basic path-matching is a subset of the edges matching a basis of M_1 to a basis of M_2, and hence corresponds to a common basis.

Most of the results for perfect path-matching above go through—existence theorem, polyhedral description, existence of polynomial-time solution algorithms (based, again, on the equivalence of separation and optimization). We limit ourselves here to stating the existence theorem. (Further results and proofs are in [7].) The reader can use it to derive the existence theorems for perfect path-matching and for common bases of two matroids.

Theorem 6.7 *There exists a basic path-matching with respect to G, M_1, M_2 if and only if there does not exist a stable pair (D_1, D_2) for which*

$$r_1(T_1 \backslash D_1) + r_2(T_2 \backslash D_2) + |R \backslash (D_1 \cup D_2)| < k + \text{odd}(G[D_1 \cap D_2]).$$

Basic path-matching seems to provide the desired solvable common generalization of (weighted) matching and (weighted) matroid intersection. We explain now how this model can be handled combinatorially. (These ideas are from [8].) Surprisingly, the techniques required to do this are essentially path-matching plus (valuated) matroid intersection. This may suggest that the generalization from path-matching to basic path-matching is not so substantial.

Let V' be a copy of V. For each subset P of V, we use P' to denote the corresponding subset of V'. We also use M_1' to denote the copy of matroid M_1 on T_1'. We define two matroids N_a and N_b on $V \cup V'$, each having rank $|V|$. Let N_a be the matroid of the matrix $(I|A(G))$, where the columns of the identity matrix I are indexed by the elements of V'. Then the bases of N_a are the sets $(V' \backslash P') \cup Q$, where P and Q are subsets of V such that the (P, Q)-submatrix of $A(G)$ is (square and) nonsingular. From Theorem 6.4, for $(V' \backslash P') \cup Q$ to correspond to a basic path-matching, it will be enough to assure in addition that

(a) $P \cap T_1$ is a basis of M_1 and $Q \cap T_2$ is a basis of M_2;

(b) $P \cap T_2 = \emptyset = Q \cap T_1$;

(c) $P \supseteq R$ and $Q \supseteq R$.

We define N_b to have as bases, sets of the form

$$R \cup T_2' \cup (T_1' \backslash B_1') \cup B_2,$$

where B_1 is a basis of M_1 and B_2 is a basis of M_2. It is easy to see that N_b is a matroid, since it is the "direct sum" of M_2, the dual of M_1', and a trivial matroid. Notice that P, Q will have properties (a),(b),(c) above if and only if $(V' \backslash P') \cup Q$ is a basis of N_b. So we get the following result.

Proposition 6.8 *There is a basic path-matching with respect to G, M_1, M_2 if and only if there is a common basis with respect to N_a, N_b.*

One can use this construction to prove the Basic Path-Matching Existence Theorem 6.7 from the Matroid Intersection Theorem and the Path-Matching Existence Theorem 6.1. From the algorithmic point of view, we can apply the Matroid Intersection Algorithm to determine whether there exists a basic path-matching. It needs efficient subroutines to determine whether a given subset S of $V \cup V'$ is independent in N_a, N_b. For N_b, one needs to check that $S \cap T_2$ is independent in M_2, that $T_1' \backslash S$ contains a basis of M_1', that $S \cap R' = \emptyset$, and that $S \cap T_1 = \emptyset$. Therefore, if we have efficient algorithms to test independence in M_1, M_2, then we have such an algorithm for N_b. To test whether S is independent in N_a, we need to know, where $P' = V' \backslash S$ and $Q = S \cap V$, whether the (P, Q)-submatrix of $A(G)$ has rank $|Q|$. As we pointed out earlier, we do have an efficient combinatorial algorithm for determining the rank of any submatrix of the Tutte matrix.

Optimal basic path-matching

Finally, we would like to have a combinatorial algorithm for finding a minimum-weight basic path-matching. Presumably, it is possible to adopt a primal-dual approach, as was done for perfect path-matchings; one would first generalize basic path-matchings to "basic even factors" in weakly symmetric digraphs. In [8], a different approach is taken; it utilizes work of Murota on valuated matroids.

A *valuation* on a matroid M on T is a function w assigning an integer weight to each basis B of M so that for bases B, B' and elements $u \in B \backslash B'$ and $v \in B' \backslash B$

$$w(B) + w(B') \leq w((B \backslash \{u\}) \cup \{v\}) + w((B' \backslash \{v\}) \cup \{u\})$$

whenever the sets on the right are bases. This notion was introduced by Dress and Wentzel [11]. The function w given by $w(B) = c(B)$, arising from an "ordinary" (integral) weighting c of the elements of T, is a valuation. Murota [34] has described a polynomial-time combinatorial algorithm that, given two matroids on T and valuations w^1, w^2 for them, finds a common basis B maximizing $w^1(B) + w^2(B)$. His algorithm requires, for each of the matroids, efficient subroutines to test the independence of a given subset, and to give the valuation associated with a given basis.

Suppose that there exists a basic path-matching, and we wish to find a maximum-weight basic path-matching with respect to given non-negative integral weights c_e. (Any minimization problem with integral data can be converted to this form, by first negating the weights and then adding a sufficiently large positive integer to each of them.) We replace the Tutte matrix $A(G)$ by the matrix A' obtained by multiplying the entries containing z_e by t^{c_e}, where t is a new variable. For a polynomial p in t and the z_e, we denote by $\deg_t(p)$, the degree of the polynomial obtained from p by treating the z_e as constants. For each square nonsingular submatrix $A'(P, Q)$ of A' and the corresponding basis $(V' \backslash P') \cup Q$ of N_a, define

$$w((V' \backslash P') \cup Q) = \deg_t(\det(A'(P, Q)).$$

It follows from a result of Dress and Wentzl [11] that w is a valuation on N_a. Note that, given P, Q, the required degree is equal to the maximum weight of a perfect path-matching with respect to $G[P \cup Q], P \backslash Q, Q \backslash P$. If we now assign N_b the zero valuation, it is easy to see that solving the valuated matroid intersection problem on these two valuated matroids, solves the maximum weight basic path-matching problem. Thus we can apply Murota's algorithm to solve the optimal basic path-matching problem. It requires a subroutine for *optimal* perfect path-matching.

7 Remarks

The importance of the two extensions that we have emphasized here remains to be seen. Path-matching seems to be well-understood in the sense that we have the basic results that we would like. What is not clear, is whether there are additional useful applications of the model. In other words, this may be an extension that is a bit too special to be really important. Jump systems have somewhat the opposite characteristic. They have very many applications, but, in the sense that the membership problem is unsolvable, they may be too general. This suggests the challenge of finding additional conditions under which the membership problem for jump systems is solvable.

Acknowledgments. I am grateful to André Bouchet and Jim Geelen for working with me on these subjects, to Lászlo Lovász, Kazuo Murota, and András Sebő for conversations, and to the referees for suggestions that improved the paper.

References

1. A. Bouchet, "Greedy algorithm and symmetric matroids", Math. Programming 38(1987), 147–159.

2. R.E. Bixby and W.H. Cunningham, "Matroid optimization and algorithms", in: R.L. Graham, M. Grötschel, and L. Lovász (eds.) *Handbook of Combinatorics,* North-Holland, 1995, pp. 550–609.

3. A. Bouchet and W. H. Cunningham, "Delta–matroids. jump systems, and bisubmodular polyhedra", SIAM J. Disc. Math. 8(1995), 17–32.

4. R. Chandrasekaran and S.N. Kabadi, "Pseudomatroids", Discrete Math. 71(1988), 205-217.

5. W.J. Cook, *et al.*, *Combinatorial Optimization*, Wiley, New York, 1998.

6. G. Cornuéjols and W.R. Pulleyblank, "A matching problem with side conditions", Discrete Mathematics 29(1980), 135–159.

7. W. H. Cunningham and J. F. Geelen, "The optimal path-matching problem", Combinatorica 17(1997), 315–337.

8. W. H. Cunningham and J. F. Geelen, "Combinatorial algorithms for path-matching", manuscript, 2000.

9. W. H. Cunningham and A. B. Marsh III, "A primal algorithm for optimum matching", Math. Programming Study 8(1978), 50–72.

10. A. Dress and T. Havel, "Some combinatorial properties of discriminants in metric vector spaces", Adv. Math. 62(1986), 285-312.

11. A.W.M. Dress and W. Wentzel, "Valuated matroids", Advances in Math. 93(1992), 214–250.

12. F.D.J. Dunstan and D.J.A. Welsh, "A greedy algorithm for solving a certain class of linear programmes," Math. Programming 5(1973), 338–353.

13. J. Edmonds, "Paths, trees and flowers", Canad. J. Math. 17(1965), 449–467.

14. J. Edmonds, "Maximum matching and a polyhedron with $0, 1$ vertices", J. Res. Nat. Bur. Standards Sect. B 69(1965), 125–130.

15. J. Edmonds, "Submodular functions, matroids and certain polyhedra", in: R.K. Guy, *et al.* (eds.) *Combinatorial Structures and their Applications*, Gordon and Breach, New York, pp. 69–87.

16. J. Edmonds and F.R. Giles, "A min-max theorem for submodular functions on graphs", Annals of Discrete Math. 1(1977), 185–204.

17. A. Frank, "Restricted t-matchings in bipartite graphs", EGRES Technical Report TR-2001-10, Budapest, 2001.

18. A. Frank and T. Jordan, "Minimal edge-coverings of pairs of sets". J. Combinatorial Theory B 65 (1995), 73–110.

19. A. Frank and L.Szegő, "A note on the path-matching formula", to appear.

20. S. Fujishige, "A min-max theorem for bisubmodular polyhedra", SIAM J. Discrete Math. 10 (1997), 294–308.

21. J.F. Geelen, *Matchings, Matroids, Unimodular Matrices,* doctoral thesis, University of Waterloo, 1995.

22. J.F. Geelen, "Notes on jump systems", preprint, 1996.

23. J.F. Geelen, "An algebraic matching algorithm", Combinatorica 20(2000), 61–70.

24. M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization", Combinatorica 1 (1981), 169–197.

25. D.B. Hartvigsen, *Extensions of Matching Theory,* Ph.D. Thesis, Carnegie Mellon University, Pittsburg, PA, 1984.

26. D.B. Hartvigsen, "The square-free 2-factor problem in bipartite graphs", in: R. Burkard, *et al.,* eds. *Proceedings of Conference on Integer Programming and Combinatorial Optimization,* Lecture Notes in Comput. Sci., 1610, Springer, Berlin, 1999, pp. 234–241,

27. P. Hell, D. Kirkpatrick, J. Kratchovil, and I. Kriz, "On restricted two-factors", SIAM J. Discrete Math. 1(1988), 472–484.

28. Z. Kiraly, "C_4-free 2-factors in bipartite graphs", EGRES Technical Report TR-2001-13, Budapest, 1999.

29. M. Koren, "Extreme degree sequences of simple graphs", J. Combinatorial Theory B 15(1973), 213–224.

30. L.Lovász, "Matroid matching and some applications", J. Combinatorial Theory B 28(1980), 208–236.

31. L.Lovász, private communication, 1992.

32. L. Lovász, “The membership problem in jump systems”, J. Combinatorial Theory B 70(1997), 45–66.

33. L. Lovász and M.D. Plummer, *Matching Theory*, North-Holland, Amsterdam, 1986.

34. K. Murota, “Valuated matroid intersection. II. Algorithms”. SIAM J. Discrete Math. 9(1996), 562–576.

35. M. W. Padberg and M. R. Rao, “Odd minimum cut-sets and b-matchings”, Math. Oper. Res. 7(1982), 67–80.

36. R. Rado, “A theorem on independence relations,” Quarterly Journal of Mathematics Oxford 13(1942), 83–89.

37. R. Rado, “A note on independence functions,” Proceedings of the London Mathematical Society 7 (1957), 300–320.

38. M. Russell, *Restricted Two-factors*, Master’s Thesis, University of Waterloo, Waterloo, Canada, 2001.

39. A. Schrijver, “Short proofs on the matching polyhedron”, J. Combinatorial Theory Ser. B 34(1983), 104–108.

40. A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York, 1986.

41. W. T. Tutte, “The factorization of linear graphs”, J. London Math. Soc. 22(1947), 107–111.

42. O. Vornberger, “Easy and hard cycle covers”, preprint, Universität Paderborn, 1980.