

A constructive approach to matching and its generalizations

Gyula Pap

Ph.D. thesis submitted to
Eötvös Loránd University, Institute of Mathematics



Doctoral School: Mathematics
Director: Miklós Laczkovich, Professor,
Member of the Hungarian Academy of Sciences

Doctoral Program: Applied Mathematics
Director: András Prékopa, Professor Emeritus,
Member of the Hungarian Academy of Sciences

Supervisor: András Frank, Professor,
Doctor of Sciences

Supported by: Egerváry Research Group
of the Hungarian Academy of Sciences

September, 2006

Contents

i. Motivation	4
ii. Outline	6
iii. Notation	10
1 Graph matching	11
1.1 Warm-up – bipartite matching	11
1.2 Restricted b -matching in bipartite graphs	15
1.3 Non-bipartite matching	23
1.4 Even factor in odd-cycle-symmetric directed graphs	29
1.5 Hypo-matching in directed graphs	37
2 Matroid matching and related problems	39
2.1 Matroid intersection	42
2.2 Linear matroid matching	49
2.3 The Partition Formula for polymatroid matching	59
2.4 Matroid fractional matching	69
3 Packing \mathcal{A}-paths	77
3.1 Node-capacitated packings of \mathcal{A} -paths	83
3.2 Node-capacitated fractional packing of \mathcal{A} -paths	86
3.3 Fractional packing of \mathcal{A} -paths reduces to matroid fractional matching	95
3.4 A semi-strongly polynomial time algorithm for node-capacitated pack- ing of \mathcal{A} -paths	97
3.5 Mader matroids are gammoids	101
3.6 Packing non-returning \mathcal{A} -paths	106
3.7 Packing non-returning \mathcal{A} -paths algorithmically	114
4 Open Questions	132

Acknowledgements

Let me grab this opportunity to express my thanks and gratitude towards my supervisor András Frank for his great support and guidance through all those years. It is a pleasure and privilege to see his inspiring motivation and experience in pure mathematics. I am also thankful for András initiating the Egerváry Research Group, thus providing the everyday atmosphere of sharing ideas. Thanks are due to members of the group, in particular to Laci Szegő for the long-lasting cooperation on path-matching, even factor and discussions on \mathcal{A} -paths, to Marci Makai, Tamás Király, Jácint Szabó for discussions on matroid matching. I thank guests of the group, in particular Dion Gijswijt for his short proof on fractional matroid matching, Lex Schrijver for his view on \mathcal{A} -paths, and Jack Edmonds for an inspiring conversation on non-bipartite matching.

It is difficult to put in words how thankful I am to my wife Olgi, the most wonderful person ever, encouraging and taking care of me every day. I would like to thank my parents, grandparents, sister for their patient support.

i. Motivation

Combinatorial optimization focuses on the computational aspect of discrete structures such as directed and undirected graphs, hypergraphs, matroids, polymatroids, which often are subject to weight, cost, or capacity functions returning integer, rational, or real values. Usually an optimization problem is that of finding a substructure fulfilling specific combinatorial properties, and maximizing or minimizing its cardinality, or its weight. In particular, we are interested in those problems admitting a good characterization, or a polynomial time algorithm. In case of a weight-function, we may as well be interested in strongly polynomial time algorithms. While the broad literature on this topic also considers approximation and randomized algorithms, heuristics, and applications, in this thesis we only discuss deterministic approaches. The study of linear cost-functions has led to a growing interest in linear programming formulations of corresponding polyhedra. From a discrete point of view, we are especially interested in the primal and dual integrality of polyhedra, which often is related to a combinatorial min-max formula. The expression “good characterization” was introduced by Jack Edmonds for those combinatorial properties in $\text{NP} \cap \text{co-NP}$, and this extends to optimization problems by requiring a certificate of optimality, which provides a min-max formula. Classical examples of a good characterization are Kőnig’s theorem on bipartite matching, max-flow-min-cut in network flows, the Berge-Tutte formula on non-bipartite matching, and Edmonds’ matroid intersection formula. In this thesis we will only discuss “tractable” problems, i.e. those admitting a good characterization. We will of course add a few remarks on how close these problems are to intractable ones.

In the context of this thesis, an algorithm is required to return not only an optimum solution, but also a certificate of optimality via the min-max formula. Decades of research have shown that, while only a small portion of combinatorial optimization problems are tractable, the impression is that most tractable problems are solved by a polynomial time algorithm. Thus, the great achievement in combinatorial optimization is showing a good characterization of a problem, distincting it from NP-hard problems. Often, a subsequent paper discusses the polynomial time solvability of the same problem, which in many cases is of a barely technical nature after already having the good characterization at hand. This is why in the eye of a spectator the discovery of a good characterization is respected as the major breakthrough with respect to a specific problem. Nevertheless, for some of those problems knowingly admitting a good characterization, the polynomial time solvability turns out rather cumbersome, indirect, or even remains unsolved for years to come. For these problems we should re-think and re-interpret some classical ideas, and also add some of our own, in order to better understand their complexity and behavior.

In this thesis we will focus on some of these problems by introducing an algorithmic framework, also in the broader context of analogous problems, and in relation with classical concepts.

For many problems there is a broad literature of improving on running time of algorithms, which leads to deep results on combinatorial data-structures. For example, network flow, bipartite and non-bipartite matching, matroid and polymatroid algorithms have been intensively studied subject to time complexity. Another prominent topic is that of pushing the boundaries of tractability by proving good characterizations and constructing efficient algorithms for more and more general problems. Some of these efforts, while claiming a deep, pioneering result, often yield sophisticated methods which are difficult to interpret, and uneasy to access for a wide audience. Let us mention for example path-matching, even factor, \mathcal{A} -paths, or square-free 2-factor, for which an attractive min-max formula has been proven, but some promising algorithmic approaches yielded unexpected complications. Thus it is not tempting to implement the given algorithm directly, or to improve on their running time, before we understand what is going on there.

My intention in writing this thesis is to access some of these tractable problems by other means, refining classical ideas, introducing algorithmic frameworks, and putting some of these results in a broader picture. We will also revisit some elementary special cases to help understanding the crucial elements of the concept, and then we go on to more general problems to see how far our framework will reach.

The algorithms constructed in the thesis are put together from well-defined simple sub-routines, and thus we avoid the detailed description of sophisticated augmentation structures, such as Edmonds' alternating forest of blossoms. The point is that, while the complexity and entanglement of an augmentation structure is rather difficult to apply for more general problems, our simple sub-routines directly generalize and are easy to understand. Let me reiterate that the simplicity and compactness of the framework is not likely to result in a world record running time, since there is little communication between the sub-routines. However, it is interesting and useful to see, what are the crucial turning points in, say Edmonds' algorithm, what information may or may not be dropped in a general step of the algorithm. Hence, we will be able to construct very simple combinatorial algorithms for path-matching, even factor, square-free 2-factor, \mathcal{A} -paths, matroid intersection. We remark that Orlin and Vande Vate constructed a linear matroid matching algorithm that is related to our framework, and we will put their algorithm into the big picture of the thesis. For some of these problems, an augmentation structure is unknown; for some others, the construction of an augmentation structure is quite complicated, and its maintenance needs special care. Actually, this is the motivation behind all of this thesis, to find other ways of dealing with the above problems.

ii. Outline

As indicated in the introduction, in this thesis we propose an algorithmic approach to a bunch of interrelated combinatorial optimization problems. The author would like to add the remark that, due to the variety of topics to be visited, a comprehensive overview will not be given for all of them. Please consider this thesis as a collection of short notes selected as a cross-section of these combinatorial optimization problems.

In the first chapter we focus on graph matching algorithms. We propose an algorithmic proof for bipartite matching, for restricted b -matching in bipartite graph (square-free 2-factor a special case), for even factor in directed graph (path-matching a special case), and for hypo-matching in directed graph (known before for undirected graphs). To outline the framework of these algorithms, first let's go back to the outstanding result in this area, Edmonds' matching algorithm, which then served as a blueprint for many generalizations. We will propose a variation of Edmonds' matching algorithm, and this variation will be easier to generalize to the of problems listed above. Here we briefly outline our matching algorithm in comparison with Edmonds'. The crucial difference between Edmonds' and our algorithm is that we avoid building up the data-structure of the alternating forest of blossoms. Instead, we use a simple subroutine to find a matching with an alternating odd cycle, and we show that the contraction of this odd cycle is an equivalent reduction. The algorithm is composed of subroutines for two simple lemmas, the 3-Way-Lemma concerning optimality and alternating odd cycles, and another lemma on the contraction of an alternating odd cycle. We remark that the algorithm we obtain this way is closely related to Edmonds'. One should think of our approach as a variation of Edmonds' algorithm where, generally speaking, we drop a lot of information acquired in a subroutine, before performing the next subroutine. That is, while Edmonds' algorithm maintains the alternating forest as our memory of alternating paths already traced, we show that these traces are not necessary for our algorithm. All we need to maintain is a single matching in the contracted graph. Hence, our approach appears a distant relative of Edmonds', since we will not at all be speaking of blossoms or alternating forests, which is hidden implicitly behind all those subroutines. The point is that these subroutines are more flexible, than the rigid structure of alternating forests. We will find this flexibility useful in all those generalizations.

Let us briefly outline the background of those other problems in the first chapter, too. We begin with a short introductory section on bipartite matching, just to show that it is quite easy to formulate our algorithmic framework there, as well. Here we propose a simple reduction principle, playing an analogous role as the contraction of odd cycles in our non-bipartite matching algorithm. We remark that this bipartite matching algorithm is the specialization of the matroid intersection

algorithm presented later, in another section.

Then we will describe an algorithm for restricted b -matching in bipartite graphs. Earlier, Frank provided a min-max formula for a special case. The problem is finding a maximum cardinality simple b -matching avoiding some pre-specified forbidden complete bipartite subgraphs. A famous special case is square-free 2-factors, for which Hartvigsen proposed a combinatorial algorithm using an augmentation structure. Another polynomial time algorithm follows from results of Benczúr and Végh. Though restricted b -matching is not known to be a special case of bipartite matching or matroid intersection, our general algorithmic framework easily applies. Here we will “contract” complete bipartite subgraphs, which plays an analogous role as the contraction of odd cycles in our non-bipartite matching algorithm.

After the detailed description of the non-bipartite matching algorithm sketched above, we will generalize this concept for even factors, and then for hypo-matchings in directed graphs. The maximum even factor problem is a generalization of the maximum path-matching problem, and both of these were proposed and first solved by Cunningham and Geelen. (For the definition, see the respective section.) They proved various polyhedral and algebraic results, and constructed two polynomial time algorithms for even factor: one via ellipsoid method, and another via algebraic properties of the so-called Tutte-matrix. The question remained, whether one could construct an Edmonds-type combinatorial algorithm for path-matching, or even factor. In the section on even factors, we generalize our version of Edmonds’ matching algorithm to even factors, and thus obtain a simple constructive proof of the even factor formula. The point is that Edmonds’ framework of alternating forests does not generalize to even factor or path-matching, since branches of an alternating forest mess up with each other, and it seems difficult to define properly what should be regarded as an alternating forest for even factor or path-matching. Luckily, the flexible sub-routines in our matching algorithm are easy to generalize for even factor and path-matching, and thus we obtain a simple combinatorial algorithm.

We propose a common generalization of hypo-matching and even factor, and prove that this problem – hypo-matching in directed graphs – admits a min-max formula and a polynomial time algorithm. This generalization relies on the framework already developed for even factor, and an idea of Cornuéjols, Hartvigsen, Pulleyblank concerning hypo-matching in undirected graphs.

In the second chapter we propose a discussion of matroid matching and related topics. Matroid matching was proposed by Lawler in the 1960’s as common generalization of matroid intersection and non-bipartite matching. The problem is that, given a matroid over a groundset partitioned into pairs, find a maximum set of pairs the union of which is an independent set. Assuming a matroid by an independence oracle, the time complexity of matroid matching is exponential. Hence our attention

is towards the growing number of tractable special cases. The groundbreaking positive result is Lovász' min-max formula on matching in linear matroids, then followed up by a sequence of generalizations, and polynomial time algorithms. Next, a brief outline of those four problems related to matroid matching which are considered in this chapter.

First, we propose a constructive proof of Edmonds' matroid intersection formula. This is based on the algorithmic framework used in the first chapter for various graph matching problems. To generalize our bipartite matching algorithm for matroid intersection, we apply two constructive lemmas in a similar recursive fashion. Here we make use of a lesser-known matroid operation, the projection of a subset of the groundset, which decreases the rank of that flat to one.

Second, we propose an equivalent discussion of Orlin and Vande Vate's linear matroid matching algorithm, which is – amongst all known linear matroid matching algorithms – closest to the point of view in this thesis. This algorithm is based on a representation of a sub-problem by an instance of matroid intersection. While their original description relies on a specific implementation of a matroid intersection algorithm, we will cite a structural description of Edmonds' matroid intersection formula.

Third, we propose a constructive proof for a class of polymatroid matching problems, which implies a polynomial time algorithm for a couple of graph theoretical special cases such as maximum genus graph embedding, and parity-constrained rooted- k -arc-connected graph orientations. The key to this is a recent result of Makai and Szabó, claiming the Partition Formula for a class of polymatroids, for which they proposed a non-constructive proof. We propose an alternative, constructive proof, which implies a semi-strongly polynomial time algorithm based on a rank-function oracle.

Fourth, we consider a fractional relaxation of matroid matching. There are various plausible definitions for a fractional relaxation, but only one which contains non-trivial applications such as matroid intersection and graph fractional matching. In a subsequent section we will also present a new application, the fractional packing of \mathcal{A} -paths. Vande Vate proved a min-max formula for matroid fractional matching, and also showed that there is a half-integer primal optimum. We will discuss an alternative proof proposed by Gijswijt, who showed the slightly stronger result that the description of the polytope of matroid fractional matchings is totally dual half-integral.

In the third chapter, we consider path-packing problems, generalizing non-bipartite matching, and disjoint paths. Mader's original problem is the following. Suppose we are given a graph $G = (V, E)$, and a family \mathcal{A} of disjoint subsets of V . A path is called an \mathcal{A} -path if it connects two distinct members of \mathcal{A} . The outstanding result

is Mader's Formula, which determines the maximum number of pairwise fully node-disjoint \mathcal{A} -paths. First we survey known results related to Mader's Formula. Then we propose a new discussion of node-capacities, fractional packing, a polyhedral description, matroid fractional matching, the Mader matroid, and non-returning A -paths. A brief overview of these results:

We consider the node-capacitated version, where the number of \mathcal{A} -paths traversing a node is bounded from above by the node's capacity. This node-capacitated problem reduces to Mader's Formula by splitting nodes into a number of copies, which only implies a weakly-polynomial time algorithm. We also develop a semi-strongly polynomial time algorithm by exploiting a polyhedral description of the fractional relaxation, and converting a maximum fractional packing into a maximum integral packing. As an interesting by-product, we will be able to show that fractional packing is equivalent with an instance of matroid fractional matching. We also show the primal and dual half-integrality of a polyhedral description. We prove a positive answer to Schrijver's question "Is each Mader matroid a gammoid?".

We prove that Mader's min-max formula for packing \mathcal{A} -paths may be generalized to the problem of packing non-returning A -paths. This also extends results of Chudnovsky et al. on packing non-zero A -paths. We discuss other possible models for path-packing, and show that these models are equivalent with packing non-returning A -paths. We propose a combinatorial algorithm for packing non-returning A -paths, which implies a new polynomial time algorithm for \mathcal{A} -paths, as well.

iii. Notation

$G = (V, E)$	An undirected graph G with node-set V and edge-set E .
$E[U]$	The set of edges induced in a set U of nodes.
$G[U] := (U, E[U])$	The subgraph induced by a set U of nodes.
$G - U := G[V - U]$	The deletion of a set U of nodes.
$\delta_G(U)$	The set of edges leaving a set U of nodes.
$\Gamma_G(U)$	The set of neighbours of a set U of nodes.
$D = (V, A)$	A directed graph (digraph) D with node-set V and arc-set A .
$A[U]$	The set of arcs induced in U .
$D[U] := (U, A[U])$	The subgraph induced by a set U of nodes.
$D - U := D[V - U]$	The deletion of a set U of nodes.
$\delta_D^{in}(U)$	The set of arc entering a set U of nodes.
$\delta_D^{out}(U)$	The set of arcs leaving a set U of nodes.
$\Gamma_D^+(U)$	The set of out-neighbours of a set U of nodes.
path	Edge/arc-set of an unclosed walk without repetition of nodes.
cycle	Edge/arc-set of a closed walk without repetition of nodes.
matching	A set of disjoint edges.
$\nu(G)$	The maximum cardinality of a matching.

1 Graph matching

1.1 Warm-up – bipartite matching

In this section we construct a conceptually simple combinatorial algorithm for bipartite matching, composed from simple sub-routines. In subsequent sections, we will exploit an analogous algorithmic framework for more general problems, and the motivation is to use bipartite matching didactically as the simplest application.

Our bipartite matching algorithm is based on a recursive subroutine which, given an initial matching, constructs either a larger matching, or a certificate of the maximality of the initial matching. The textbook solution of this sub-routine is based on building up a search trees of possible starting segments of alternating paths. Instead we apply a simple reduction principle called V-reduction, which is more flexible and easier to generalize to those other problems considered in subsequent sections.

We prove the following version of König's Theorem.

Theorem 1.1 (König, [47]) *If $G = (A, B; E)$ is a bipartite graph, then*

$$(1) \quad \nu(G) = \min_{X \subseteq A} (|\Gamma_G(X)| + |A - X|).$$

It is straightforward that $|M| \leq |\Gamma_G(X)| + |A - X|$ holds for any matching $M \subseteq E$ and any set $X \subseteq A$. This implies $\nu(G) \leq \min_{X \subseteq A} |\Gamma_G(X)| + |A - X|$, which is the easy part of the theorem. A set $X \subseteq A$ is called a **verifying set** for a matching M , if equality $|M| = |\Gamma_G(X)| + |A - X|$ holds. König's Theorem is equivalent with the statement that there is a matching with a verifying set. Algorithmically, the goal is to find a pair of a matching and a verifying set. We reach this goal by a recursive application of subroutines for the following two easy lemmas.

Lemma 1.2 (3-Way Lemma for bipartite matching) *Let $G = (A, B; E)$ be an arbitrary bipartite graph, and consider an arbitrary matching $M \subseteq E$. Let $X := \{a \in A : a \text{ is an isolated node in } G\}$. At least one of the following alternatives holds:*

- (2) X is a verifying set for M .
- (3) There is an edge $ab \in E$ such that $a \in A - V(M)$ and $b \in B - V(M)$.
- (4) There is are edges $ab \in E, cb \in M$ such that $a \in A - V(M)$.

Moreover, one can find in polynomial time X , or ab , or ab, cb for one of the respective alternatives.

Proof. If $X = A - V(M)$, then alternative (2) holds. Otherwise, choose a node $a \in A - X - V(M)$. Then there is an edge $ab \in E$. If $b \in B - V(M)$, then (3) holds. If $b \in V(M)$, then (4) holds. \square

Next, consider the case when (4) holds, i.e. there are edges $ab \in E$, $cb \in M$ such that $a \in A - V(M)$. We define $G' = (A', B'; E') := (G - b)/\{a, c\}$, where $q := \{a, c\}$ denotes the new node. Notice that $M' := M - cb$ is a matching in G' . The reduction $G, M \rightarrow G', M'$ is called a **V-reduction**. We show that the V-reduction is an “equivalent reduction”, that is, a possible augmentation or a verifying set in G', M' can be lifted to the original pair G, M . This is what the following Lemma states.

Lemma 1.3 *Using notation from above, both of the following assertions hold.*

- (5) *If we are given a matching in G' larger than M' , then one can find in polynomial time a matching in G larger than M .*
- (6) *If X' is a verifying set for M' in G' , then $\{q\} \in X'$, and $X := X' - \{q\} + \{a, c\}$ is a verifying set for M in G .*

Proof. For claim (5), consider an arbitrary matching N in G' . If $q \notin V(N)$, then $N + ab$ (and $N + cb$) is a matching in G . If $qr \in N$, then $N - qr + ar + cb$ or $N - qr + cr + ab$ is a matching in G . For claim (6), notice that the new node q is M' -exposed, thus $q \in X'$. It is easy to see that $A - X = A' - X'$ and $\Gamma_G(X) = \Gamma_{G'}(X') + b$. Thus $|M| - 1 = |M'| = |\Gamma_{G'}(X')| + |A' - X'| = |\Gamma_G(X)| + |A - X| - 1$. \square

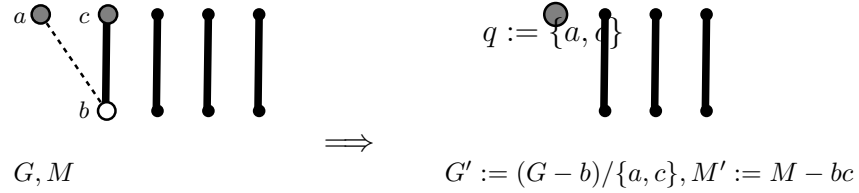


Figure 1: A V-reduction.

Proof of König’s Theorem. The proof is by induction on the number of nodes. Consider an arbitrary maximum matching M , and apply the 3-Way Lemma 1.2. If (2) holds, then we are done. If (3) holds, then $M + ab$ is a matching, which is a contradiction. If (4) holds, then construct G', M' as above. By the first part of Lemma 1.3, M' is a maximum matching in G' . By induction there is a verifying set X' for M' with respect to G' . By the second part of Lemma 1.3, there is a verifying set X for M with respect to G . This completes the proof of König’s Theorem. \square

Algorithmic proof of König’s Theorem. Next, we discuss how a bipartite matching algorithm follows from the two lemmas. For this, it suffices to construct a procedure which, given a matching M in a bipartite graph G , returns either a

larger matching, or a verifying set. A procedure solving this is called a VERIFY-OR-AUGMENT procedure, and will be constructed by a recursive application of subroutines for the 3-Way Lemma 1.2 and Lemma 1.3 as follows. Consider a pair of a graph G and a matching M . Applying the 3-Way Lemma, we obtain either a verifying set, or a larger matching, or a V-configuration. We are done, except when we find a V-configuration, and in that case we construct the pair G', M' , and apply VERIFY-OR-AUGMENT recursively to the pair G', M' . Given a larger matching or a verifying set for G', M' , by Lemma 1.3, this can be lifted to G, M . This recursion is feasible, since a V-reduction decreases the number of nodes in A . \square

Remarks

- The running time of this bipartite matching algorithm is polynomial in the input size. Here is a running-time calculation of a trivial implementation. Clearly, we apply the VERIFY-OR-AUGMENT procedure at most $|A|$ times, and at most $|A|^2$ V-reductions in total. Finding and constructing the V-reduction takes $O(|V| + |E|)$ time. Lifting a larger matching or a verifying set from G' to G takes $O(|V| + |E|)$ time. Thus the total running time is $O(|A|^2(|V| + |E|))$. (Using an advanced data structure to represent merged nodes after a V-reduction, the running time improves to $O(|V|^3)$.)
- The running time is not competitive with the best known bounds. The reason is that we maintain only a small amount of information through our reduction principles. For example, after checking the nodes in $A - V(M)$ whether they are isolated in G , we drop the acquired information, and check many of these nodes once again after the V-reduction. However, the above presentation points out a small (possibly the smallest in some sense) amount of information which is enough to go on after the reduction.
- How does this algorithm relate to well-known alternating-path-algorithms? To understand this relation, suppose G^m, M^m is obtained from G, M by a sequence of m V-reductions. Let X be the set of isolated nodes of A^m in G^m . Then the pre-image of X induces an M -alternating forest in the original graph G . Thus our algorithm can simulate any alternating-path-algorithm which builds an alternating forest rooted in A . We could of course apply a mixture of V-reductions rooted in A and B . This would give a slightly more general version of our algorithm, which can simulate any alternating forest algorithm.

- An interpretation of the above matching algorithm is the following. Let us call a set of edges an ***A*-matching**, if it hits pairwise distinct nodes of *A*. Every matching is an *A*-matching as well, hence *A*-matching is a relaxation of matching. Finding a maximum *A*-matching is very easy: we pick exactly one edge from those incident with a non-isolated node of *A*. Now, consider an arbitrary matching *M*. As we understand *A*-matchings quite well, we can easily check whether *M* is a maximum *A*-matching. If the answer is positive, then it is easy to see that the set *X* defined in Lemma 1.2 is a verifying set. Otherwise, there is an edge $ab \in E$ such that $M + ab$ is an *A*-matching, that is $a \in A - V(M)$. Thus the 3-Way Lemma claims that *M* is a maximum *A*-matching, or there is a larger matching, or there is a *V*-configuration.

1.2 Restricted b -matching in bipartite graphs

The main result in this section is a min-max formula characterizing the maximum cardinality of so-called \mathcal{K} -free b -matchings in bipartite graphs. This comes as a generalization of earlier results of D. Hartvigsen [39, 40], Z. Király [45], and A. Frank [24]. Our proof of the formula is constructive and thus implies a combinatorial algorithm to find a maximum \mathcal{K} -free b -matching. The first polynomial time algorithm follows from a recent work of Benczúr and Végh [2, 3], via the general theory of covering pairs of sets. Here we show that restricted b -matching is easier to solve directly.

Throughout this paper, the expression “ b -matching” stands for edge-sets in a given undirected graph obeying an upper-bound condition on the degrees. Such edge-sets are often referred to as “simple” b -matchings. Here we omit the word “simple”. A b -matching is called \mathcal{K} -free if it contains no subgraphs from a specific class \mathcal{K} of forbidden subgraphs. We will prove a min-max formula, assuming that the class \mathcal{K} of forbidden subgraphs fulfills a technical condition (8). Let us give an overview of earlier results which have motivated and are generalized by our main theorem.

In a simple undirected graph, a 2-matching is the edge-set of the node-disjoint union of cycles and paths. A 2-matching is called square-free if it contains no cycle of length 4. Putting effort into square-free 2-matchings is motivated by the observation that the complements of $(n - 3)$ -connected graphs are exactly the square-free 2-matchings. Hence one can establish a relation with a node-connectivity augmentation problem (see Frank [24]). It is worth mentioning that, replacing $(n - 3)$ by $(n - 2)$ we get a problem which is equivalent to the maximum matching problem. The first positive result on square-free 2-matchings is due to D. Hartvigsen [39], who provided a min-max formula in the bipartite case, i.e. a min-max formula to determine the maximum cardinality of a square-free 2-matching in a simple bipartite graph. Z. Király [45] proved a simplified version of this min-max formula in a purely combinatorial way. Hartvigsen’s original paper proposed an algorithmic approach, as well. Recently, Hartvigsen [40] verified this proposed algorithm, and turned it into an algorithmic proof of Király’s formula. These results provide a solution of the above mentioned connectivity augmentation problem when the complement of the graph is bipartite. The complexity of the non-bipartite case is still open.

A significant step towards a better understanding of restricted matchings is the interpretation of A. Frank [24], who represented square-free 2-matchings in bipartite graphs using the theory of covering pairs of sets. (For pairs of sets, see Frank, Jordán [25].) Using this interpretation, Frank also proposed other general min-max

¹This section has been published as TR-2005-14 in the Technical Reports series of the Egerváry Research Group.

formulas. For example, he proved an analogous formula for the maximum cardinality of a K_{tt} -free t -matching in a simple bipartite graph, for any fixed $t \geq 2$. For $t = 2$ this specializes to square-free 2-matchings. M. Makai [60] observed that Frank's formula holds for the following generalization. Suppose we are given an arbitrary bipartite graph, a positive integer t , and a pre-specified list \mathcal{K} of forbidden subgraphs isomorphic to K_{tt} . (Hence some K_{tt} -subgraphs may be forbidden, while others may be allowed.) Find a maximum cardinality t -matching avoiding subgraphs from the forbidden list. Makai cited general results on pairs of sets to prove a min-max formula for this \mathcal{K} -free t -matching problem. It is straightforward that the family \mathcal{K} in Makai's problem fulfill the condition (8), and thus Makai's observation follows from results in this section, as well. Using the theory of pairs of sets, Frank also proved a different kind of formula concerning large-bi-clique-free t -matchings. The class of large-bi-cliques does, however, not fulfill the condition (8), hence this result is not a special case of our main theorem.

The theory of covering pairs of sets, invented by A. Frank and T. Jordán [25], is a common generalization of a whole bunch of min-max formulas, including connectivity augmentation, Györi's theorem on intervals, and restricted matching problems. It was a great breakthrough when A. Benczúr and L. Végh [2, 3] constructed a polynomial running time algorithm for this general framework. Basic observations show that the problem considered in this paper is a special case of covering pairs of sets, and thus is solved by Benczúr and Végh's algorithm in polynomial time. The goal of this section is to show that restricted b -matchings can in fact be handled in an easier way than pairs of sets in general.

b -matchings

To solve the maximum restricted b -matching problem, it will be useful to give an account of basic results concerning (unrestricted) b -matchings, which is in fact a special case of the general problem. We will only define later what this general restricted b -matching problem is. For purposes in our general problem, we will require a tool for b -matchings which either provides an augmenting path, or provides a set which verifies optimality via the min-max formula. This section is devoted to the construction of this elementary tool.

Consider a bipartite graph $G = (A, B; E)$ with node set $V = A \cup B$ and a function $b : V \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$. A **b -matching** is a set $M \subseteq E$ of edges such that $\delta_M(v) \leq b(v)$ holds for all $v \in V$. The following good characterization of the maximum cardinality of a b -matching is well-known.

Theorem 1.4 *The maximum cardinality of a b -matching in a bipartite graph $G =$*

$(A, B; E)$ is equal to

$$(7) \quad \min_{Z \subseteq V} (b(V - Z) + |E[Z]|).$$

It is straightforward that the maximum is at most the minimum in this formula. Let us call a set Z a verifying set for a given b -matching M , if equality $|M| = b(V - Z) + |E[Z]|$ holds. Equality in the min-max formula is equivalent with the existence of a b -matching with a verifying set. Next we will prove Lemma 1.5 on alternating paths, the proof of the theorem is then immediate. Later, the Lemma 1.5 will be useful in the proof for restricted b -matchings, as well.

Consider an arbitrary b -matching $M \subseteq E$. A node $v \in V$ is called M -loose if $\delta_M(v) < b(v)$ holds. A path in G is called an M -alternating augmenting path, if it starts in an M -loose node in A , ends in an M -loose node in B , and its edges are alternatingly in $E - M, M$ with the first and last being in $E - M$. Clearly, the symmetric difference of M and the edge-set of an augmenting path is a b -matching of cardinality $|M| + 1$. The essence of the following lemma is that an augmenting path is all we have to be looking for to find a larger b -matching.

Lemma 1.5 *Given a b -matching M in a bipartite graph $G = (A, B; E)$, one can find in linear time either an augmenting M -alternating path, or a set Z for which $|M| = b(V - Z) + |E[Z]|$ holds.*

Proof. Let us define a digraph $D = (V, E')$ by orienting edges of M towards A , and orienting edges of $E - M$ towards B . There is a one-to-one correspondence between edges in E and arcs in E' . Notice that an augmenting path corresponds to a directed path in D starting in an M -loose node in A and ending in an M -loose node in B . Hence, if there exists an augmenting path, then one can construct an augmenting path in linear time by breadth first search. Otherwise, if there is no augmenting path, then we construct the set $S \subseteq V$ of nodes reachable in D from M -loose nodes in A . The construction of S can be performed in linear time by breadth first search starting at M -loose nodes in A . We claim that the set defined by $Z := S \Delta B$ satisfies $|M| = b(V - Z) + |E[Z]|$. To prove this, we partition M into four disjoint sets defined by $M_1 := M \cap \delta(A \cap Z, B \cap Z)$, $M_2 := M \cap \delta(A - Z, B \cap Z)$, $M_3 := M \cap \delta(A \cap Z, B - Z)$, $M_4 := M \cap \delta(A - Z, B - Z)$. By definition of S we get that $\delta_D^{out}(S) = 0$, which implies $M_1 = \delta(A \cap Z, B \cap Z) = E[Z]$ and $M_4 = \emptyset$. We have assumed that there is no augmenting path, which implies that all M -loose nodes are in Z . Thus $|M_2 \cup M_3| = b(V - Z)$. \square

Restricted b -matchings

Consider a simple bipartite graph $G = (A, B; E)$ with node set $V = A \cup B$ and a function $b : V \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$. Let \mathcal{K} be a family of some complete bipartite subgraphs (bi-cliques) of G with both classes consisting of at least two nodes. So for each $K \in \mathcal{K}$ we have $|A \cap V(K)| \geq 2$ and $|B \cap V(K)| \geq 2$. Let us point out that we only distinguish two subgraphs of G if their node-sets or edge-sets are not identical. Thus there may well be two isomorphic subgraphs only one of which is in \mathcal{K} . A b -matching M is called \mathcal{K} -free if no component of the subgraph $(A, B; M)$ is a member of \mathcal{K} . We consider the problem of maximizing the cardinality of a \mathcal{K} -free b -matching.

In Theorem 1.6, we propose a min-max formula to determine the maximum cardinality of a \mathcal{K} -free b -matching. It will be easy to check that the expression in the “minimum” is an upper bound. To prove equality, we make an assumption on the family \mathcal{K} . Equality does not follow without the assumption. We consider triples G, b, \mathcal{K} enjoying the following property.

$$(8) \quad \text{For every } K \in \mathcal{K}, \text{ we have } k := |A \cap V(K)| \geq 2, l := |B \cap V(K)| \geq 2, \\ b(v) = l \text{ for any } v \in A \cap V(K), \text{ and } b(v) = k \text{ for any } v \in B \cap V(K).$$

It is easy to check that those problems considered in the introduction obey (8). For example Makai’s list problem corresponds to the case when $b \equiv t$, and \mathcal{K} consists of a number of subgraphs isomorphic to K_{tt} .

To formulate the dual expression in our formula, we need some more notation. A component of an induced subgraph $G[Z]$ is called a \mathcal{K} -component of $G[Z]$ if it is a component of $G[Z]$, and member of \mathcal{K} . Let $c_{\mathcal{K}}(G[Z])$ denote the number of \mathcal{K} -components of $G[Z]$. Our main result of this section is the following characterization of the maximum cardinality of a \mathcal{K} -free b -matching.

Theorem 1.6 *Given a simple bipartite graph $G = (A, B; E)$, a function $b : A \cup B \rightarrow \mathbb{N}$ and a family \mathcal{K} of complete bipartite subgraphs of G so that the triple G, b, \mathcal{K} fulfills (8). Then the maximum cardinality of a \mathcal{K} -free b -matching is equal to*

$$(9) \quad \min_{Z \subseteq V} (b(V - Z) + |E[Z]| - c_{\mathcal{K}}(G[Z])).$$

We will see easily that the expression in the right hand side is an upper bound. Let us point out that this min-max formula is a particularly strong kind of characterization. To determine the value of the dual expression in (9), we need to test for membership in \mathcal{K} only a small number (at most $|A|/2$) of subgraphs of G . In some sense, the formula is a good characterization only assuming that \mathcal{K} is given by a membership oracle.

To prove this theorem, it seems an attractive idea to use an augmentation structure, such as an alternating forest. For unrestricted b -matchings, the augmentation

structure amounts to finding directed paths connecting loose nodes in an auxiliary digraph. An attractive approach to restricted b -matchings would be looking for directed paths in the same auxiliary digraph, trying to avoid that squares appear in the symmetric difference. D. Hartvigsen [40] constructed an algorithm from this idea. However, Hartvigsen’s algorithm has to deal with complicated issues, which yields a quite complicated augmentation structure. In our approach, we don’t deal with those complicated issues, and instead we go a different way to avoid squares in the b -matching.

Our approach differs from Hartvigsen’s quite significantly in the respect that we do not directly try to avoid squares to appear. Instead, whenever a square – the “first” square – appears, we stop there, and use that constellation for a reduction. The actual way and the circumstances of finding this specific constellation is described in the 3-Way Lemma 1.7. This constellation – a “fitting bi-clique” – is useful in the reduction of our problem to a smaller instance. In Lemma 1.8 we will prove that this reduction is an “equivalent reduction” in some sense – playing a similar role as the contraction of an odd cycle in Edmonds’ matching algorithm.

The constructive proof

In this subsection we prove Theorem 1.6 in such a way that our key lemmas will directly imply a polynomial running time algorithm. To prove the easy part, consider a \mathcal{K} -free b -matching M and a set $Z \subseteq V$. Then $|M[Z]| \leq |E[Z]| - c_{\mathcal{K}}(G[Z])$, since M does not contain all the edges of any member of \mathcal{K} . Moreover, $|M - M[Z]| \leq b(V - Z)$, since the edges in $M - M[Z]$ are incident with a node in $V - Z$. These two inequalities sum up to $|M| \leq b(V - Z) + |E[Z]| - c_{\mathcal{K}}(G[Z])$. We obtain the following slackness conditions – that is, equality $|M| = b(V - Z) + |E[Z]| - c_{\mathcal{K}}(G[Z])$ implies:

- (10) $\delta_M(v) = b(v)$ for all nodes $v \in V - Z$.
- (11) $M[Z]$ contains $E[Z]$ except for exactly one edge from each \mathcal{K} -component of $G[Z]$.

To formulate our key lemma, let us introduce the notion of a fitting bi-clique which, in some sense, is the analogue of alternating odd cycles in Edmonds’ matching algorithm. Consider a \mathcal{K} -free b -matching N . A bi-clique $K \in \mathcal{K}$ is called a **bi-clique fitting** N , if there is an edge $uw \in E(K)$ with $u \in A, w \in B$ such that $E(K) - uw \subseteq N$ and $\delta_N(u, B - V(K)) = \emptyset$. In words: N contains all but one of the edges of K , and at most one edge leaving the nodeset of K . A fitting bi-clique will be given by the four-tuple N, K, u, w . Our key lemma provides a method to find some certificate of optimality, or an augmentation, or a fitting bi-clique.

Lemma 1.7 (3-Way Lemma for \mathcal{K} -free b -matchings) *Let M be a \mathcal{K} -free b -matching. Then at least one of the following assertions holds.*

- (12) *There is a set Z such that $|M| = b(V - Z) + |E[Z]|$ holds.*
- (13) *There is a \mathcal{K} -free b -matching N such that $|N| = |M| + 1$.*
- (14) *There is a \mathcal{K} -free b -matching N such that $|N| = |M|$, and there is a bi-clique fitting N .*

Moreover, one can find in polynomial time either Z , or N , or the four-tuple N, K, u, w for one of the respective assertions.

Proof. If M is a maximum b -matching, then (12) follows from Theorem 1.4. Otherwise, by Lemma 1.5, there is an augmenting path P say with nodes in order $V(P) = \{v_0, z_0, v_1, z_1, \dots, v_k, z_k\}$. (This means that the path P has an edge-set of $2k+1$ edges $E(P) = \{v_0z_0, z_0v_1, \dots, v_kz_k\}$ such that $v_i \in A, z_i \in B, v_i z_i \in E - M, z_i v_{i+1} \in M$, where v_0, z_k are M -loose.) If the b -matching $M \Delta E(P)$ is \mathcal{K} -free, then assertion (13) holds. Otherwise we consider for $i = 0, 1, \dots, k$ the subpath P_{2i} on the nodes $V(P_{2i}) = \{v_0, z_0, \dots, v_i\}$, which is the subpath of P starting in $v_0 \in A$ and having $2i$ edges. Then $M_i := M \Delta E(P_{2i})$ is a b -matching for any i . Let j be the maximal index for which M_j is \mathcal{K} -free.

Firstly, suppose $j = k$. That is, M_k is a \mathcal{K} -free b -matching, but $M \Delta E(P) = M_k + v_k z_k$ is a b -matching which is not \mathcal{K} -free. Let $N := M_k$, $u := v_k$ and $w := z_k$, and let K be the unique $K \in \mathcal{K}$ such that $E(K) \subseteq M_k + v_k z_k$. Since $M_k + v_k z_k$ is a b -matching, and \mathcal{K} fulfills property (8), we get that $\delta_N(V(K)) = \emptyset$. This implies $\delta_N(u, B - V(K)) = \emptyset$, hence (14) holds.

Secondly, suppose $j < k$. That is, M_j is a \mathcal{K} -free b -matching, but $M_{j+1} = M_j + v_j z_j - z_j v_{j+1}$ is a b -matching which is not \mathcal{K} -free. Let $N := M_j$, and let K be the unique $K \in \mathcal{K}$ which is subgraph of M_{j+1} . Let $u := v_j$ and $w := z_j$. Since M_{j+1} is a b -matching, and \mathcal{K} fulfills property (8), we get that $\delta_{M_{j+1}}(V(K)) = \emptyset$. This implies $\delta_N(u, B - V(K)) = \emptyset$, hence (14) holds. \square

To formulate our reduction lemma, suppose there is a \mathcal{K} -free b -matching N , and a fitting bi-clique given by N, K, u, w . We define a ‘‘contracted graph’’ $G' = (A', B'; E')$ by deleting the edges in $E(K)$, identifying the nodes in $K \cap A$ by a new node k_A and identifying the nodes in $K \cap B$ by a new node k_B , only keeping one edge from each bunch of parallel edges to retain a simple bipartite graph. By definition, there is no edge joining the two new nodes. We define a function b' on V' which returns the value of b for the old nodes, and for the new nodes we define $b'(k_A) := b'(k_B) := 1$. Let \mathcal{K}' be the family of subgraphs in \mathcal{K} which are disjoint from $V(K)$. The above definitions make sense, since G', b', \mathcal{K}' fulfills condition (8). Let N' denote the image

of N in G' , i.e. we get N' by deleting all the $kl - 1$ edges in $E(K) \cap N$. It is easy to see that N' is a \mathcal{K}' -free b' -matching in G' of cardinality $|N| - kl + 1$.

Lemma 1.8 *If G', b', \mathcal{K}' is constructed as above, then both of the following claims hold.*

- (15) *If we are given a \mathcal{K}' -free b' -matching M' in G' , then one can construct in polynomial time a \mathcal{K} -free b -matching in G of cardinality $|M'| + (kl - 1)$.*
- (16) *If Z' is an inclusionwise minimal verifying set for N', G', b', \mathcal{K}' , then the pre-image Z of Z' is a verifying set for N, G, b, \mathcal{K} .*

Proof. To prove the first claim, let M'' denote the pre-image of M' . By definition, $b'(k_A) = b'(k_B) = 1$. Hence $|\delta_{M''}(A \cap V(K), B - V(K))| \leq 1$ and $|\delta_{M''}(A - V(K), B \cap V(K))| \leq 1$. Thus, by adding a properly chosen set of $kl - 1$ edges of $E(K)$ to M'' , we obtain a b -matching M in G . It is easy to check that M is \mathcal{K} -free. This completes the proof of the first claim.

To prove the second claim, we make use of complementary slackness conditions with respect to N', Z' . By definition, $\delta_{N'}(k_A) = 0$. Hence, condition (10) implies $k_A \in Z'$. No member of \mathcal{K}' contains k_A , hence by criterion (11), a maximum \mathcal{K}' -free b' -matching must contain all edges of $E'[Z']$ incident with k_A . However, the maximum \mathcal{K}' -free b' -matching N' contains no edge incident with k_A , thus there is no edge $k_A t \in E'$ with $t \in Z'$. Next we distinguish the two cases whether k_B is in Z' or not. Firstly, if $k_B \notin Z'$, then it is easy to see that $b(V - Z) = b'(V' - Z') + kl - 1$, $c_{\mathcal{K}}(G[Z]) = c_{\mathcal{K}'}(G'[Z'])$, and $|E[Z]| = |E'[Z']|$. Secondly, if $k_B \in Z'$, then $b(V - Z) = b'(V' - Z')$ and $|E[Z]| = |E'[Z']| + kl$ follow easily. We also claim $c_{\mathcal{K}}(G[Z]) = c_{\mathcal{K}'}(G'[Z']) + 1$, which needs a bit of reasoning: we must prove that there is no edge $k_A t \in E'$ or $k_B t \in E'$ such that $t \in Z'$. We have already seen that there is no edge $k_A t \in E'$ with $t \in Z'$. Suppose for contradiction that there is an edge $k_B t \in E'$ with $t \in Z'$. But then it is easy to see that $Z' - k_B$ is a verifying set, as well. This contradicts the minimal choice of Z' , thus we have proved $c_{\mathcal{K}}(G[Z]) = c_{\mathcal{K}'}(G'[Z']) + 1$. The following calculation can be verified in both of our cases. $|N| \leq b(V - Z) + |E[Z]| - c_{\mathcal{K}}(G[Z]) = b'(V' - Z') + |E'[Z']| - c_{\mathcal{K}'}(G'[Z']) + (kl - 1) = |N'| + (kl - 1) = |N|$. \square

Proof of Theorem 1.6. We prove sufficiency in formula (9) by induction on $|V|$. Consider a maximum \mathcal{K} -free b -matching M , and apply the 3-Way Lemma 1.7 for G, b, \mathcal{K}, M . Now, assertion (13) is impossible. If assertion (12) holds, then we are done, since Z is a verifying set for M . If assertion (14) holds, then we use N, K, u, w to construct G', b', \mathcal{K}', N' , and apply Lemma 1.8. By the first claim of Lemma 1.8, N' is a maximum \mathcal{K}' -free b' -matching. Thus, by induction, there is a verifying set Z' for N' . By the second claim of Lemma 1.8 there is a verifying set Z for N . \square

Algorithmic proof of Theorem 1.6. A polynomial time algorithm is given just as easily. We maintain G, b, \mathcal{K}, M throughout the algorithm, and apply 3-Way Lemma 1.7 for G, b, \mathcal{K}, M . When (12) holds, then optimality is achieved and certified by set Z . When (13) holds, then we replace G, b, \mathcal{K}, M by G, b, \mathcal{K}, N , which is an augmentation. When (14) holds, then we apply the procedure recursively for G', b', \mathcal{K}', N' , that is, we will either find an augmentation, or a verifying set with respect to G', b', \mathcal{K}', N' . Lemma 1.8 implies that an augmentation with respect to G', b', \mathcal{K}', N' can be used to augment with respect to G, b, \mathcal{K}, M , moreover, a verifying set with respect to G', b', \mathcal{K}', N' can be used to construct a verifying set with respect to G, b, \mathcal{K}, M . (For the latter notice that, given a verifying set, one can easily find an inclusionwise minimal verifying set just by dropping some nodes as long as we retain a verifying set. Also, we have to remark that, a membership oracle for \mathcal{K}' is easy to derive from a membership oracle for \mathcal{K} – we just have to check whether the specific bi-clique is disjoint from k_A, k_B , or not.) \square

Remarks

- Let us sketch the complexity analysis of the algorithm. We initiate the algorithm from a maximum $(b - 1)$ -matching, which is guaranteed to be \mathcal{K} -free. Then the above algorithm performs at most $|V|$ augmentations. Between two augmentations, we find at most $|V|$ fitting bi-cliques. We construct $|V|^2$ reduced graphs, and there we perform $|V|^2$ breadth-first-searches. Thus we have to test at most $|V|^3$ subgraphs for membership in \mathcal{K} . Thus the above algorithm has a total running time of $O(|V|^4)$. (The most time-consuming operation is the construction of the reduced graphs, hence one may hope for a slightly better running-time by applying a nice data-structure to represent merged nodes. Here we omit the discussion of such data-structures.)
- An interesting observation is that our algorithm only uses a membership oracle of \mathcal{K} , even though \mathcal{K} is an arbitrary subset of the potentially exponential set of all complete bipartite subgraphs fulfilling (8). Thus the algorithm will finish the job without having tested the membership of most of those subgraphs. (In contrast, the hypo-matching problem is intractable, if the family of allowed hypo-matchable subgraphs is only given by a membership oracle.)

1.3 Non-bipartite matching

Non-bipartite matching is the most developed area of all those combinatorial optimization problems where parity plays a significant role. Our efforts dealing with non-bipartite matching will be useful in front of those more general problems, such as even factor, A -paths, and matroid matching. Classical papers provide us with an enormous background in this area, where some of the milestones are Tutte's [84] good characterization of perfectly matchable graphs, the Berge-Tutte Formula (Berge, [4]) on the maximum cardinality of a matching, and Edmonds' algorithm to find a maximum matching. In order to facilitate the transition to more general frameworks, we try to develop a comfortable algorithmic framework which is flexible to generalize. Here, one should not hesitate playing with well-known algorithmic principles, like Edmonds' alternating forest and blossoms. Our contribution to non-bipartite matching is a variation of Edmonds' algorithm, developed in this section. The crucial difference is that we avoid building up the well-known structure of blossoms and alternating forests. Instead we use a simple recursive principle to search for a larger matching, or a certificate of maximality.

We consider an undirected graph $G = (V, E)$, and for technical reasons we allow G to have parallel edges and loops. A component of a graph is called odd if it has an odd number of nodes, and $c(G)$ denotes the number of odd components of G .

Theorem 1.9 (Berge-Tutte Formula) *In an undirected graph $G = (V, E)$ the maximum cardinality $\nu(G)$ of a matching is determined by*

$$(17) \quad \nu(G) = \min_{Z \subseteq V} (|V| + |Z| - c(G - Z)) / 2.$$

For the easy part, notice that $\nu(G) \leq (|V| - c(G))/2$, and $\nu(G) \leq \nu(G - Z) + |Z|$ holds for any $Z \subseteq V$. A set $Z \subseteq V$ is called a **verifying set** for M if equality $2|M| = |V| + |Z| - c(G - Z)$ holds. To prove equality in the Berge-Tutte Formula, we have to find a matching with a verifying set. The 3-Way Lemma for matching considers the special case when there is a verifying set Z such that all the odd components of $G - Z$ are isolated nodes.

To formulate the 3-Way Lemma we have to recall a couple of well-known notions from matching theory. For a matching M , nodes in $V - V(M)$ are called **M -exposed**. An odd cycle $C \subseteq E$ is called an **M -alternating odd cycle** if $|C \cap M| = (|C| - 1)/2$ and C is incident with an M -exposed node. So, for example a loop on an M -exposed node is an M -alternating odd cycle. Let $is(G)$ denote the number of isolated nodes in G . Notice that $is(G) \leq c(G)$.

Lemma 1.10 (3-Way Lemma for matching) *Consider an undirected graph G*

with a matching M . At least one of the following alternatives holds:

- (18) There is a set $Z \subseteq V$ such that $2|M| = |V| + |Z| - \text{is}(G - Z)$.
- (19) There is a matching N with $|N| = |M| + 1$.
- (20) There is a matching N with $|N| = |M|$ s.t. there is an N -alternating odd cycle C in G .

Moreover, one can find in polynomial time N , or Z , or N, C for one of the respective alternatives.

First proof. The proof is by induction on $|V|$. First, check whether $Z = \emptyset$ provides equality in alternative (18). Otherwise, there is an M -exposed node a which is not an isolated node in G . Then at least one of the following cases holds.

There is a loop on a . In this case alternative (20) holds.

There is an edge $ab \in E$ with b M -exposed. In this case alternative (19) holds for $N := M + ab$.

There is an edge $ab \in E$ with b covered by some edge $bc \in M$. This is what we called a V-configuration. We construct the V-reduction, also depicted in Figure 1. That is, we define $G' = (V', E') := (G - b) / \{a, c\}$ (delete b and identify a and c by a new node $q := \{a, c\}$). At this point it is essential that we keep possible parallel edges, loops. Let $M' := M - bc$. Clearly, M' is a matching in G' . By induction, one of the alternatives holds for G', M' . We conclude by showing that the same alternative also holds for G, M .

Suppose alternative (18) holds for G', M' , say with a set $Z' \subseteq V'$. Since the new node q is M' -exposed, we get that $q \notin Z'$, and q is an isolated node of $G' - Z'$. Define $Z := Z' + b$. Then a and c are isolated nodes in $G - Z$. Thus, $|V| + |Z| - \text{is}(G - Z) = (|V'| + 2) + (|Z'| + 1) - (\text{is}(G' - Z') + 1) = 2|M'| + 2 = 2|M|$.

Suppose alternative (19) holds for G', M' , i.e. there is a matching N' with $|N'| = |M'| + 1$. Let N'' be the pre-image of N' in G . Then N'' is a matching such that at least one of the nodes a, c is N'' -exposed. Thus at least one of $N := N'' + ab$ or $N := N'' + ac$ is a matching in G .

Suppose alternative (20) holds for G', M' , i.e. there is a same-size matching N' with an N' -alternating odd cycle C' in G' . Let C'', N'' denote the pre-images of C', N' , respectively. At least one of $N := N'' + ab, N := N'' + bc$ is a matching. If $q \notin V(C')$ then $C := C''$ is an N -alternating odd cycle, where N is one of $N'' + ab, N'' + bc$. If $q \in V(C')$, then either C'' or $C'' + ab + bc$ is an odd cycle, we denote it by C . We can choose $N := N'' + ab$ or $N := N'' + bc$ such that C is an N -alternating odd cycle. (See Figure 2.) \square

We give a second proof, since it directly extends to path-matching and even factors considered in a later section.

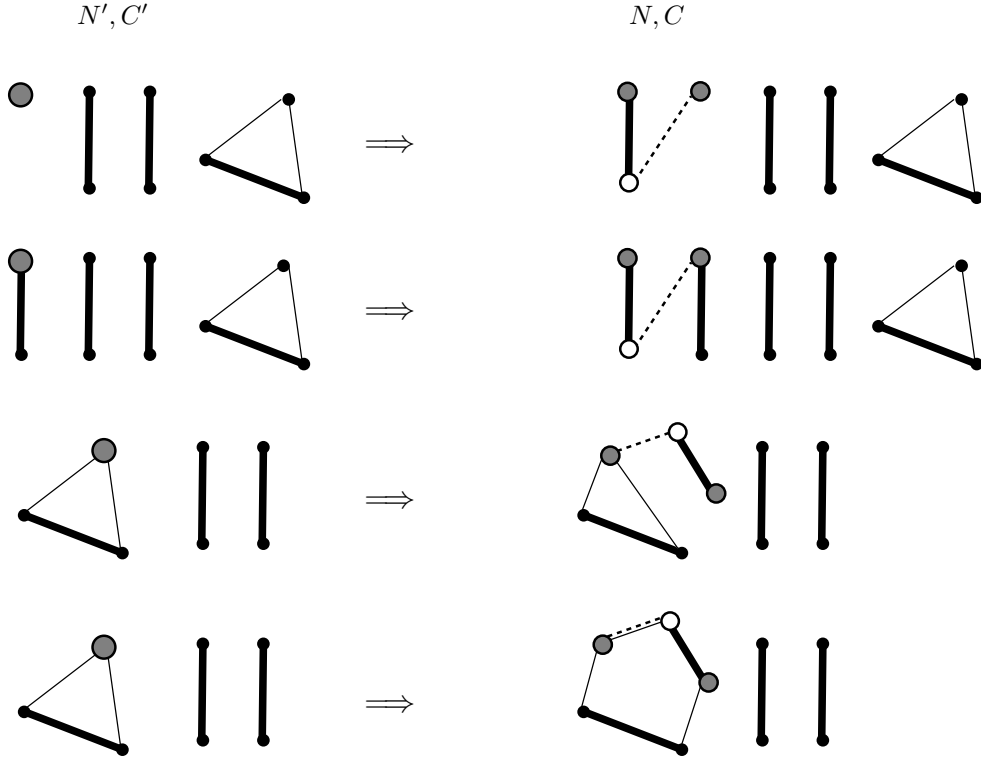


Figure 2: The construction of N, C from N', C' in the first proof.

Second proof of Lemma 1.10. We define an auxiliary bipartite graph $G' = (V' \cup V'', E')$ by $V' := \{v' : v \in V\}$, $V'' := \{v'' : v \in V\}$ and $E' := \{v'z'' : vz \in E\}$. Then $M' := \{v'z'' : vz \in M\}$ is a matching of size $2|M|$. Consider König's Theorem in bipartite graph G' . Then either there is a set $X \subseteq V'$ such that $|M'| = |\Gamma_{G'}(X)| + |V' - X|$, or there is a larger matching in G' .

In the former case, let $Z := \{z \in V : z'' \in \Gamma_{G'}(X), z' \notin X\}$ and $Q := \{z \in V : z'' \notin \Gamma_{G'}(X), z' \in X\}$. It is easy to see that all the nodes in Q are isolated nodes in $G - Z$, hence $is(G - Z) \geq |Q|$. Notice that $|Q| - |Z| = |X| - |\Gamma_{G'}(X)|$. This implies that $2|M| = |V| + |Z| - is(G - Z)$, thus alternative (18) holds.

In the latter case, there is an M' -alternating augmenting path P , that is a path $(v'_0, z''_0, \dots, v'_k, z''_k)$ such that $v'_0 \in V' - V(M')$, $z''_k \in V'' - V(M')$, $v'_i z''_i \in E' - M'$, and $z''_i v'_{i+1} \in M'$. Consider the walk in G defined by the pre-image of P , which is $(v_0, z_0, \dots, v_k, z_k)$. This walk has the following properties. Its first and last nodes are M -exposed, and its edges are alternately in and not in M . It traverses every node at most twice – i.e. the nodes z_i are pairwise distinct and the nodes v_i are pairwise distinct. Such a walk is called an M -alternating augmenting walk. If this walk is a path (i.e. an M -alternating augmenting path) then the symmetric difference of M and the edgeset of the walk produces a larger matching, thus alternative (19) holds. Otherwise, consider the first node w in the walk which is traversed a second time.

Since the walk is an M -alternating walk, we get that there are indices $j \leq m$ such that $w = v_j = z_m$. Our choice of j, m implies that $P := \{v_0z_0, \dots, z_{j-1}v_j\}$ is an even path, and $C := \{v_jz_j, \dots, v_mz_m\}$ is an odd cycle. Thus alternative (20) holds for $N := M\Delta P$, and C . \square

Lemma 1.11 *Consider an undirected graph G , a matching N , and an N -alternating odd cycle C . Then both of the following assertions hold.*

- (21) *If we are given a matching M in G/C larger than N/C , then one can construct a matching M' in G larger than N .*
- (22) *If Z is a verifying set for N/C in G/C , then $\{C\} \notin Z$, and Z is a verifying set for N in G .*

Proof. The first assertion is proved by the well-known concept of expanding C .

To see the second assertion, notice that $\{C\}$ is N/C -exposed. Hence the slackness condition implies that $\{C\}$ must be in an odd component of $G/C - Z$. So, the odd components in $G - Z$ are exactly the pre-images of the odd components of $G/C - Z$. Thus, $|N| = |N/C| + (|C| - 1)/2 = (|V/C| + |Z| - c(G/C - Z))/2 + (|C| - 1)/2 = (|V| + |Z| - c(G - Z))/2$, we are done. \square

Proof of the Berge-Tutte Formula. The proof is by induction on the number of nodes. Consider an arbitrary maximum matching M , and apply the 3-Way Lemma 1.10. If (18) holds, then we are done. If (19) holds, then $M + ab$ is a matching, which is a contradiction. If (20) holds, then construct $G/C, N/C$ as above. By the first part of Lemma 1.11, N/C is a maximum matching in G/C . By induction there is a verifying set Z' for M' with respect to G' . By the second part of Lemma 1.11, there is a verifying set Z for M with respect to G . \square

Algorithmic proof of the Berge-Tutte Formula. We maintains a pair G, M of a graph and a matching, and recursively, we find either a larger matching or a verifying set. First we apply Lemma 1.10 for G, M , and find N , or Z , or N and C for one of the respective alternative. If alternative (19) holds, then N is a larger matching, and we are done. If alternative (18) holds, then Z is a verifying set, and we are done. If alternative (20) holds, then notice that N/C is a matching in G/C . Reduce to the pair $G/C, N/C$, and apply the above procedure for $G/C, N/C$. Recursively, we either obtain a verifier Z for N/C , or larger matching in G/C . Then, by Lemma 1.11, we can lift this to graph G to construct either a larger matching, or a verifying set. \square

Remarks

- Here is a running-time calculation of a trivial implementation. We augment at most $O(|V|)$ times, and contract at most $O(|V|^2)$ odd cycles. Performing the 3-Way Lemma via the second proof, and constructing a contracted graph takes $O(|V| + |E|)$ time. Thus, the total running time is $O(|V|^2(|V| + |E|))$.
- In our second proof of the 3-Way-Lemma we concluded that path P , and odd cycle C compose a so-called M -alternating blossom. This configuration is in fact a very special case of Edmonds alternating forest, with a single blossom.

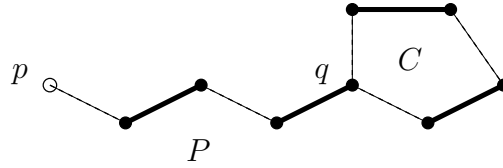


Figure 3: An M -alternating blossom.

- A maximum 2-matching can be found in polynomial time as it reduces to bipartite matching. Hence, the following seems a good idea for a non-bipartite algorithm.

“Find a maximum 2-matching in an undirected graph G . If it has no odd cycles, then a maximum matching is given by decomposing its even cycles into a matching. Otherwise there is a least one odd cycle, let C be one of them. Recursively, find a maximum matching N in G/C . Expand N to a matching M of cardinality $|N| + |C| - 1$. Output M .”

This idea is indeed attractive, but unluckily, it does not work out to find a maximum matching. However, our matching algorithm works quite similarly to this proposal. The difference is that we maintain not only the graph G/C , but we also maintain a matching in the current graph.

- Let us compare our matching algorithm with Edmonds’. The most important similarity is that in both algorithms we contract odd cycles. The crucial difference lies within the subroutine of finding these odd cycles. Edmonds’ algorithm builds up the global structure of an alternating forest, and maintains that after the contraction. In contrast, our algorithm will find the odd cycle by a subroutine for the 3-Way-Lemma, constructed in either of our two proofs. The subroutine of the first proof is based on the recursive application of the

V-reduction. The subroutine of the second proof is based on alternating paths in the auxiliary bipartite graph.

- The following lemma reflects the idea behind the 3-Way from another point of view, that is, with respect to half-integral matchings. It is well-known that the polytope of fractional matchings is half-integral, and the problem of finding a maximum half-integral matching is easy reducing to bipartite matching. Moreover, extremal fractional matchings decompose into the disjoint union of edges with 1's, and odd cycles of $\frac{1}{2}$'s. These odd cycles are said to be “used” by that fractional matching. The 3-Way Lemma arises as a special case of the following lemma by setting $k = 1$.

Lemma 1.12 *Suppose that x is an extremal fractional matching the size of which is not maximal, and that x uses k odd cycles. Then there is a larger size extremal fractional matching using at most $k + 1$ odd cycles. \square*

1.4 Even factor in odd-cycle-symmetric directed graphs

In this section we construct a polynomial time algorithm to find a maximum even factor in odd-cycle-symmetric digraphs. The main result is a constructive proof of Theorem 1.15 of L. Szegő and the author, generalizing results of W.H. Cunningham and J.F. Geelen on path-matching and even factor.

The even factor problem is a generalization of path-matching, which in turn is a generalization of non-bipartite matching; the story behind this is the following. Cunningham and Geelen were working on a strongly polynomial time separation algorithm for the matchable set polytope, for which Balas and Pulleyblank, [1] gave a nice description. For this separation algorithm, they needed a polynomial time subroutine for a discrete subproblem. This subproblem is equivalent with an interesting matching-type problem in an undirected graph, which Cunningham and Geelen [17] called path-matching problem. The perfect path-matching problem is that, given an undirected graph, and two disjoint same-size terminal sets, find a perfect path-matching, i.e. a family of node-disjoint paths linking the terminal sets, such that the deletion of these paths leaves a perfectly matchable graph. The corresponding optimization is finding a maximum value path-matching, for the definition, see [17]. They constructed a maximum path-matching algorithm via ellipsoid method, and thus settled the complexity of the above separation problem, as well. They proved a combinatorial min-max formula for path-matching, where the dual expression is concerned with “stable pairs”. Shortly thereafter, Frank and Szegő [27] gave a simple proof of an equivalent min-max formula, and Spille and Szegő [81] provided a Gallai-Edmonds-type structural description. Then, Cunningham and Geelen [18] constructed another polynomial time algorithm by investigating integer substitutions of the Tutte-matrix. It follows from this latter algebraic approach that path-matching induces a matroid on the nodeset, which is a generalization of the matching matroid. It seems path matching is in many regards similar to non-bipartite matching, except for the algorithmic aspect, that is. For non-bipartite matching, one always has Edmonds’ famous algorithm in mind, see [21], where he grows an alternating forest in search of an alternating path. Thus one wonders if, and to what extent Edmonds’ approach extends to path-matching. Spille and Weismantel [82] attempted constructing such an algorithm, which first looks appealing, but then one gets stuck with a number of technical issues when branches of the alternating forest get scrambled. There must be a simpler way extending Edmonds’ approach to path-matching, and this is set as the major goal of this section.

W.T. Tutte [84] used a matrix over a field-extension of the rationals to deal with non-bipartite matching, and proved his famous good characterization of the existence of a perfect matching. The Tutte-matrix $T_G \in \mathbb{R}^{V \times V}$ is given for a simple undirected graph $G = (V, E)$, and is defined by $T_G(u, v) := 0$ for $uv \notin E$,

$T_G(u, v) := -T_G(v, u) := x_{uv}$ for $uv \in E$. Here, x_{uv} are algebraically independent indeterminates over the rationals. It is quite easy to prove that this matrix is nonsingular if and only if G has a perfect matching, and indeed, $\text{rank}(T_G) = 2\nu(G)$ holds. We remark that this matrix is skew-symmetric, and such matrices play a fundamental role in combinatorial optimization elsewhere, for example the linear delta-matroids are represented by skew-symmetric matrices. It is tempting to construct a polynomial time algorithm to determine the rank of matrices over field-extensions of the rationals, and thus solving all those special cases, including maximum matching, path-matching, even factor, linear matroid matching, 3D generic rigidity, parity constrained strong orientations, etc. We remark that a randomized polynomial time algorithm follows from substituting the indeterminates with random integers, see Lovász [51]. However, it is unknown whether there is a deterministic algorithm determining the rank of such matrices, even some of those special cases are still unsolved, not to mention that the solutions of the solved ones is quite interesting on their own. Thus we have to be careful when dealing with problems of this kind, especially when looking for a good characterization, or a deterministic algorithm. Hence, it was a breakthrough when Cunningham and Geelen [18] constructed a matching algorithm by running through substitutions of the Tutte matrix. This algorithm deterministically assigns small integer number to variables in the Tutte matrix, and gets closer to a maximum rank substitution by altering the value of some of its indeterminates. Having the path-matching problem in mind, they observed that the scheme of substitutions in their matching algorithm works out for this, as well. Moreover, they observed that one can go a step further, and generalize this algorithm to even factors, which they introduced in [19]. Given a digraph $D = (V, A)$, an even factor is the arcset of the disjoint union of directed paths and directed even cycles, and the problem is to find a maximum cardinality even factor. They mentioned that the maximum even factor problem is intractable, but it is tractable for a special class of digraphs called weakly symmetric. A digraph is called weakly symmetric if arcs induced in a strongly connected component are symmetric, i.e. the reverse of that arc is in the digraph, too. Equivalently, arcs in every directed cycle are required to be symmetric. Szegő and the author proposed that it is enough requiring that the digraph be odd-cycle-symmetric (i.e. only arcs in odd cycles to be symmetric) and proposed the even factor formula for odd-cycle-symmetric digraphs. We remark that the algebraic methods of Cunningham and Geelen extend to odd-cycle-symmetric digraphs, thus providing a polynomial time algorithm. The author [66], however, constructed a combinatorial algorithm for even factors, which is presented below. We will add a section afterwards on hypo-matchings in digraphs, which is a common generalization of this, and hypo-matching.

Our constructive approach below is basically the straightforward generalization

of our non-bipartite matching algorithm, and thus is related to Edmonds' algorithm. Edmonds' original algorithm is not easy to generalize to even factor; the concept of an alternating forest of blossoms is not flexible enough. The author, while trying to extend Edmonds' original algorithm, encountered various technical problems, and then decided to go a different way, presented below. The constructive proof below is in fact slightly simpler than the original non-constructive proof of Szegő and the author in [74].

Path-cycle-matchings

Our main goal is to solve the maximum even factor problem, but first we develop a small theory of path-cycle-matching. There is nothing new about path-cycle-matching, since they are equivalent with matching in a bipartite graph. The point is that we formulate a consequence of alternating paths in the bipartite graphs, which will be very useful to deal with even factors in the following section.

Consider a directed graph $D = (V, A)$. An arc-set $M \subseteq A$ is called a **path-cycle-matching** if $\delta_M^{in}(v), \delta_M^{out}(v) \leq 1$ for all $v \in V$. The set $\{v_i v_{i+1} : 1 \leq i \leq n+1\} \subseteq A$ of arcs is called a **path** if the nodes v_i ($1 \leq i \leq n+1$) are pairwise distinct. It is called a **cycle** if the nodes v_i ($1 \leq i \leq n$) are pairwise distinct, and $v_{n+1} = v_1$. It is easy to see that path-cycle-matchings are those arc-sets which arise as a fully node disjoint union of paths and cycles. The maximum cardinality of a path-cycle-matching reduces to bipartite matching, this provides the following min-max formula.

Theorem 1.13 *The maximum cardinality of a path-cycle-matching in a directed graph $D = (V, A)$ is determined by*

$$(23) \quad \min_{Z \subseteq V} (|V| + |\Gamma_D^+(Z)| - \sigma_1(D[Z])),$$

where $\Gamma_D^+(Z) := \{x \in V - Z : \exists y \in Z, yx \in A\}$, and $\sigma_1(D[Z])$ denotes the number of source-nodes in the induced subgraph $D[Z]$.

Proof. We reduce path-cycle matchings in $D = (V, A)$ to matching in an auxiliary bipartite graph $G = (V', V''; E)$. G is defined by $V' := \{v' : v \in V\}$, $V'' := \{v'' : v \in V\}$, $E := \{a'b'' : ab \in A\}$. Clearly, images of the path-cycle-matchings in D are exactly the matchings in G . From König's Theorem 1.1 we get that the maximum cardinality of a matching in G is equal to $|\Gamma_G(Z')| + |V' - Z'|$ for some set $Z' \subseteq V'$. Define $Z := \{z : z' \in Z'\}$. It is easy to see that $|\Gamma_G(Z')| + |V' - Z'| = |V| + |\Gamma_D^+(Z)| - \sigma_1(D[Z])$. \square

Our algorithmic approach is based on the following Lemma, where we formulate the analogue of an augmenting alternating path for path-cycle-matchings. Alternating walks provided by this Lemma will be useful to manipulate path-cycle-matchings.

Suppose M, N are path-cycle-matchings such that $|N| > |M|$. Consider the bipartite graph $G = (V', V''; E)$ constructed in the proof of Theorem 1.13. We use $'$ to denote the image of an arc-set. Then M', N' are matchings in G , so there exists an M' -alternating augmenting path $P \subseteq M' \Delta N'$. That is, P consists of $2k + 1$ edges $P = \{a'_i b''_i : i = 0, \dots, k\} \cup \{a'_{i+1} b''_i : i = 0, \dots, k - 1\}$, where a'_i are $k + 1$ distinct nodes in V' and b''_i are $k + 1$ distinct nodes in V'' , moreover $P \cap M' = \{a'_{i+1} b''_i : i = 0, \dots, k - 1\}$, and a'_0, b''_k are M' -exposed. Hence we get the following Lemma.

Lemma 1.14 *Suppose M is a path-cycle-matching the cardinality of which is not maximum. Then there is an M -alternating augmenting walk, that is a sequence of $2k + 1$ arcs $W = \{a_i b_i : i = 0, \dots, k\} \cup \{a_{i+1} b_i : i = 0, \dots, k - 1\} \subseteq A$, where a_i are $k + 1$ distinct nodes in V , b_i are $k + 1$ distinct nodes in V , $a_i b_i \in A - M$, and $a_{i+1} b_i \in M$. \square*

Even factors

We define notions in order to formulate the even factor problem. A cycle $C \subseteq A$ is called **symmetric** if for any arc $uv \in C$ there is an arc $vu \in A$. Otherwise, C is called **asymmetric**. A cycle or a path is called **even/odd** by the number of its arcs. A digraph is called **odd-cycle-symmetric**, if all odd cycles are symmetric. A path-cycle-matching is called an **even factor** if all of its cycles are even. Let $\nu(D)$ denote the maximum cardinality of an even factor. A set $S \subseteq V$ is called a **source-component** in digraph D , if $D[S]$ is strongly connected and $\delta_D^{in}(S) = 0$. Let $\sigma_{odd}(D[Z])$ denote the number of source-components S in $D[Z]$ such that $|S|$ is odd. These are called **odd source-components**. For example, a source-node is an odd source-component, thus $\sigma_1(D[Z]) \leq \sigma_{odd}(D[Z])$.

The author and Szegő [74] proved the following theorem. Here we provide a constructive proof, which also implies a polynomial time algorithm.

Theorem 1.15 (Even Factor Formula, Pap, Szegő [74]) *Let $D = (V, A)$ be an odd-cycle-symmetric digraph. Then*

$$(24) \quad \nu(D) = \min_{Z \subseteq V} |V| + |\Gamma_D^+(Z)| - \sigma_{odd}(D[Z]).$$

Proof of easy part of Theorem 1.15. Consider a set $Z \subseteq V$ and an even factor M . From $\delta_M^{in}(v) \leq 1$ we get $|\{uv \in M : u \in X, v \in V - Z\}| \leq |\Gamma_D^+(Z)|$. From $\delta_M^{out}(v) \leq 1$ we get $|\{uv \in M : u \in V - Z\}| \leq |V| - |Z|$. If $|S|$ is odd, then $|M[S]| \leq |S| - 1$. This implies $|\{uv \in M : u, v \in Z\}| \leq |Z| - \sigma_{odd}(D[Z])$. These inequalities add up to $|M| \leq |V| + |\Gamma_D^+(Z)| - \sigma_{odd}(D[Z])$. \square

A set $Z \subseteq V$ is called a **verifying set** for an even factor M if $|M| = |V| + |\Gamma_D^+(Z)| - \sigma_{\text{odd}}(D[Z])$. The proof of Theorem 1.15 will be completed by finding an even factor with a verifying set. If some even factor has a verifying set, then each of the three inequalities proved in the easy part must hold with equality. This implies the following “slackness condition”: If M is an even factor with a verifying set Z , then $\delta_M^{\text{out}}(v) = 1$ holds for all $v \in V - Z$.

We say that an **an odd cycle C fits even factor N** , if $|N \cap C| = |C| - 1$ and $\delta_N^{\text{out}}(V(C)) = 0$.

Lemma 1.16 (3-Way Lemma for even factors) *Let $D = (V, A)$ be an odd-cycle symmetric directed graph, and let M be an even factor in D . At least one of the following alternatives holds.*

(25) *There is a set $Z \subseteq V$ such that $|M| = |V| + |\Gamma_D^+(Z)| - \sigma_1(D[Z])$.*

(26) *There is an even factor N with $|N| = |M| + 1$.*

(27) *There is an even factor N with $|N| = |M|$ and an odd cycle C fitting N .*

Moreover, one can find in polynomial time N , or Z , or N, C for one of the respective alternatives.

Proof. If M is a maximum path-cycle-matching, then by Theorem 1.13 there is a set X for alternative (25). Otherwise, there is an M -alternating walk given by Lemma 1.14, let us use the notation from that lemma. Let $W_j = \{a_i b_i : i = 0, \dots, j\} \cup \{a_{i+1} b_i : i = 0, \dots, j-1\}$, and $M_j := M \Delta P_j$. Then $M \Delta W$ and M_j (for all $j = 0, \dots, k$) are path-cycle-matchings.

Case I. Suppose that every M_j is an even factor. So, in particular, M_k is an even factor. Moreover, $M_k + a_k b_k$ is a path-cycle-matching. If $M_k + a_k b_k$ is an even factor, then alternative (26) holds. Otherwise there is a unique odd cycle C with $a_k b_k \in C \subseteq M_k + a_k b_k$. It is easy to check that alternative (27) holds for C , $N := M_k$.

Case II. Consider the smallest index j such that M_j is not an even factor. Since $M_0 = M$ is an even factor, $j \geq 1$. So M_{j-1} is an even factor. $M_j = M_{j-1} + a_{j-1} b_{j-1} - a_{j+1} b_j$. Thus, there is a unique odd cycle C with $a_{j-1} b_{j-1} \in C \subseteq M_j$. It is easy to see that alternative (27) holds for C , $N := M_{j-1}$.

This proof also provides an algorithm: We construct the auxiliary bipartite graph G , and we apply an arbitrary bipartite matching algorithm. Thus we find the verifying set, or an augmenting walk. We exploit the augmenting walk in one of the cases I or II to find a larger even factor, or a fitting odd cycle. \square

Lemma 1.17 (Contracting odd cycles) *Suppose N is an even factor in the odd-cycle-symmetric digraph $D = (V, A)$, and C is an odd cycle in D such that N fits C . Then each of the following assertions hold.*

- (28) D/C is odd-cycle-symmetric, and N/C is an even factor.
- (29) If we are given an even factor in D/C larger than N/C , then one can construct an even factor in D larger than N .
- (30) If Z' is a verifying set for N/C in D/C , then $\{C\} \in Z'$ and its pre-image $Z := Z' - \{C\} + V(C)$ is a verifying set for N in D .

Proof. N contains a path-segment inside C of $|C| - 1$ arcs, which is even. This implies that N/C is an even factor. To prove that D/C is odd-cycle-symmetric, consider an odd cycle C_1 in D/C . If $\{C\} \notin V(C_1)$, then C_1 is symmetric, since $D/C - \{C\}$ is isomorphic to $D - V(C)$. Suppose that $\{C\} \in V(C_1)$. Let C_2 denote the pre-image of C_1 in D . Then C_2 is a path from $t \in V(C)$ to $s \in V(C)$, say. Since C is a symmetric odd cycle in D , there must be an even path P_{st} from s to t in $D[V(C)]$. Then $C_2 \cup P_{st}$ is an odd cycle in D . So all arcs in C_2 must be symmetric in D . Thus, all arcs in C must be symmetric in D/C , too.

For the second assertion (29) we need to prove the following statement. Given an even factor N_1 in D/C one can construct an even factor in D of cardinality $|N_1| + |C| - 1$. To prove this, let N_2 denote the pre-image of N_1 in D . Then N_2 contains at most one arc entering $V(C)$, let this arc be $s's$ if any, otherwise choose $s \in V(C)$ arbitrarily. Similarly, N_2 contains at most one arc leaving $V(C)$, let this arc be tt' if any, otherwise choose $t \in V(C)$ arbitrarily. Since D is odd-cycle-symmetric, there is a set $N_{st} \subseteq D[V(C)]$ of arcs such that $|N_{st}| = |C| - 1$, and it is the disjoint union of two-arc cycles and an even-length s - t path P_{st} . It is easy to see that $N_2 \cup N_{st}$ is a path-cycle-matching of cardinality $|N_1| + |C| - 1$. We claim that $N_2 \cup N_{st}$ is an even factor, in fact. We need to check whether there is some odd cycle in $N_2 \cup N_{st}$. Clearly, there is no odd cycle inside N_{st} or inside M_2 . If C' was an odd cycle in $N_2 \cup N_{st}$, then $s's + P_{st} + tt' \subseteq C'$. Thus C'/C is a cycle in N_1 . If C'/C is asymmetric, then C' is asymmetric. If C'/C is even, then C' is even.

To prove the third assertion, notice that by definition, $\delta_{N/C}^{out}(\{C\}) = 0$. From the slackness condition we get that $\{C\} \in Z'$. Notice, if $\{C\} \in S'$ for some odd source-component S' of $D[Z']$, then its pre-image $S := S' - \{C\} + V(C)$ is a odd source-component of $D[Z]$. This implies $\sigma_{odd}(D[Z']) = \sigma_{odd}(D/C[Z])$. Assertion (30) follows from the following calculation:

$$\begin{aligned} |N| &= |N/C| + |C| - 1 = \nu(D/C) + |C| - 1 = |V/C| + |\Gamma_{D/C}^+(Z')| - \\ &\quad - \sigma_{odd}(D/C[Z']) + |C| - 1 \geq |V| + |\Gamma_D^+(Z)| - \sigma_{odd}(D[Z]) \geq |N|. \end{aligned}$$

□

Proof of the Even Factor Formula. The proof is by induction on the number of nodes. Consider a maximum even factor M , and apply the 3-Way Lemma 1.16. If (25) holds, then we are done. If (26) holds, then N is a larger even factor, which is a contradiction. If (27) holds, then construct $D/C, N/C$ as above. By Lemma 1.17, N/C is a maximum even factor in the odd-cycle-symmetric D/C . By induction there is a verifying set Z' for N/C with respect to D/C . By the second part of Lemma 1.17, there is a verifying set Z for M with respect to D . \square

Algorithmic proof of the Even Factor Formula. We maintain a pair D, M of an odd-cycle-symmetric digraph and an even factor, and recursively, we find either a larger even factor or a verifying set. We apply Lemma 1.10 for D, M , and find N , or Z , or N and C for one of the respective alternative. If alternative (19) holds, then Z is a verifying set, and we are done. If alternative (18) holds, then we augment. If alternative (20) holds, then notice that N/C is an even factor in D/C . Recursively, we either obtain a verifying set Z' for N/C , or even factor in D/C larger than N/C . Then, by Lemma 1.11, we can lift this to digraph D , and thus find either a larger even factor, or a verifying set. \square

Conclusions and examples

- M -alternating walks are easier to handle than M -alternating paths, since the concatenation of two M -alternating walks is an M -alternating walk, but the analogue statement does not hold for M -alternating paths. This may be the reason why our proof of the 3-Way-Lemma was so simple.
- In Figures 4 and 5, the orientation of arcs is not displayed, but will be quite clear from the context. The set of bold lines on the left represents an even factor M , the dashed line represents an M -alternating walk W , and the symmetric difference $M\Delta W$ is displayed on the right hand side. These examples demonstrate how the odd cycle is extracted from the pattern of an M -alternating walk. For example, in Figure 4 we see three odd cycles emerging from the symmetric difference, and C will be our choice in the proof of the 3-Way-Lemma. In Figure 5, the long odd cycle appearing on the right hand side has nothing to do with the odd cycle C chosen in our proof of the 3-Way-Lemma. The reason is that C appears earlier, after composing the symmetric difference with a starting segment of W . These cycles C are subject to a contraction in the next step of the even factor algorithm.

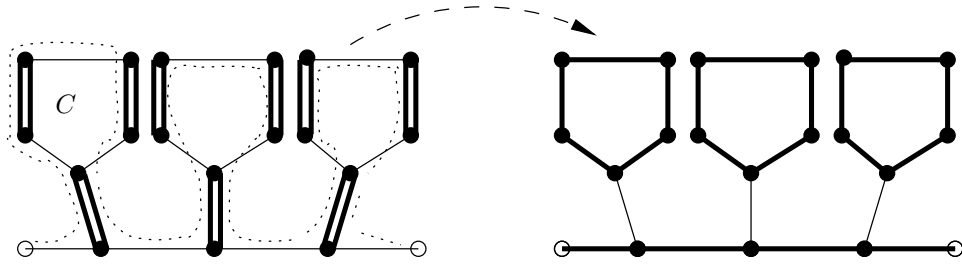


Figure 4:

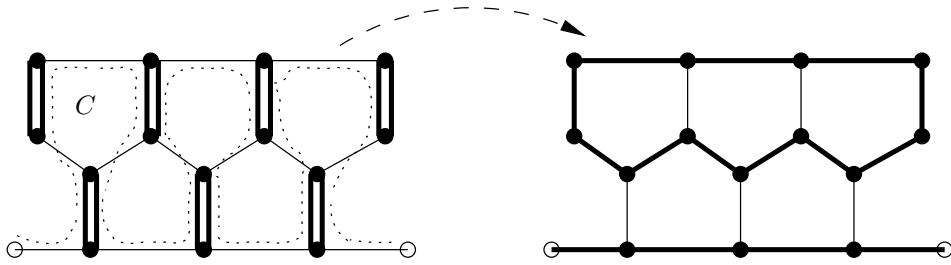


Figure 5:

- Why don't we construct a truly Edmonds-type algorithm for even factors? Spille and Weismantel [82] went that way to construct a path-matching algorithm, which is a special case of even factors. They proposed an involved definition of alternating forests and blossoms for path-matching, but then encountered various technical difficulties. This setback motivated me to go a slightly different way in solving the path-matching and the even factor problem. The reason of little success of alternating forests is, in my mind, the following. If we stick to Edmonds' original idea, then the alternating forest should be maintained after the contraction of an odd cycle C . While this is doable for matching, the same goal seems difficult for path-matching or even factor, since the contraction of an odd cycle may completely destroy the alternating forest. Considering ideas from Spille, Weismantel, and some of my own, I don't see a way to avoid this self-destruction to happen.

1.5 Hypo-matching in directed graphs

In this section we describe a common generalization of results on hypo-matching of Cornuéjols, Hartvigsen and Pulleyblank [12, 13, 14, 15] and results on even factor of Cunningham and Geelen [17] and the author and Szegő [74].

The maximum hypo-matching problem is the following. An undirected graph is called **factor-critical** (or hypo-matchable) if the deletion of any of its nodes produces a perfectly matchable graph. Suppose we are given an undirected graph G and a family \mathcal{F} of factor-critical subgraphs of G . A **hypo-matching** in G is a subgraph the components of which are either members of \mathcal{F} or isomorphic with K_2 . The problem is to maximize the number of nodes covered by a hypo-matching. For results on this problem, see papers of G. Cornuéjols, D. Hartvigsen and W. Pulleyblank [12, 13, 14, 15], where a hypo-matching formula is shown and a polynomial time algorithm is developed. In [50] M. Loeb and S. Poljak gave another elegant constructive proof of the hypo-matching formula, which uses the Edmonds-Gallai decomposition of G .

Recall some definitions from the previous section concerning even factors. The main result on even factors is the min-max formula in Theorem 1.15, which determines the maximum cardinality of an even factor in odd-cycle-symmetric digraphs. We remark that the author and Szegő [75] proposed a generalization of the even factor formula for the following problem. Suppose we are given an odd-cycle-symmetric digraph, and a family \mathcal{L} of directed odd cycles. The problem is to find a maximum cardinality set of arcs which is the node-disjoint union of directed paths, directed even cycles, and cycles in \mathcal{L} . This result is a common generalization of one the one hand, the even factor formula, and on the other hand, the special case of the hypo-matching formula concerning a set \mathcal{F} of odd cycles.

The hypo-matching problem in directed graphs is defined as follows.

Definition 1.18 *Consider a digraph $D = (V, A)$ which may have loops or parallel arcs. A digraph is called **symmetric-critical** if all its arcs are symmetric and the underlying undirected graph is factor-critical. Let \mathcal{H} be a family of symmetric-critical subgraphs in D . An **\mathcal{H} -matching** in $D = (V, A)$ is a subset M of A so that for each component (V_0, M_0) of (V, M) , either M_0 is a path, or M_0 is an even cycle, or M_0 is an asymmetric odd cycle, or (V_0, M_0) is a member of \mathcal{H} . The **size** of M is defined as the number of arcs in all these paths and cycles plus the number of nodes covered by members of \mathcal{H} used in M . A hypo-matching M is called **perfect** if $\text{size}(M) = |V|$. Let $\nu^{\mathcal{H}}(D)$ denote the maximum size of an \mathcal{H} -matching in D .*

To define the dual expression in the min-max formula, we have to consider the following definitions.

Definition 1.19 *A symmetric-critical induced subgraph $D[U]$ is called \mathcal{H} -critical if there is no perfect \mathcal{H} -matching in $D[U]$. For a set $Z \subseteq V$, let $\sigma_{\mathcal{H}}(D[Z])$ denote the number of those source-components in $D[Z]$ which are \mathcal{H} -critical.*

Thus the main result on hypo-matching in digraphs is the following.

Theorem 1.20 ([70, 72]) *If $D = (V, A)$ is a digraph and \mathcal{H} is a family of symmetric-critical subgraphs of D , then*

$$(31) \quad \nu^{\mathcal{H}}(D) = \min_{Z \subseteq V} (|V| + |\Gamma_D^+(Z)| - \sigma_{\mathcal{H}}(D[Z])).$$

The proof given in [70, 72] is composed of ideas of the proof on even factor in the previous section, and ideas of Cornuéjols, Hartvigsen, Pulleyblank on hypo-matching in undirected graphs. We should also remark that the proof is constructive, assuming an oracle which, given a symmetric-critical subgraph, either confirms that it is \mathcal{H} -critical, or returns a perfect hypo-matching in that subgraphs. It follows from results of Cornuéjols, Hartvigsen, Pulleyblank that such an oracle may be constructed for example if \mathcal{H} consists of odd cycles of length at least k , or of odd cycles of length at most k , where k is fixed.

2 Matroid matching and related problems

A brief historical survey

Matroid matching was proposed by Lawler [49] as a common generalization of matroid intersection and non-bipartite matching. A variation of this problem is that, given a matroid on a groundset partitioned into pairs (2-element sets), find a maximum independent set from a union of pairs. A slightly more general problem is the polymatroid matching problem, where we are looking for a maximum even vector in a polymatroid. This reduces to matroid matching in the “pre-matroid” over the expanded groundset. Lovász [55], and Jensen, Korte [42] pointed out that the time complexity of matroid matching is exponential, assuming the matroid is given by an independence oracle. Thus we turn our attention towards hopefully broad classes of (poly)matroids in which the matching problem is tractable.

Matroid matching became a powerful tool in combinatorial optimization when Lovász [51, 52, 54] proved a min-max formula and constructed a polynomial time algorithm for matching in a linearly representable matroids, given by an explicit linear representation. The most famous application of this result is a polynomial time algorithm for packing \mathcal{A} -paths, which follows from the linear representation given by Schrijver [77]. Lovász developed a theory of matroid matching which resulted in the polynomial time solution of other problems, including the problem of finding a maximum triangle cactus in a graph, and pinning down a minimum number of nodes of a graph in \mathbb{R}^2 to obtain a rigid framework. Subsequent papers proposed other linear matroid matching algorithms, see Gabow, Stallmann [31], and Orlin, Vande Vate [63]. Vande Vate [87] proved a Gallai-Edmonds-type structural description.

Lovász’ min-max formula was extended to superclasses of linear matroids, such as algebraic matroids by Dress, Lovász [23], and pseudomodular matroids by Björner, Lovász [6], and by Hochstättler, Kern [41]. An important observation was that these classes share a combinatorial property, the so-called “double-circuit property”, DCP for short. Lovász’ proof could be extended to DCP matroids, see [51, 23].

Geelen, Iwata, Murota [34] proposed and solved the linear delta-matroid matching problem, a generalization of linear matroid matching.

As opposed to the above abstract classes of matroids, one could also consider special cases and concrete application. As a matter of fact, a wide range of combinatorially meaningful matroids are known to be linearly representable, and thus Lovász’ linear matroid matching theorem applies. Unluckily, for some of these interesting special cases, an explicit representation is difficult to construct. This is the case for example for parity-constrained strongly connected orientations. Moreover, a combinatorial characterization is difficult to extract from Lovász’ linear al-

gebraic min-max formula. Hence it is a great challenge to prove a combinatorial min-max formula, and construct an efficient deterministic algorithm. Such results were proved by Szigeti [83] on the maximum triangle-cactus problem, and by Frank, Jordán, Szigeti [26] on parity constrained rooted- k -arc-connected orientations. A related result is Nebeský’s [62] characterization of a maximum genus graph embedding, which is equivalent with parity constrained rooted-connected orientations. Recently, T. Király, Szabó [44], and Makai, Szabó [59] proved further generalizations.

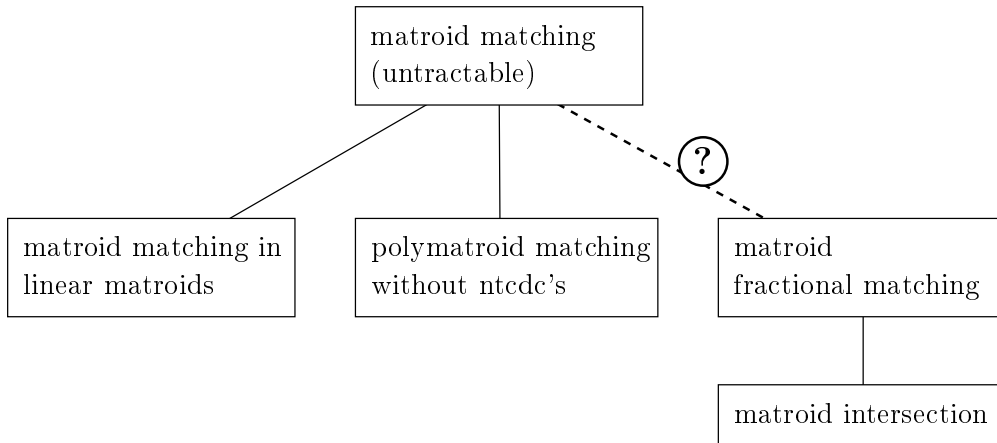


Figure 6: Tractable problems in relation with matroid matching.

Outline of this chapter

In this chapter we consider those problems in Figure 6 related to matroid matching. As already mentioned above, matroid matching and polymatroid matching are intractable. We consider two tractable classes of matching problems, which together cover most known applications. These classes are: matroid matching in linearly representable matroids, and polymatroid matching in ntdc-free polymatroids. As an introduction to the linear matroid matching algorithm, we first discuss a matroid intersection algorithm. In the end, we discuss a fractional relaxation of matroid matching, which is shown tractable for arbitrary matroids.

Our approach to matroid intersection is the direct generalization of our approach to bipartite matching in the first chapter.

For linear matroid matching, we discuss Orlin and Vande Vate’s algorithm based on the representation of so-called “very strong covers” by matroid intersection. We propose an equivalent, but slightly different recursive scheme. We also discuss the question why the size of rational numbers does not explode while performing this algorithm for matroids represented over the rationals. An answer to this technical

question is easily given from well-known properties of very strong covers, but is essential in the polynomial bound of the running-time.

Then we consider a special class of polymatroid matching problems, for which Makai and Szabó [59] proved a min-max formula, the so-called Partition Formula. For this, we construct a semi-strongly polynomial time algorithm. This applies to special cases such as maximum genus graph embedding, and rooted- k -arc-connected orientation.

Finally we survey results on matroid fractional matching, a fractional relaxation of matroid matching, a generalization of graph fractional matching, and matroid intersection. As displayed in Figure 6 by the dashed line and the question-mark, it is unknown whether matroid fractional matching reduces to a special case of matroid matching. The main result on matroid fractional matching is Vande Vate's [86] min-max formula, and Gijswijt's [36] result on total dual half-integrality.

An interesting observation is the following. We prove min-max formulae for those three matching problems mentioned above, and joined by a line to matroid matching in Figure 6. The dual expression of these min-max formulae depends on a projection and a partition for linear matroid matching, only on a partition for ntcde-free polymatroids, and only a projection for matroid fractional matching. A challenging research area is that of trying to understand the connection between these three min-max formulae, and why some matching problems only depend on a partition, or only on a projection.

2.1 Matroid intersection

Matroid intersection is one of the most famous problems in combinatorial optimization, which has been studied via various methods and generalizations. This is our opportunity to describe the algorithmic framework which we have successfully used previously for matching problems. We hope that this algorithmic framework will bring more light and a better understanding to all those generalizations of matroid intersection, as well.

Edmonds [22] proved that matroid intersection admits a min-max formula, and also constructed a polynomial time algorithm. There is a wide range of related results on polyhedra, maximum weight intersection, polymatroid intersection, matroid union. Here we will focus on the computational aspect of matroid intersection.

Our algorithmic framework is somewhat different from Edmonds' original matroid intersection algorithm, which looks for alternating paths in an auxiliary digraph, where we take every element of the groundset into consideration. Quite contrary to this, our concept exploits "local" observations and constructions, like finding and projecting fundamental circuits. Here we make good use of a kind of projections, a lesser-known matroid operation which decreases the rank of subset to one. In a general step of the algorithm, given a common independent set, we pick an element the addition of which stays independent in one of the matroids, but not in the other one. Then, we contract the induced fundamental circuit in one of the matroids, and project it in the other matroid. The crucial observation will be that via this operation we retain two matroids with a given common independent set the maximality of which is equivalent with maximality in original problem. We remark that, although our approach seems quite different to Edmonds' in the first place, these approaches are closely related. The idea behind the above general step is that it takes care of the arcs in Edmonds' auxiliary digraph starting in the element we have picked. The advantage of our approach is that we don't bother shortest paths in a digraph for which we have to overview the whole matroid. Our framework is composed of very simple subroutines, and we only maintain very little, in some sense the smallest amount of information between these subroutines. The disadvantage clearly is that, since we drop useful information between these subroutines, we will not achieve the best running-time possible. However, and this is our main objective here, we propose a simple framework which exploits only the most important features of matroid intersection, and thus we may analyze this framework for generalizations, too.

We conclude that projections and contractions of matroids provide a proof of Edmonds' matroid intersection formula. This proof is more or less constructive: a polynomial time algorithm follows by a small de-tour to Rado's formula. Contractions of arbitrary matroids are not easy to determine algorithmically! To resolve

this uncomfortable situation, the crucial observation is that the class of partition matroids is closed under taking projections of circuits. Hence, a polynomial time algorithm follows for the intersection of a partition matroid and an arbitrary matroid, which is the subject of Rado’s theorem. This implies a polynomial time algorithm for the intersection of two arbitrary matroids by exploiting a simple reduction principle.

Basics

We assume the reader is familiar with basic matroid theoretical notions from Welsh [88]. We recall some well-known notation and important constructions concerning matroids. A matroid is given by $M = (S, \mathcal{I})$, where \mathcal{I} is the family of all independent sets, and $r = r_M$ denotes its rank-function. Consider an arbitrary set $Z \subseteq S$.

- **Deletion of Z .** $M - Z := (S - Z, \{X \subseteq S - Z : X \in \mathcal{I}\})$.
- **Dual.** $M^* = (S, \mathcal{I}^*)$ is the matroid such that its bases are exactly the complements of the bases of M . Its rank-function is given by $r^*(X) = |X| - r(S) + r(S - X)$.
- **Contraction of Z .** $M/Z = (S - Z, \mathcal{I}_{M/Z}) = (M^* - Z)^*$. Its rank-function is equal to $r_{M/Z}(Y) := r(Y \cup Z) - r(Z)$. For any maximum independent subset X of Z we have $\mathcal{I}_{M/Z} = \{Y \subseteq S - Z : Y \cup X \in \mathcal{I}\}$. Deletions and contractions of disjoint sets “commute”, i.e. the resulting matroid is independent of the order in which we perform these operations.

Edmonds’ and Rado’s Theorems

Edmonds’ matroid intersection formula determines the maximum cardinality of a common independent set of two matroids. Using his well-known concept of alternating paths, Edmonds also constructed a polynomial time algorithm to find a maximum common independent set, together with a certificate of its maximality.

Theorem 2.1 (Edmonds, [22]) *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be matroids, with rank-functions r_1, r_2 . Then the maximum cardinality of a set in $\mathcal{I}_1 \cap \mathcal{I}_2$ is equal to*

$$(32) \quad \min_{U \subseteq S} (r_1(U) + r_2(S - U)).$$

A number of problems are known to be equivalent with matroid intersection, for our purposes it will be useful to consider the special case where one of the matroids

is assumed a partition matroid with capacity 1 for each member of the partition. Consider a set S partitioned into a family of disjoint sets $\mathcal{S} = \{S_1, \dots, S_n\}$. A **partial system of representatives** (with respect to \mathcal{S}) – **psr**, for short – is a set containing at most one element of any S_i . The family of psr’s is the family of independent sets of a matroid, namely of a partition matroid with capacities 1 for each S_i .

Theorem 2.2 (Rado, [76]) *Let $M = (S, \mathcal{I})$ be a matroid with rank-function r , and let \mathcal{S} be a partition of the groundset S . The maximum cardinality of an independent psr is equal to*

$$(33) \quad \min_{\mathcal{Z} \subseteq \mathcal{S}} (r(\bigcup \mathcal{Z}) + |\mathcal{S}| - |\mathcal{Z}|).$$

It is well-known that Edmonds’ formula is polynomial time equivalent with Rado’s formula – for sake of self-containedness, we include the below “folklore” proof of equivalence.

Proof of equivalence of theorems 2.2 and 2.1. Suppose Edmonds’ Theorem 2.1 holds. Let M_1 be the partition matroid with the psr’s being the independent sets, and let $M_2 := M$. Let r_1 and $r_2 = r$ be the rank-functions of M_1 and M_2 , respectively. $r_1(U)$ is just the number of members of the partition hit by U . It is easy to see that M -independent psr’s are just the common independent sets of M_1, M_2 . By Theorem 2.1 the maximum cardinality of a common independent set is equal to $r_1(U) + r_2(S - U)$ for some set $U \subseteq S$. Let \mathcal{Z} denote the family of sets in \mathcal{S} which are disjoint from U ; and let $U' := S - \bigcup \mathcal{Z}$. Clearly $U \subseteq U'$, thus $r_1(U) + r_2(S - U) \geq r_1(U') + r_2(S - U') = |\mathcal{S}| - |\mathcal{Z}| + r(\bigcup \mathcal{Z})$.

Suppose Rado’ Theorem 2.2 holds. Let $S' := \{s' : s \in S\}$ be a disjoint copy of S , and let M'_2 be the copy of M_2 . Consider Rado’s theorem with groundset $S \cup S'$, the matroid $M := M_1^* \oplus M'_2$, and the partition $\{\{s, s'\} : s \in S\}$. Let γ denote the maximum cardinality of an independent psr. By Rado’s Theorem $\gamma = |S - T| + r_1^*(T) + r_2(T)$ holds for some $T \subseteq S$. Consider an independent psr $X' \subseteq S \cup S'$ of cardinality γ . Then $X' = \{x' : x \in X_1\} \cup \{x' : x \in X_2\}$ for some disjoint sets $X_1, X_2 \subseteq S$ such that $X_1 \in \mathcal{I}_1^*$ and $X_2 \in \mathcal{I}_2$. Consider a base B_1 of M_1 disjoint from X_1 . $|B_1 \cap X_2| \geq |X_1| + |X_2| + |B_1| - |S| = \gamma + r_1(S) - |S| = |S - T| + r_1^*(T) + r_2(T) + r_1(S) - |S| = |T| + r_1(S - T) - r_1(S) + r_2(T) + r_1(S) - |T| = r_1(S - T) + r_2(T)$. \square

In the sequel we propose a constructive proof of Edmonds’ Theorem, which uses a sequence of projections defined below. Unluckily, it is non-trivial to evaluate the rank-function of the composition of a sequence of projections – in fact, this evaluation is equivalent with matroid intersection. However, the class of partition matroids

in Rado's Theorem is closed under taking projections, thus, our constructive proof implies a polynomial time algorithm for Rado's Theorem. Then, the reduction from Edmonds' to Rado's Theorem implies a polynomial time algorithm for matroid intersection algorithm in general.

Projections

In this section we define projections, which construct matroids from another matroids. Let $M = (S, \mathcal{I})$ be a matroid $M = (S, \mathcal{I})$ with rank-function r , and consider an independent set $U \in \mathcal{I}$. The **projection** of U is given as $M^U = (S - U, \mathcal{I}^U)$, which is defined by

$$\mathcal{I}^U := \bigcup \{ \mathcal{I}_{M/(U-y)-y} : y \in U \}.$$

with rank-function given by

$$r^U(Y) = \max \{ r_{M/(U-y)}(Y) : y \in U \}.$$

To prove that this definition gives a matroid indeed, notice that \mathcal{I}^U contains the emptyset, and is closed under taking subsets. What remains is to check the exchange axiom, that is, for any two sets $Y, Z \in \mathcal{I}^U$, $|Y| < |Z|$ there is an element $a \in Z - Y$ such that $Y + a \in \mathcal{I}^U$. Say $Y \cup (U - y), Z \cup (U - z) \in \mathcal{I}$. Apply the exchange axiom for $Y \cup (U - y), Z \cup (U - z)$! If $y = z$, then we are done. If $y \neq z$, then the axiom provides an element $a \in (Z - Y) \cup \{y\}$ such that $Y \cup (U - y) \cup \{a\} \in \mathcal{I}$. If $a \neq y$, then we are done. Otherwise we get $Y \cup U \in \mathcal{I}$, thus $Y \cup (U - z) \in \mathcal{I}$, and we are back to the case $y = z$ from above.

A constructive proof of Edmonds' Theorem

We prove Edmonds's Theorem 2.1 on matroid intersection by induction on $|S|$. Consider an arbitrary common independent set $X \subseteq S$. Our goal is finding a larger common independent, or finding a set $U \subseteq S$ certifying its maximality. We say U is a **certificate** if $|X| = r_1(U) + r_2(S - U)$ holds.

Case 1. If $|X| = r_1(S)$, then $U := S$ is a certificate.

Case 2. Otherwise, consider some $a \in S - X$ for which $X + a \in \mathcal{I}_1$. If $X + a \in \mathcal{I}_2$, then $X + a$ is a larger common independent set.

Case 3. Otherwise, let $C = C_2(X, a) \subseteq X + a$ be the unique M_2 -circuit in $X + a$. We define two matroids on the groundset $S' := S - C$ by $M'_1 = (S', \mathcal{I}'_1) := M_1^C$, and $M'_2 = (S', \mathcal{I}'_2) := M_2/C$. Let r'_1, r'_2 denote their rank-functions. Clearly, $X' := X - C$ is a common independent set of M'_1, M'_2 . The inductive hypothesis implies that,

either there is a larger common independent set, or there is a certificate U' . We claim that any of these two cases holds, then its analogue also holds for M_1, M_2, X .

Claim 2.3 *Using the notation from above, both of the following assertions hold.*

(34) *If $Y \in \mathcal{I}'_1 \cap \mathcal{I}'_2$ then $Y \cup (C - y) \in \mathcal{I}_1 \cap \mathcal{I}_2$ for some $y \in C$.*

(35) *If $|X'| = r'_1(U') + r'_2(S' - U')$ holds for some $U' \subseteq S'$, then $|X| = r_1(U') + r_2(S - U')$.*

Proof. The first assertion follows from the definitions. For the second assertion, notice that $r'_1(U') = |X' \cap U'| = |X \cap U'|$. Next, we prove $r_1(U') = r'_1(U')$. Suppose for contradiction that $(X \cap U') \cup \{u\} \in \mathcal{I}_1$ for some $u \in U' - X$. For $(X \cap U') \cup C$ and $(X \cap U') \cup \{u\}$ the exchange axiom implies $(X \cap U') \cup \{u\} \cup (C - y) \in \mathcal{I}_1$ for some $y \in C$. Thus $r'_1(U') \geq |(X \cap U') \cup \{u\}| = |X \cap U'| + 1$. This contradiction proves $r_1(U') = r'_1(U')$. We conclude by $|X| = |X'| + (|C| - 1) = r'_1(U') + r'_2(S' - U') + (|C| - 1) = r_1(U') + r_2(S - U')$. \square

Remarks

- Assuming we are able to construct M'_1 defined above, we construct a matroid intersection algorithm. Following the lines of the above proof, we maintain a triple M_1, M_2, X , where X is a common independent set of the matroids M_1, M_2 . In case 1 we find a certificate, we are done. In case 2 we augment, that is we step to triple $M_1, M_2, X+a$. In case 3 we reduce to triple M'_1, M'_2, X' , where we recursively find a certificate or an augmentation. By Claim 2.3, a certificate or an augmentation can be lifted to the original triple M_1, M_2, X .
- Unluckily, it is not easy to construct M'_1 . In fact, for an algorithm, we would need to construct a sequence of reductions $M_1 \rightarrow M'_1$, and an independence oracle of such matroids is based on matroid intersection. However, this is not the case for partition matroids considered in Rado's Theorem, since the class of those partition matroids is closed under the projection of circuits.
- The analogue of the 3-Way Lemma is trivial, and implicit in the above proof, where we distinguish the cases 1,2,3.
- Assuming Edmonds's theorem, Claim 2.3 can be reformulated as follows. X is a maximum common independent set with respect to M_1, M_2 if and only if X' is a maximum common independent set with respect to M'_1, M'_2 .

A constructive proof of Rado's Theorem

In this section we describe a constructive proof of Rado's Theorem 2.2 which implies a polynomial time algorithm. We will find a maximum independent psr and a certificate of its maximality. We have proved the equivalence of theorems of Edmonds and Rado, thus the below results also imply a polynomial time algorithm for matroid intersection.

We prove Rado's Theorem 2.2 by induction on $|S|$. Consider a matroid $M = (S, \mathcal{I})$ with its groundset S partitioned into a family $\mathcal{S} = \{S_1, \dots, S_n\}$ of disjoint sets. For technical reasons, we allow that some S_i 's are empty, i.e. a couple of emptysets may be listed in \mathcal{S} . Consider an arbitrary independent psr $T \in \mathcal{I}$. Our goal is finding an independent psr larger than T , or finding a family $\mathcal{Z} \subseteq \mathcal{S}$ certifying its maximality. We say $\mathcal{Z} \subseteq \mathcal{S}$ is a **certificate** if $|T| = r(\bigcup \mathcal{Z}) + |\mathcal{S}| - |\mathcal{Z}|$ holds.

Case 1. If T is a maximum psr, then $\mathcal{Z} := \{S_i \in \mathcal{S} : S_i = \emptyset\}$ is a certificate.

Case 2. Otherwise $T_0 := T + s$ is a psr for $s \in S_n$, say. If T_0 is independent, then we are done.

Case 3. Otherwise let $C = C(T, s)$ be the unique circuit in T_0 . Let $S' := S - C$, $S_C := \bigcup \{S_i : S_i \cap C \neq \emptyset\} - C$, and $\mathcal{S}' := \{S_i : S_i \cap C = \emptyset\} \cup \{S_C\}$. Define $M' = (S', \mathcal{I}') := M/C$, say with rank-function r' . Clearly, $T' := T - C \subseteq S'$ is an independent psr with respect to \mathcal{S}', M' . The inductive hypothesis implies that there is a larger independent psr, or there is a certificate \mathcal{Z}' . We claim that a larger independent psr, or a certificate may be constructed for M, \mathcal{S}, T , too. More precisely:

Claim 2.4 *Using the notation from above, both of the following assertions hold.*

(36) *If T'' is an independent psr (with respect to M', \mathcal{S}'), then $T'' \cup (C - y)$ is an independent psr (with respect to M, \mathcal{S}) for some $y \in C$.*

(37) *If for some $\mathcal{Z}' \subseteq \mathcal{S}'$ we have $|T'| = r'(\bigcup \mathcal{Z}') + |\mathcal{S}'| - |\mathcal{Z}'|$, then $S_C \in \mathcal{Z}'$ and for $\mathcal{Z} := \mathcal{Z}' - \{S_C\} \cup \{S_i : S_i \cap C \neq \emptyset\}$ we have $|T| = r(\bigcup \mathcal{Z}) + |\mathcal{S}| - |\mathcal{Z}|$.*

Proof. By definition, $T'' \cup (C - y)$ is independent for any $y \in C$, and $T'' \cup (C - y)$ is a psr for at least one $y \in C$. This proves the first claim. The second claim is also easy to see. T' is disjoint from S_C , which implies $S_C \in \mathcal{Z}'$. Thus $\bigcup \mathcal{Z} = \bigcup \mathcal{Z}' \cup C$, and we get $|T| = |T'| + |C| - 1 = r'(\bigcup \mathcal{Z}') + |\mathcal{S}'| - |\mathcal{Z}'| + |C| - 1 = r(\bigcup \mathcal{Z}) + |\mathcal{S}| - |\mathcal{Z}|$. \square

Remarks

- The above proof implies a polynomial time algorithm. Maintain a triple M, \mathcal{S}, T , where T is an independent psr. Follow the lines of the above proof. In case 1 we find a certificate, we are done. In case 2 we augment, that is we step to triple $M, \mathcal{S}, T + s$. In case 3 we reduce to triple M', \mathcal{S}', T' , and recursively find an augmentation or a certificate. By Claim 2.4, a certificate or an augmentation may be lifted to the original triple M, \mathcal{S}, T . The algorithm performs at most $r(S)$ augmentations, and at most $|S|$ reductions between any two augmentations.
- The analogue of the 3-Way Lemma is trivial, and implicit in the above proof, where we distinguish the cases 1,2,3.
- Assuming Rado's theorem, Claim 2.4 can be reformulated as follows. T is a maximum independent psr with respect to M, \mathcal{S} if and only if T' is a maximum independent psr with respect to M', \mathcal{S}' .

Gallai-Edmonds decomposition for matroid intersection

In this section we prove Theorem 2.5, a strengthening of Edmonds' Theorem 2.1. This will be useful in the following section, where we discuss a linear matroid matching algorithm. Let

$$W_1 := \bigcap \{sp_1(X) : X \text{ a maximum common independent set of } M_1, M_2\}.$$

Theorem 2.5 *The maximum cardinality of a common independent set of matroids M_1, M_2 is exactly $r_1(W_1) + r_2(S - W_1)$. Moreover, W_1 is the unique inclusionwise minimal set attaining this value.*

Proof. Let $\alpha := \min_{U \subseteq S} r_1(U) + r_2(S - U)$, which by Edmonds' Theorem is equal to the maximum cardinality of a common independent set. By submodularity, the family $\mathcal{X} := \{U : r_1(U) + r_2(S - U) = \alpha\}$ is closed under taking unions and intersections. Consider the unique inclusionwise maximal set $U \in \mathcal{X}$.

Consider a maximum common independent set X . By Edmonds' Theorem, $|X| = r_1(U) + r_2(S - U)$. This implies $r_1(U) = |X \cap U|$ and thus $U \subseteq sp_1(X)$. We conclude that $U \subseteq W_1$.

Consider an arbitrary element $x \in W_1$. Then the maximum cardinality of a common independent set of the matroids M_1/x and $M_2 - x$ is $\alpha - 1$. Consider a set $U' \subseteq S - x$ such that $r_{M_1/x}(U') + r_{M_2-x}(S - x - U') = \alpha - 1$. This implies $r_1(U' + x) + r_2(S - (U' + x)) = \alpha$. Thus $x \in U$, and we conclude that $U = W_1$. \square

2.2 Linear matroid matching

In this section we present the linear matroid matching algorithm of Orlin and Vande Vate [63], based on a matroid intersection representation of a structured sub-problem called a “very strong cover”. Though we stick to the basic ideas developed in the aforementioned paper, we propose a different framework of recursions, and some simplifications. This algorithm generalizes Edmonds’ non-bipartite matching algorithm in the sense that bipartite matching is a special case of matroid intersection, and Edmonds’ alternating forest may be represented by bipartite matching between the inner and outer nodes of the forest. Similarly, the algorithm of Orlin, Vande Vate is “growing” the very strong cover by adding lines, or building hypo-matchable components. The success of this approach hangs upon a precise definition of these analogous notions.

We assume a matroid is given by the explicit representation over some specific field, and arithmetic operations over that field can be performed in constant time. Subject to these crucial assumptions, this algorithm solves matroid matching problems in polynomial time.

Let us sketch why these assumptions are indeed necessary. Note that Lovász’ formula, and all known linear matroid matching algorithms rely on properties of linear matroids, which are not satisfied by all matroids. They rely on the fact that the matroid is surrounded by the full linear matroid, and properties of the full linear matroid are what matter most. Lovász’ approach is based on the way how so-called double-circuits behave in the full linear matroid; this behavior generalizes to superclasses, but not to arbitrary matroids. This behavior is the so-called double-circuit property, or DCP for short. Now, while full linear matroids obey the DCP property, linear matroids in general don’t. To solve matroid matching in linear matroids, in principle, we have to extend to the full linear matroid, and consider vectors, subspaces not in our matroid, but in the full linear matroid. To perform linear algebraic computations on the input vectors, we have to use the linear representation, which is not easy to reconstruct from the independence oracle. These linear algebraic computations, for example determining a basis of the intersection of two subspaces, are possible, given the above assumptions.

Let $E = \{l_1, \dots, l_n\}$ be a family of **lines**, i.e. 2-dimensional linear subspaces of vectorspace V . For our notation in vectorspaces, see below. A set $M \subseteq E$ is called **matching** if it is a direct sum, or equivalently, if $r(M) = 2|M|$. Let $\nu(E)$ denote the maximum cardinality of a matching. The pair K, π is called a **cover**, if K is a linear subspace and $\pi = \{A_0, \dots, A_k\}$ is a partition of E . Lovász proved that the maximum cardinality of a matching is equal to the minimum value of a cover, which

is defined by

$$val(K, \pi) := r(K) + \sum_{A_i \in \pi} \left\lfloor \frac{r_{V/K}(A_i)}{2} \right\rfloor.$$

Theorem 2.6 (Lovász, [51]) $\nu(E) = \min val(K, \pi)$, where the minimum is taken over all covers K, π .

Lovász' Theorem is a remarkable achievement that sparked the ongoing research in this area. To obtain a good characterization, we have to add that the minimum attains at a special cover K, π where K may be explicitly represented by a basis. Note that, although this representability of K is not directly stated in Lovász' Theorem, it is easy to show that a bases of K may be determined by performing a polynomial number of “pivot operations” on the input lines. This is the point where we heavily rely on the explicit linear representation of the lines, and that arithmetic operations over the field take constant time. Note that in the case of a representation over the rationals, we have to make sure that the size of rational numbers does not explode while performing the pivot operations! This may also be proved for the well-known linear matroid matching algorithms, see the short section on a basis of K . We conclude that matching is polynomial time solvable for lines represented by given rational numbers.

Notation

Let $r(\cdot) = r_V(\cdot)$ denote the dimension of a given linear subspace \cdot of a given vectorspace V . $sp(\cdot) = sp_V(\cdot)$ denotes the linear subspace spanned by a set \cdot of vectors. To simplify the notation, if \cdot is a set of sets of vectors or subspaces, then we will use the shorthand notation $sp(\cdot) := sp(\bigcup \cdot)$, $r(\cdot) := r(\bigcup \cdot)$. The meet and join operations in the lattice of subspaces is defined by $K \vee L := sp(K \cup L)$, $K \wedge L := K \cap L$. For a subspace K , V/K denotes the quotient space. For a subspace Z we denote by $Z/K := (Z \vee K)/K$ its image in the quotient space. Sometimes we will use the shorthand notation Z – instead of Z/K . Hence if \mathcal{Z} is a family of subspaces of V , then we will also use \mathcal{Z} to denote their images in V/K – instead of the cumbersome notation $\{Z/K : Z \in \mathcal{Z}\}$. For a subspace L of V/K , let $L \times K$ denote its pre-image in V , which is the unique subspace such that $(L \times K)/K = L$ and $K \subseteq L \times K$. $r_{V/K}$ denotes the rank-function over the quotient space, which is determined by $r_{V/K}(L) = r(L \times K) - r(K)$. $sp_{V/K}$ denotes the span function in the quotient space. If Z is a subspace of V , then let $r_{V/K}(A) := r_{V/K}(sp_{V/K}(A))$. A family $\mathcal{Z} = \{Z_1, \dots, Z_k\}$ of subspaces is called a **direct sum** if $r(\bigvee \mathcal{Z}) = \sum_{i=1}^k r(Z_i)$ holds. In this case, \mathcal{Z} is called a direct sum decomposition of $\bigvee \mathcal{Z}$.

Proof of necessity

Consider an arbitrary matching M and an arbitrary cover $K, \pi = \{A_0, A_1, \dots, A_k\}$. Let $M_i := M \cap A_i$ and let $\gamma_i := r(K \cap sp(M_i))$. Then $2|M_i| \leq \gamma_i + r_{V/K}(A_i)$, moreover, since M is a matching, $\sum \gamma_i \leq r(K)$. These inequalities imply

$$|M| \leq \sum \left\lfloor \frac{\gamma_i + r_{V/K}(A_i)}{2} \right\rfloor \leq \sum \left\lfloor \frac{2\gamma_i + r_{V/K}(A_i)}{2} \right\rfloor = val(K, \pi).$$

A cover K, π is called an **optimum cover** if there is a matching of cardinality $val(K, \pi)$. Let us call $A_i \in \pi$ an **even/odd component** if $r_{V/K}(A_i)$ is even/odd. For an optimum cover and a maximum matching, equality must hold throughout, implying the following “slackness conditions”. If A_i is an even component, then $\gamma_i = 0$ and $2|M_i| = r_{V/K}(A_i)$. If A_i is an odd component, then $\gamma_i = 0$ or 1 , where $\gamma_i = 1$ holds for a number of exactly $r(K)$ indices i .

Definition of very strong covers

In this section we define the notion of very strong covers, introduced by Orlin and Vande Vate [63] as a generalization of Edmonds’ alternating forest of blossoms in the graph matching algorithm. While performing the algorithm, we maintain a very strong cover of a set of lines, and by adding one of the remaining lines, we try to find a larger matching, or extend the very strong cover. We remark that the definition of very strong covers is slightly more technical than that of alternating forests and blossoms, but a thorough examination of special cases shows that all features of the definition are indeed necessary to cope with linear matroid matching.

Consider a set $A \subseteq E$ of lines. A matching $N \subseteq A$ is called a **K -hyper-matching in A** if $2|N| = r(N) = r_{V/K}(N) + 1 = r_{V/K}(A) + 1$. Notice that, if there is a K -hyper-matching in a component A_i , then it must be an odd component.

A useful property of a component is when there is a rich collection of K -hyper-matchings. A set A of lines is called **K -hyper-matchable** if $K \wedge sp(A) = \bigvee \{K \wedge sp(N) : N \text{ a } K\text{-hyper-matching in } A\}$. Notice that the K -hyper-matchability of a set may be certified by a list of at most $r_{V/K}(A)$ K -hyper-matchings in A . Algorithmically, a K -hyper-matchable set we will be maintained together with such a short list, which will be useful in the construction of other matchings, as well.

We define the **kernel** of a set of lines as the intersection of the span of maximum matchings. Thus, if K, π is an optimum cover, then the kernel is given by $ker := ker(K, \pi) := \bigwedge \{sp(M) : |M| = val(K, \pi)\}$. It follows from slackness conditions that, if K, π is an optimum cover, then K and all the even components are subsets of $ker(K, \pi)$.

An optimum cover is called **very strong cover** if it fulfills the following properties. A_0 plays a special role in this definition, which is displayed by labelling it as the first member of the partition.

(38) A_0 is a matching in V/K .

(39) A_i is K -hyper-matchable for $i \geq 1$.

(40) K, π is **kernel-inducing**, that is $K \vee sp(A_0) = ker(K, \pi)$.

Algorithmically, a very strong cover will be maintained together with short lists to certify properties (39),(40). That is, for $i \geq 1$ we maintain a short list $\mathcal{L}_{(39)}^i$ of K -hyper-matchings in A_i such that $K \wedge sp(A_i) = \bigvee \{K \wedge sp(N) : N \in \mathcal{L}_{(39)}^i\}$. This implies (39). Moreover, we maintain a nonempty short list $\mathcal{L}_{(40)}$ of maximum matchings such that $K \vee sp(A_0) = \bigwedge \{sp(M) : M \in \mathcal{L}_{(40)}\}$. This implies that (40) holds, since $K \vee sp(A_0) \subseteq ker(K, \pi)$ follows from the fact that K, π is an optimum cover, and that A_0 is a matching in V/K . It is easy to keep these lists short by dropping redundant members.

Properties of very strong covers

In this section we sum up corollaries of the definition of very strong covers. A matching $N \subseteq A$ is called a **K -hypo-matching in A** if $2|N| = r(N) = r_{V/K}(N) = r_{V/K}(A) - 1$. Thus, K -hypo-matchings in A are the “near-perfect” matchings of A in V/K . A set $A \subseteq E$ is called **K -hypo-matchable** if $K = \bigwedge \{K \vee sp(N) : N \text{ a } K\text{-hypo-matching in } A\}$.

Consider a very strong cover K, π , and a maximum matching M . It follows from the slackness conditions that, $A_0 \subseteq M$; $A_i \cap M$ is a K -hyper-matching for exactly $r(K)$ indices $i \geq 1$; and $A_i \cap M$ is a K -hypo-matching for the remaining indices $i \geq 1$. We also claim that

(41) A_i is K -hypo-matchable for all $i \geq 1$.

To prove this, consider an arbitrary vector $v \in sp(A_i) - K$ for some $i \geq 1$. Then, by kernel-inducing, there is a maximum matching M such that $v \notin sp(M)$. Since $K \subseteq sp(M)$, $M \cap A_i$ is a K -hypo-matching such that $v \notin K \vee sp(M \cap A_i)$. This implies (41). Moreover, it is in fact easy to derive from $\mathcal{L}_{(40)}$ a short list of K -hypo-matchings in A_i to certify that A_i is K -hypo-matchable.

Definition of strong covers

A cover K, π is called a **strong cover** if it fulfills conditions (39) and (41), that is, for $i \geq 1$, A_i is both K -hyper- and K -hypo-matchable. We remark that a very

strong cover is a strong cover, as well. However, a strong cover may not even be optimum. We remark that A_0 is arbitrary, and thus A_0 plays a special role which is displayed by labelling it as the first member of the partition. Algorithmically, we assume that a strong cover is given together with short lists certifying K -hypo- and -hyper-matchabilities. Strong covers will be useful in the algorithm in the following setting: Given a very strong cover of a subset F of lines, adding a line $l \in E - F$ to A_0 produces a strong cover of $F + l$. To figure out whether there is a larger matching in $F + l$, we apply the following matroid intersection representation.

Representing strong covers by matroid intersection

We describe a crucial element of our algorithm, which relates matchings to matroid intersection, if we have a strong cover K, π at hand. The existence of a matching of cardinality $|A_0| + \text{val}(K, \pi - A_0)$ is equivalent with the following instance of matroid intersection.

Let $K, \pi = \{A_0, A_1, \dots, A_k\}$ be a strong cover. The common groundset of the two matroids is constructed as a multiset of vectors of V . We define a multiset $D_0 := \cup\{B_e : e \in A_0\}$, where B_e is a basis of e for $e \in A_0$. For $i \geq 1$, let $D_i \cup K_i \subseteq V$ denote a basis of $\text{sp}(A_i)$ such that $\text{sp}(K_i) = \text{sp}(A_i) \wedge K$. Choose K_i such that for any $k \in K_i$ there is a K -hyper-matching $N \in \mathcal{L}_{(39)}^i$ such that $\text{sp}(k) = K \wedge \text{sp}(N)$. We define multisets $S_D := \bigcup_{i \geq 0} D_i$, $S_K := \bigcup_{i \geq 1} K_i$, and $S_i := D_i \cup K_i$ for $i \geq 1$. The common groundset of the two matroids is the multiset $S := S_D \cup S_K$. The first matroid $\mathcal{M}_1 := (S, \mathcal{I}_1)$ is a partition matroid the independent sets of which are those sets containing at most one element from S_i , for every $i \geq 1$, and no element from D_0 . The second matroid is defined by a direct sum $\mathcal{M}_2 := (S, \mathcal{I}_2) := \mathcal{M}_V|_{S_K} \oplus (\mathcal{M}_{V/K}|_{S_D})^*$. Thus, a set $X \subseteq S$ is independent in \mathcal{M}_2 if and only if $X \cap S_K$ is independent in V , and $(S_D - X) \cup K$ spans $S_D \cup K$.

Let Y denote a basis of \mathcal{M}_1 maximizing $r_2(Y)$, and let $\delta := r_2(S) - r_2(Y)$. Our main concern is whether δ is zero, or positive.

The following claim implies that, if $\delta = 0$, then K, π is an optimum cover. In the proof we perform the most important ‘‘primal’’ construction, which is a way how to construct a big maximum matching from a strong cover. The additional assertion will be useful to control the span of maximum matchings.

Claim 2.7 *If $\delta = 0$, then one can construct a matching M of cardinality $|A_0| + \text{val}(K, \pi - A_0)$. If, moreover, we are given a vector $v \notin \text{sp}(S_K \cap Y) \times \text{sp}_{V/K}(S_D - Y)$, then M can be constructed such that $v \notin \text{sp}(M)$ holds, in addition.*

Proof. If $\delta = 0$, then Y is a basis of \mathcal{M}_1 and a spanning set of \mathcal{M}_2 . Define $T_i := \text{sp}((K_i \cap Y) \cup (D_i - Y))$ for $i \geq 1$. Then $\{\text{sp}(A_0), T_1, \dots, T_k\}$ is a direct sum

not spanning v , and such that $r(T_i) = r_{V/K}(A_i) \pm 1$, with “+1” for exactly $r(K)$ indices. A “pivot operation” is that, we replace subspace T_i by $sp(N_i)$, where N_i is a K -hyper-matching for indices i with “+1”, and a K -hypo-matching for indices i with “-1”. The definition of K -hypo/hyper-matchability implies that each of these pivots can be performed in such a way that we retain a direct sum $\{sp(A_0), T_1, \dots, T_k\}$ not spanning v . Thus, in the end, $M := A_0 \cup \bigcup_{i \geq 1} N_i$ will be a matching of the desired cardinality. We remark that all the N_i 's may be chosen from the certifying lists, and thus, the above construction may be performed in polynomial time. \square

Next we perform the most important “dual” construction of our algorithm, which is a way how to construct a small-value strong cover. By Theorem 2.5, there is a unique inclusionwise minimal set U such that $r_1(S-U) + r_2(U) = r_2(Y) = r_2(S) - \delta$, and U can be constructed in polynomial time. Our choice implies that U is equal to the union of a couple of sets $S_i, i \geq 1$. Say $U = \bigcup_{i \in J} S_i$ for some $J \subseteq \{1, \dots, k\}$, and let $I := \{1, \dots, k\} - J$. Then, let $K_J := \bigvee \{K_i : i \in J\}$, let $A_0^J := A_0 \cup \bigcup \{A_i : i \in I\}$, and let $\pi_J := \{A_0^J\} \cup \{A_i : i \in J\}$. The following claim asserts that this construction provides a small-value strong cover K_J, π_J .

Claim 2.8 $val(K_J, \pi_J) \leq val(K, \pi - A_0) + |A_0| - \lfloor \frac{1}{2}(\delta + 1) \rfloor$.

Proof. Clearly, $r_1(S - U) = |I|$, and $r_2(S) = r(K) + |S_D| - r_{V/K}(S_D)$, and $r_2(U) = r(K_J) + |S_D \cap U| - r_{V/K}(S_D) + r_{V/K}(S_D - U)$, and $|S_D - U| - |I| = 2|A_0| + \sum_{i \in I} (r_{V/K}(A_i) - 1)$, and $r_{V/K}(S_D - U) = r_{V/K}(A_0^J)$. Substituting in $r_1(S - U) + r_2(U) = r_2(S) - \delta$ we get that $r_{V/K}(A_0^J) \leq r(K) - r(K_J) + \sum_{i \in I} (r_{V/K}(A_i) - 1) + 2|A_0| - \delta$. Notice that $K \subseteq sp(K_J \cup A_0^J)$, thus $r_{V/K}(A_0^J) + r(K) = r_{V/K_J}(A_0^J) + r(K_J)$. This gives

$$\begin{aligned} \lfloor \frac{1}{2} r_{V/K_J}(A_0^J) \rfloor &= \lfloor \frac{1}{2} (r_{V/K}(A_0^J) + r(K) - r(K_J)) \rfloor \leq \\ &\leq \lfloor \frac{1}{2} (2r(K) - 2r(K_J) + \sum_{i \in I} (r_{V/K}(A_i) - 1) + 2|A_0| - \delta) \rfloor \\ &= r(K) - r(K_J) + \sum_{i \in I} \lfloor \frac{1}{2} r_{V/K}(A_i) \rfloor + |A_0| - \lfloor \frac{1}{2}(\delta + 1) \rfloor, \end{aligned}$$

Notice that for every $i \in J$ we have $K_i \subseteq K_J \subseteq K$, which implies $r_{V/K_J}(A_i) = r_{V/K}(A_i)$. Thus,

$$\begin{aligned} val(K_J, \pi_J) &= val(K, \pi - A_0) - r(K) - \sum_{i \in I} \lfloor \frac{1}{2} r_{V/K}(A_i) \rfloor + \\ &\quad + r(K_J) + \lfloor \frac{1}{2} r_{V/K_J}(A_0^J) \rfloor \leq \\ &\leq val(K, \pi - A_0) + |A_0| - \lfloor \frac{1}{2}(\delta + 1) \rfloor. \end{aligned}$$

\square

Corollary 2.9 *If $\delta \geq 1$, then $val(K_J, \pi_J) \leq val(K, \pi - A_0) + |A_0| - 1$. Moreover, if $\delta = 1$, then A_0^J is an odd component.* \square

Corollary 2.10 *If $\delta = 0$, then K_J, π_J is an optimum, kernel-inducing cover, and A_0^J is its even component.* \square

A linear matroid matching algorithm

We maintain a very strong cover K, π of a subset $F \subseteq E$ of lines. As already mentioned before, we also maintain the short lists $\mathcal{L}_{(39)}^i, \mathcal{L}_{(40)}$ to certify that K, π is indeed a very strong cover of F . Thus, any matching $M \in \mathcal{L}_{(40)}$ is a maximum matching in F . We try to find a larger matching by considering lines from $E - F$. The call for termination is the following **sufficient condition of optimality**.

$$(42) \quad \text{Every line in } E - F \text{ is subset of } \ker := \ker(K, \pi) = K \vee \text{sp}(A_0).$$

If this condition holds, then $K, \bar{\pi} := \{A_0 \cup (E - F), \dots, A_k\}$ is an optimum cover of E , and we are done. In a general step of the algorithm we check whether the sufficient condition of optimality (42) holds, and in that case, the algorithm terminates. Otherwise we choose a line $l \in E - F$ such that $r_{V/\ker}(l) \geq 1$. Then we construct either a matching in $F + l$ of cardinality $\text{val}(K, \pi) + 1$, or a very strong cover K', π' for $F + l$. This construction goes as follows.

Notice that $K, \pi^{+l} := \pi - \{A_0\} + \{A_0 + l\}$ is a strong cover of $F + l$, and $A_0 + l$ is the designated first member of π^{+l} . Consider the matroid intersection representation of K, π^{+l} , and let Y be a basis of \mathcal{M}_1 that maximizes $r_2(Y)$.

Suppose $\delta = 0$. Then, by Claim 2.7, we can construct a matching in $F + l$ of cardinality $\text{val}(K, \pi^{+l} - \{A_0 + l\}) + |A_0 + l| = \text{val}(K, \pi) + 1$, and we are done.

Suppose $\delta \geq 1$. Let K_J, π_J denote the cover of $F + l$ obtained from Claim 2.9, that is $K_J = \bigvee \{K_i : i \in J\}$, and $\pi_J := \{Q\} \cup \{A_i : i \in J\}$, where $Q = \{A_0 + l\} \cup \bigcup \{A_i : i \in I\}$. We get that $\text{val}(K_J, \pi_J) = \text{val}(K, \pi^{+l} - \{A_0 + l\}) + |A_0 + l| - 1 = \text{val}(K, \pi)$, implying the following claim.

Claim 2.11 *K_J, π_J is an optimum cover of $F + l$, and Q is an odd component.*

Unluckily, K_J, π_J might not be a very strong cover. To obtain a very strong cover, we dismantle the odd component Q via the following simple construction. Clearly, the trivial cover $\mathbf{0}, \{Q\}$ is an optimum cover of Q with respect to vectorspace $Z := V/K'$. Only using that the trivial cover is optimum, the following construction produces another optimum cover. For $q = 0, 1, 2$, let $Q_q := \{e \in Q : r_Z(Q) - r_Z(Q - e) = 2 - q\}$. For $e \in Q_1$, choose $k_e \in \text{sp}_Z(e)$ such that $\text{sp}_Z(k_e) = \text{sp}_Z(e) \wedge \text{sp}_Z(Q - e)$. Let $K_Q := \text{sp}_Z(\{k_e : e \in Q_1\})$, and $\pi_Q := \{Q_0\} \cup \{Q_2\} \cup \{\{e\} : e \in Q_1\}$. It is straightforward that $r_{Z/K_Q}(Q_2) = r_Z(Q) - |Q_1| - 2|Q_0|$, and $r_{Z/K_Q}(e) = 1$ for all

$e \in Q_1$, and $r_{Z/K_Q}(Q_0) = 2|Q_0|$. Thus, K_Q, π_Q is an optimum cover of Q in Z , and Q_0 is a matching in Z/K_Q .

Now, let $K' := K_J \times K_Q$ and let $\pi' := \pi_J - \{Q\} + \pi_Q$, such that Q_0 is the designated first member of π' . Since K_J, π_J is an optimum cover of $F + l$, and K_Q, π_Q is an optimum cover of $Q \in \pi_J$ in V/K_J , we get that K', π' is an optimum cover of $F + l$, that is, $val(K', \pi') = \nu(F + l) = \nu(F) = val(K, \pi)$. Moreover, since Q_0 is a matching in Z/K_Q , Q_0 is a matching in V/K' , as well. The following claim sums up what we have proved above, which is in fact two of the properties of a very strong cover.

Claim 2.12 K', π' is an optimum cover, and Q_0 is a matching in V/K' . \square

Claim 2.13 K', π' is kernel-inducing, and the certifying list $\mathcal{L}_{(40)}$ can be updated in polynomial time.

Proof. Note that

$$(43) \quad K' \vee sp(Q_0) = (K \vee sp(A_0)) \wedge (K' \vee sp(A_0 + l)) \wedge (K \vee K_Q \vee sp(Q_0)),$$

thus we only have to construct maximum matchings such that the intersection of their spans is no more than $(K' \vee sp(A_0 + l)) \wedge (K \vee K_Q \vee sp(Q_0))$.

First we construct maximum matchings such that the intersection of their spans is no more than $K' \vee sp(A_0 + l)$. Firstly, if $A_0 + l$ is not a matching, then $K \subseteq K'$, and in this case we are done by considering maximum matchings in the old list $\mathcal{L}_{(40)}$. Henceforth we assume that $A_0 + l$ is a matching. Consider those maximum matchings M satisfying $A_0 + l \subseteq M \subseteq F + l$. Consider an arbitrary vector $v \notin K' \vee sp(A_0 + l)$. Consider the matroid intersection representation of K, π^{+l} . By Theorem 2.5, there is a maximum common independent set X such that $v \notin sp(X \cap S_K)$. Since $A_0 + l$ is a matching, we may add some elements from S_D to X such that we obtain a basis Y of \mathcal{M}_1 spanning \mathcal{M}_2 , and $v \notin sp(Y \cap S_K)$. By Claim 2.7, there is a maximum matching M_v such that $v \notin sp(M_v)$.

Second we construct maximum matchings such that the intersection of their spans is no more than $K \vee K_Q \vee sp(Q_0)$. For this we consider maximum matchings $M \subseteq F + l$ such that $|(A_0 + l) - M| = 1$. Consider an arbitrary vector $v \notin K \vee K_Q \vee sp(Q_0)$. We may assume that $v \in K \vee sp(A_0)$. Thus, suppose that $v \in K \vee sp(A_0)$, say $v = k + \sum_{e \in A_0} v_e$, where $v_e \in e$ for $e \in A_0$ and $k \in K$. Since $v \notin K \vee K_Q \vee sp(Q_0)$, we get that there is a line $e \in A_0$ such that either $e \in Q_2$ and $v_e \neq \mathbf{0}$, or $e \in Q_1$ and $v_e \notin sp(k_e)$. It suffices to prove that there is a maximum matching $M \subseteq F + l - e$ such that $v \notin sp(M)$. Notice that $K, \pi^{+l-e} := \pi - \{A_0\} + \{A_0 + l - e\}$ defines a strong cover of $F + l - e$, and consider its matroid intersection representation. Let U_e be the inclusionwise minimal set minimizing $r_1(S - U_e) + r_2(U_e)$, and the set J_e of

indices is defined such that $U_e = \bigcup_{i \in J_e} S_i$. Let K_e, π_e denote the cover constructed from J_e , and let Q_e denote the new component in π_e .

Suppose that $\delta \geq 1$. By Corollary 2.9 we get that $\text{val}(K_e, \pi_e) \leq \text{val}(K_J, \pi_J) - 1$, and Q_e is an odd component of K_e, π_e . This implies that $\text{val}(K_e, \pi_e - \{Q_e\} + \{Q_e + e\}) = \text{val}(K_J, \pi_J)$. Since U and U_e were chosen inclusionwise minimal, we get that $J = J_e$, $Q = Q_e + e$, and $K_e = K_J$. Since $\text{val}(K_J, \pi_J) > \text{val}(K_e, \pi_e)$, and both Q and Q_e are odd components, we get that $e \in Q_0$. A contradiction.

Suppose that $\delta = 0$. Then Corollary 2.10 applies, and we get that $\ker(K_e, \pi_e) = K_e \vee \text{sp}(Q_e)$. Suppose for contradiction that $v \in K_e \vee \text{sp}(Q_e)$. This implies that $r_{V/K_e}(Q_e + e) \leq r_{V/K_e}(Q_e) + 1$. Since Q_e is an even component, we get that $\text{val}(K_e, \pi_e) = \text{val}(K_e, \pi_e - \{Q_e\}) + \{Q_e + e\}$. Since U and U_e were chosen inclusionwise minimal, we get that $J = J_e$, $Q = Q_e + e$, $K_e = K_J$, and thus, $e \in Q_1$. Since $v \in K_e \vee \text{sp}(Q_e)$, we get that $v_e \in e \wedge \text{sp}(Q_e) = \text{sp}(k_e)$. A contradiction.

This concludes the proof of kernel-inducing. A polynomial time algorithm to update the certifying list follows from the fact that the construction in Claim 2.7 can be performed in polynomial time. \square

It remains to prove that all components of K', π' , except for Q_0 , are K' -hyper-matchable. Since $K_i \subseteq K'$ for $i \in J$, the K' -hyper-matchability of components A_i follows from their K -hyper-matchability. Next, the K' -hyper-matchability of components $\{e\} : e \in Q_1$ is trivial. Thus the following claim is all left to prove.

Claim 2.14 Q_2 is a hyper-matchable component of K', π' , and the certifying list can be constructed in polynomial time.

Proof. We may assume w.l.o.g. that $Q_0 = \emptyset$, and $J = \emptyset$, thus $K' = K_Q$. Then, given a proper subspace $K^* < K'$ with $r(K^*) = r(K') - 1$, we must find a perfect matching in Q_2 with respect to V/K^* , that is of cardinality $r_{V/K^*}(Q_2)/2$. (The iterative application of this sub-routine provides the certifying list for K' -hyper-matchability of Q_2 .) Now, consider the set Q_2 of lines in vector space V/K^* , and its strong cover defined by $\pi^* := \{A_i : i \in I\} \cup \{Q_2 \cap (A_0 + l)\}$ and $\{\mathbf{0}\}$. Consider the matroid intersection representation of $\pi^*, \{\mathbf{0}\}$ with respect to V/K^* . Note that $\delta \geq 1$ is not possible, since this would contradict our choice of U in Edmonds' intersection formula. Thus $\delta = 0$. By Corollary 2.10, we construct a perfect matching in Q_2 with respect to V/K^* , and we are done. \square

Determining a basis of K

Next we show a formula to determine the subspace K in a very strong cover, which will be useful to prove that our matroid matching algorithm terminates in poly-

mial time, if the matroid is represented over the rationals. Recall that we proved the polynomial time complexity on the assumption that arithmetic operations may be performed in constant time in the specific field. Unluckily, this does not hold for rationals, and one could imagine a sequence of linear algebraic operations performed on some rational vectors, which results in an exponential growth in the size of rationals. To resolve this situation, we show that the subspace K in a very strong cover arises in a special way, and thus it can be computed from coefficients/coordinates of the given lines directly, without going all the way through the algorithm. This direct representation of K follows from the below claim, saying that K arises from the union of 2-intersections.

Suppose $K, \pi = \{A_0, \dots, A_k\}$ is a very strong cover of F . Kernel-inducing implies that for all indices $1 \leq i \leq k$ there is a maximum matching M_i such that $sp(A_i) \not\subseteq sp(M_i)$. Slackness conditions imply that A_i and $M_i - A_i$ is a direct sum in V/K . Moreover, $M_i \cap A_i$ is a K -hypo-matching, hence $K \subseteq sp(M_i - A_i)$, which implies that $sp(A_i) \cap sp(M_i - A_i) = K_i$. Since $K = \bigvee_{i=1}^k K_i$, we have proved the following claim.

Claim 2.15 $K = \bigvee_{i=1}^k (sp(A_i) \wedge sp(M_i - A_i))$. □

Remarks

- A significant open question concerns a couple of combinatorially meaningful matroids, which are known to be representable over complicated fields, not satisfying the above assumptions. Some matroids related to graph orientations and connectivity are known to be representable over an algebraic field extension of the rationals, but an “explicit” representation is not known. (The “implicit” representation is given by a matrix over indeterminates.) For example, such a representation is known for the following unsolved problem: Decide whether a given undirected graph has a strongly connected orientation in which every node has even in-degree. For such problems, a polynomial time randomized algorithm follows by substituting indeterminates of the field extension by random integers, a method first used by Lovász [54]. This randomized approach does not provide a good characterization, and the deterministic complexity of these problems is unsolved.
- The difference between this algorithm, and those graph matching algorithms in the first chapter, and this linear matroid matching algorithm is the following. Algorithms in the first chapter maintain a pair of a graph and a matching. The above algorithm maintains a stronger structure, namely a set of lines, and its very strong cover with those certifying lists.

2.3 The Partition Formula for polymatroid matching

In this section we propose a constructive proof for the Partition Formula on matching in the class of *ntcdc*-free polymatroids. This provides a polynomial time algorithm for special cases, including maximum genus graph embedding, and graph orientation with parity and rooted-connectivity requirements. We remark that the class of *ntcdc*-free polymatroids does not contain most linear polymatroids, hence the result in this section is not related to the mainstream of matroid matching theory. The motivation is to construct a polynomial time algorithm for some of those special cases, for which well-known results earlier did not provide a polynomial time algorithm.

Lovász [51] proved the linear matroid matching formula by using a technical lemma, which claims the Partition Formula assuming a technical condition on the matroid. A direct application of this result seems rather difficult, since there are few special cases of that technical condition. Recently, Makai and Szabó [59] proved an analogous sufficient condition, that is, the Partition Formula holds for polymatroids without non-trivial compatible double circuits – *ntcdc*'s, for short – see definition later. In this section we prove this result of Makai and Szabó in a constructive way, that is, we provide a polynomial time algorithm which either returns a matching and a partition certifying the Partition Formula, or returns a *ntcdc*. The state of the art is that, general results from (poly)matroid matching theory do not imply a polynomial time algorithm for all polymatroids without *ntcdc*'s. The non-constructive proof of Makai and Szabó relied on Lovász' sufficient condition, and showing that some operations preserve *ntcdc*-freeness of polymatroids. Unluckily, existing linear matroid matching algorithms cannot be applied directly to this setting, as it is unknown and unlikely that this setting reduces to linear matroid matching. Here we apply a well-known reduction principle, a specific kind of projections, to construct the analogue of Edmonds' matching algorithm [21], or Orlin and Vande Vate's linear matroid matching algorithm [63]. Despite all the similarities, it took a great effort to figure out the precise definition of the analogue of contracting blossoms. The conclusion is that polymatroid matching can be solved in semi-strongly polynomial time by a compact algorithmic framework, similar to those considered in the first chapter of the thesis.

We clarify some useful notation concerning polymatroids later, here is a brief outline of what is going on in this section. Consider a polymatroid function b on a finite groundset S , and let $\mathcal{P}(b)$ denote its induced polymatroid. A vector is called **even** if all of its entries is an even number. **Matchings** are the even vectors in $\mathcal{P}(b)$. The **size** of a vector $x \in \mathbb{N}^S$ is the sum of all of its entries, that is $x(S)$. The polymatroid matching problem is that, given a polymatroid function by an evaluation oracle, find a maximum size matching. Let $\nu(b)$ denote half the

maximum size of a matching, that is

$$\nu(b) := \max \left\{ \frac{x(S)}{2} : x \text{ a matching with respect to } b \right\}.$$

Polymatroid matching, being a generalization of matroid matching, is a striking notion in combinatorial optimization. Jensen and Korte [42] proved that, unluckily, matroid matching has exponential time complexity in the independence oracle setting, and hence, polymatroid matching is at least as hard. Thus, in a quest to extend our knowledge of tractable polymatroid matching problems, we restrict our investigation to classes of polymatroids. In order to solve maximum matching in a specific polymatroid, one has to figure out an upper bound on the size of matchings. The “**partition upper bound**” is the following expression obtained from partitions of the groundset and taking parity into consideration. Let

$$\tau(b) := \min \left\{ \sum_{S_i \in \pi} \left\lfloor \frac{b(S_i)}{2} \right\rfloor : \pi = \{S_1, \dots, S_t\} \text{ a partition of } S \right\}.$$

Clearly, for any matching x and any set $Z \subseteq S$ we get that $\frac{1}{2}x(Z) \leq \lfloor \frac{1}{2}b(Z) \rfloor$, hence $\nu(b) \leq \tau(b)$ holds for any polymatroid function b . We remark that there are small examples for strict inequality: for example, if there is an even ntdc with support S , then $\nu(b) = \tau(b) - 1$.

Hence, we are interested in sufficient conditions on b implying the Partition Formula

$$\nu(b) = \tau(b).$$

We hope for a sufficient condition satisfied by a broad class of special cases. The class of linear (poly)matroids is, however not such a class, since even ntdc’s are spread all over the place in most linear (poly)matroids. Notice that Lovász’ linear matroid matching formula is more complicated than the Partition Formula, since there we take a partition and a projection into consideration. Here we focus on classes of matroid matching problems where partitions are sufficient to describe the maximum size of a matching. Lovász discovered the following sufficient condition for the Partition Formula for matroid matching.

Theorem 2.16 (Lovász, 1980, [51]) *The Partition Formula holds, if there is no even ntdc, and for every element $i \in S$ there is a maximum matching not spanning i .*

The sufficient condition in this theorem is only designed for the setting occurring in those proofs in Lovász’ paper, and is not meant for direct application. Recently, Makai and Szabó observed a powerful analogue of Lovász’ sufficient condition, and this applies to a bunch of combinatorial optimization problems directly. In this

section of the thesis, we provide an alternative proof of this result, which implies a polynomial time algorithm, as well.

Theorem 2.17 (Makai, Szabó, 2006, [59]) *The Partition Formula holds for all $ntcdc$ -free polymatroids.*

We remark that the algorithm constructed below can be implemented in semi-strongly polynomial running time. By definition, an algorithm has strongly polynomial running time, if it performs a polynomial number of operations, counting arithmetic operations on rationals as one step, and moreover, the size of the rationals does not go beyond a polynomial factor of the size of rationals in the input. The question is, whether taking the lower integer part of a rational number is considered an arithmetic operation. Schrijver [77] calls such algorithms **semi-strongly polynomial time**, while he reserves the expression **strongly polynomial time** for those algorithms only performing elementary arithmetic operations. For capacitated matching problems, the odds are against constructing a strongly polynomial time algorithm, since we have to take the parity of capacities into consideration. Thus, the most we can hope for, is a semi-strongly polynomial time algorithm for the Partition Formula.

Special cases

According to Makai, Szabó [59], the following problems are equivalent with matching in polymatroids without $ntcdc$'s.

- Pinning-down a minimum number of nodes of a framework in the Euclidean plane to make it generically rigid. A good characterization, and in fact a polynomial time algorithm follows from a reduction to non-bipartite matching, Fekete [29].
- Finding a maximum genus embedding of a graph. A good characterization is given by Nebeský [62]. A polynomial time algorithm is given by Furst, Gross, McGoech [30] via matching in a co-graphical matroid.
- Given a number $k \geq 1$, and an undirected graph with a prescribed root, find a rooted- k -arc-connected orientation maximizing the number of nodes of even in-degree. For $k = 1$, this is equivalent with maximum genus embedding. A good characterization is given by Frank, Jordán, Szigeti [26]. The state-of-the-art is that a polynomial time algorithm does not follow from well-known general results of matroid matching theory. A polynomial time algorithm follows from results in this section.

- Given an undirected (hyper-)graph and a non-negative intersecting submodular set-function on its node-set, find an orientation covering the set-function and maximizing the number of nodes of even in-degree. (The previous problem is a special case.) A good characterization is given by T. Király, Szabó [44]. A semi-strongly polynomial time algorithm follows from results in this section.
- Makai, Szabó [59] observed that the following operations preserve ntedc-freeness of polymatroid functions. Direct sum, addition of a non-negative modular function, imposing an upper bound on singletons. However, the following operations may introduce ntedc in polymatroids. Dual, truncation, Dilworth truncation.

Polymatroids

Consider a set-function $b : 2^S \rightarrow \mathbb{Z}$. b is called **non-negative** if it returns only non-negative values. b is called **submodular** if it fulfills the submodular inequality

$$(44) \quad b(X) + b(Y) \leq b(X \cup Y) + b(X \cap Y)$$

for any pair $X, Y \subseteq S$ of sets. b is called **supermodular** if $-b$ is submodular. b is called **non-decreasing** if for any $X \subseteq X' \subseteq S$ we get $b(X) \leq b(X')$. A set-function $b : 2^S \rightarrow \mathbb{Z}$ is called a **polymatroid function** if it is non-negative, submodular, and $b(\emptyset) = 0$. This implies that b is non-decreasing. For a polymatroid function b , the **polymatroid** induced by b is defined by

$$(45) \quad \mathcal{P}(b) := \{x \in \mathbb{R}^S : x \geq 0, x(Z) \leq b(Z) \text{ for all } Z \subseteq S\}.$$

It is well-known and easy to see that $b(Z) = \max\{x \cdot \chi_Z : x \in \mathcal{P}(b)\}$, thus a polymatroid (assumed as a subset of the Euclidean space) uniquely determines the polymatroid function b it is induced by. Vectors in $\mathcal{P}(b)$ of maximum size $b(S)$ are called **bases**.

Consider a polymatroid function $b : 2^S \rightarrow \mathbb{N}$, and a vector $x \in \mathbb{N}^S$. For a set $Z \subseteq S$, we call $def_{b,x}(Z) := x(Z) - b(Z)$ the **deficiency of set Z** with respect to b, x . This is equal to the violation of the inequality corresponding to Z in the above description of polymatroid $\mathcal{P}(b)$. A set is called **m -deficient** with respect to b, x if $def_{b,x}(Z) = m$. The **deficiency of a vector x** is defined by $def_b(x) := \max_{Z \subseteq S} def_{b,x}(Z)$. Notice that the deficiency of a vector is non-negative, since $def_{b,x}(\emptyset) = 0$. Thus, 0-deficient vectors are precisely those vectors in the polymatroid. Notice that the deficiency of a vector thus refers to the so-called

most violated inequality. For a strongly polynomial time algorithm to find a most violated inequality, see Schrijver [77]. Notice that for any vector x , $def_{b,x}(\cdot)$ is a supermodular set-function. The family of sets Z such that $def_{b,x}(Z) = def_b(x)$ is closed under taking unions and intersections. If $def_b(x) > 0$, then the unique inclusionwise minimal set Z attaining $def_{b,x}(Z) = def_b(x)$ is non-empty, and for any $u \in Z$, we get $def_b(x - \chi_u) = def_b(x) - 1$.

Circuits and double circuits

Consider a 1-deficient vector $x \in \mathbb{N}^S$. The unique inclusionwise minimal 1-deficient set is called the **fundamental circuit** of x (with respect to b). It is easy to see that $x - \chi_a \in \mathcal{P}(b)$ if and only if a is in the fundamental circuit of x . A 1-deficient vector x is called a **circuit** if its support is equal to its fundamental circuit.

Consider a 0-deficient vector $x \in \mathbb{N}^S$, and suppose that $u \in C \subseteq S$. We say that “ u induces a circuit on C in x ” if $x + 2\chi_u$ is 1-deficient, and its fundamental circuit is C .

Consider a 2-deficient vector $x \in \mathbb{N}^S$, and let $W := \text{supp}(x)$. x is called a **compatible double circuit** (or **cdc**, for short), if W is the unique inclusionwise minimal 2-deficient set, and there is a partition $\pi = \{W_1, \dots, W_k\}$ of W such that $k \geq 2$ and $\{W - W_i : i = 1, \dots, k\}$ is equal to the family of all inclusionwise minimal 1-deficient sets. We remark that if x is a cdc, then π is uniquely determined – let it be called the **principal partition** of x . If $k = 2$, then x is called a **trivial cdc**. If $k \geq 3$, then x is called a **non-trivial compatible double circuit**, or **ntcdc**, for short.

We remark that there is a strongly polynomial time algorithm to find the fundamental circuit of a 1-deficient vector, or to test whether a 2-deficient vector is a compatible double circuit and returning its principal partition.

Projections

In this section we consider so-called projections of subsets $B \subseteq S$. The **projection of B** is a set-function $b^B : 2^{S-B} \rightarrow \mathbb{N}$ defined for $X \subseteq S - B$ by

$$(46) \quad b^B(X) := \min \{ b(X), b(X \cup B) - b(B) + 1 \}.$$

We remark that $b^\emptyset \equiv b$, and moreover, Claim 2.18 implies that projections provide polymatroid functions. This construction will prove useful in dealing with polymatroid matchings. Also let us remark that projections may be defined through well-known (poly)matroid operations. The projection of B may be defined by first

adding $b(B) - 1$ generic elements to the flat spanned by B , and then contracting these elements. The precise definition of these concepts is unnecessary, we just prove the following claim directly.

Claim 2.18 b^B is a polymatroid function.

Proof. b^B clearly is non-negative and integer; it remains to check whether b^B fulfills the submodular inequality for sets $X, Y \subseteq S - B$.

Firstly, suppose that $b^B(X) = b(X)$ and $b^B(Y) = b(Y)$. By definition, $b^B(X \cup Y) \leq b(X \cup Y)$ and $b^B(X \cap Y) \leq b(X \cap Y)$, implying the submodular inequality.

Secondly, suppose that $b^B(X) = b(X \cup B) - b(B) + 1$ and $b^B(Y) = b(Y \cup B) - b(B) + 1$. By definition, $b^B(X \cup Y) \leq b(X \cup Y \cup B) - b(B) + 1 = b((X \cup B) \cup (Y \cup B)) - b(B) + 1$ and $b^B(X \cap Y) \leq b((X \cap Y) \cup B) - b(B) + 1 = b((X \cup B) \cap (Y \cup B)) - b(B) + 1$, implying the submodular inequality.

Thirdly, suppose that $b^B(X) = b(X)$ and $b^B(Y) = b(Y \cup B) - b(B) + 1$. By definition, $b^B(X \cup Y) \leq b(X \cup Y \cup B) - b(B) + 1 = b(X \cup (Y \cup B)) - b(B) + 1$ and $b^B(X \cap Y) \leq b((X \cap Y) \cup B) - b(B) + 1 = b(X \cap (Y \cup B)) - b(B) + 1$, implying the submodular inequality. \square

Claim 2.19 $\mathcal{P}(b^B) = \{y \in \mathbb{R}^{S-B} : \text{there is } [y, z] \in \mathcal{P}(b) \text{ s.t. } z(B) = b(B) - 1\}$.

Proof. Consider a vector $y \in \mathbb{R}^{S-B}$ and a vector $w \in \mathbb{R}^B$ such that $w(B) = b(B)$ and $[w, 0] \in \mathcal{P}(b)$. We claim that $\text{def}_b([w, y]) \leq 1$. Suppose for contradiction that $\text{def}_{b, [w, y]}(Z) \geq 2$ for some $Z \subseteq S$. Clearly, since $[w, 0] \in \mathcal{P}(b)$, we get that $\text{def}_{b, [w, y]}(Z \cap B) \leq 0$. Then, by supermodularity of deficiency, $\text{def}_{b, [w, y]}(Z \cup B) \geq 2$. This implies that $y \notin \mathcal{P}(b^B)$. \square

In the sequel we will investigate the projection of a family of disjoint sets which, in general, depends on the order in which we perform their projection. We consider the specific setting occurring in our proofs later, and this setting implies that the projection does not depend on the order.

Consider a family $\mathcal{H} = \{H_1, \dots, H_m\}$ of disjoint subsets of S . Assume that there is a vector $x \in \mathcal{P}(b)$ such that for all $i = 1, \dots, m$, we have $x(H_i) = b(H_i) - 1$, and there is an element $h_i \in H_i$ such that $x + \chi_{h_i} \in \mathcal{P}(b)$. By Claim 2.19 we get that $x|_{S-H_i} \in \mathcal{P}(b^{H_i})$, thus $b^{H_i}(H_j) = b(H_j)$ for all $i \neq j$. This implies that we obtain the same polymatroid function on groundset $S - \cup \mathcal{H}$ no matter which order the sets H_i are projected. Let $b^{\mathcal{H}}$ denote the unique polymatroid function obtained by projecting all the members of \mathcal{H} . Then, by Claim 2.19,

$$(47) \quad \mathcal{P}(b^{\mathcal{H}}) = \{y \in \mathbb{R}^{S-\cup \mathcal{H}} : \text{there is } [y, z] \in \mathcal{P}(b) \\ \text{such that } z(H_i) = b(H_i) - 1 \text{ for } i = 1, \dots, m\},$$

and we get that for any $X \subseteq S - \bigcup \mathcal{H}$,

$$(48) \quad b^{\mathcal{H}}(X) = \min \{b(X \cup \bigcup \mathcal{H}') - x(\bigcup \mathcal{H}') : \mathcal{H}' \subseteq \mathcal{H}\}.$$

Finally, we prove that there is a strongly polynomial time algorithm to evaluate $b^{\mathcal{H}}$. Here we make use of a strongly polynomial time algorithm for maximum weight common base of two polymatroids, given in Schrijver [77]. (We remark that a maximum weight common base is the biggest gun around, and is cited for momentary comfort. Recall that the base polyhedron of a polymatroid function b on groundset S is defined by $\mathcal{BP}(b) = \mathcal{P}(b) \cap \{x : x(S) = b(S)\}$.) Notice that

$$\begin{aligned} \mathcal{BP}_1 &:= \{[z_S, z_a] : z_a = b(S) - z_S(S), z_S \in \mathcal{P}(b)\}, \text{ and} \\ \mathcal{BP}_2 &:= \{[z_S, z_a] : z_a = b(S) - z_S(S), z_S(H_i) = b(H_i) - 1\} \end{aligned}$$

are base polyhedra over the common groundset $S' := S + \{a\}$. It follows from (47) that $b^{\mathcal{H}}(X) = \max\{z \cdot \chi_X : z \in \mathcal{BP}_1 \cap \mathcal{BP}_2\}$ holds for any $X \subseteq S - \bigcup \mathcal{H}$. Thus, $b^{\mathcal{H}}(X)$ may be determined via a maximum weight common base algorithm in strongly polynomial time.

Claim 2.20 $b^{\mathcal{H}}$ may be evaluated in strongly polynomial time. □

Blossoms

The notion of blossoms comes from an algorithmic point of view, which is the analogue of Edmonds' blossoms in the matching algorithm, or the analogue of hypo-matchable components in a very strong cover in Orlin and Vande Vate's linear matroid matching algorithm. An "ear-decomposition" of a matching is constructed by finding a circuit induced in the matching, and iterating this procedure after the projection. More precisely, the definition is the following.

Consider a matching x with respect to a polymatroid function $b : 2^S \rightarrow \mathbb{N}$. Consider a laminar family $\mathcal{F} = \{B_1, \dots, B_k\}$ of subsets of S , that is, any two members of \mathcal{F} are either disjoint or one contains the other. For indices $i = 1, \dots, k$, let \mathcal{F}_i denote the family of inclusionwise maximal proper subsets of B_i in \mathcal{F} , and let $G_i := B_i - \bigcup \mathcal{F}_i$. Consider a set $U = \{u_1, \dots, u_k\} \subseteq S$ such that $u_i \in G_i$. Hence \mathcal{F}, U is called an x -**ear-decomposition** if

$$(49) \quad x(B_i) = b(B_i) - 1, \text{ and}$$

$$(50) \quad u_i \text{ induces a circuit on } G_i \text{ in } x|_{S - \bigcup \mathcal{F}_i} \text{ with respect to } b^{\mathcal{F}_i}.$$

Notice that the above definition implies that $x + \chi_{u_i} \in \mathcal{P}(b)$ holds whenever B_i is an inclusionwise minimal member of \mathcal{F} . This implies that the projection of \mathcal{F} , or

\mathcal{F}_i satisfies the assumption in the previous section, and thus the projection may be performed in arbitrary order. Notice, if we drop an inclusionwise maximal member $B_i \in \mathcal{F}$ together with u_i , we retain another ear-decomposition. A set B appearing in the family \mathcal{F} of some ear-decomposition is called an x -**blossom**. An ear-decomposition of a blossom B is an ear-decomposition \mathcal{F}, U such that B is the unique inclusionwise maximal member of \mathcal{F} .

The following Lemma 2.21 will be our crucial inductive tool to deal with ear-decompositions by extending a matching with respect to $b^{\mathcal{F}}$ to a matching with respect to b .

Lemma 2.21 *Suppose we are given a matching x , an x -blossom B together with an x -ear-decomposition, and a vector $y \in \mathcal{P}(b^B)$. There is a polynomial time algorithm to find either*

(51) *a ntcde, or*

(52) *an even vector $z \in (2\mathbb{N})^B$ such that $z(B) = b(B) - 1$ and $[z, y] \in \mathcal{P}(b)$.*

Proof. Let us use notation from above. The algorithm is recursive on the number k of ears. Firstly, notice that $def_b([x|_B y,]) \leq 1$. If $def_b([x|_B, y]) = 0$, then (52) holds for $z = x|_B$, and we are done. Henceforth we suppose that $def_b([x|_B, y]) = 1$, and let D denote the fundamental circuit of $[x|_B, y]$. Say $B = B_k$ and $G = G_k$.

We claim that either $[x|_G, y] \in \mathcal{P}(b^{\mathcal{F}_k})$, or $D \subseteq (S - B) \cup G$. Suppose $[x|_G, y] \notin \mathcal{P}(b^{\mathcal{F}_k})$. By Claim 2.19, there is a set Q such that $def_{b, [x|_B, y]}(Q) \geq 1$, and for all $B_i \in \mathcal{F}_k$ we have $Q \cap B_i = \emptyset$ or $Q \supseteq B_i$. Clearly, $def_{b, [x|_B, y]}(B) = -1$. Since $y \in \mathcal{P}(b^B)$, we get that $def_{b, [x|_B, y]}(B \cup Q) \leq 0$. Thus, by supermodularity of deficiency, $0 \leq def_{b, [x|_B, y]}(B \cap Q) = def_{b, x}(B \cap Q)$. Recall that for every inclusionwise minimal set $B_i \in \mathcal{F}$ we have $x + \chi_{u_i} \in \mathcal{P}(b)$ for $u_i \in B_i$. Thus, $u_i \notin B \cap Q$, which implies that $D \subseteq Q \subseteq (S - B) \cup G$.

Now suppose that $[x|_G, y] \in \mathcal{P}(b^{\mathcal{F}_k})$. Thus, by Claim 2.19, there is a (not necessarily even) vector $z' \in \mathbb{N}^{\cup \mathcal{F}_k}$ such that $[z', x|_G, y] \in \mathcal{P}(b)$, and $z'(B_i) = b(B_i) - 1$ for all $B_i \in \mathcal{F}_k$. Then we apply the algorithm recursively for $B_i \in \mathcal{F}_k$ and $[z', x|_G, y]$, that is, we replace $z'|_{B_i}$ step-by-step by an even vector retaining the above properties – or we find a ntcde.

Finally suppose that $D \subseteq (S - B) \cup G$. Since $y \in \mathcal{P}(b^B)$, $D \cap B \neq \emptyset$. Since $x \in \mathcal{P}(b)$, $D - B \neq \emptyset$. Since $y \in \mathcal{P}(b^B)$, $def_{b, [y, x|_B]}(B \cup D) \leq 0$. Recall that $def_{b, [x|_B, y]}(D) = 1$ and $def_{b, [x|_B, y]}(B) = -1$. By supermodularity of deficiency, $def_{b, [x|_B, y]}(B \cap D) \geq 0$. Thus, by (50) we get that $u_k \notin D$. Consider an arbitrary element $d \in D \cap B$. By (50), $[x|_G + 2\chi_{u_k} - \chi_d, 0] \in \mathcal{P}(b^{\mathcal{F}_k})$. By applying the algorithm recursively for $[x|_G + 2\chi_{u_k} - \chi_d, 0]$ one can find either a ntcde, or an even vector $q \in (2\mathbb{N})^{\cup \mathcal{F}_k}$ such that $[q, x|_G + 2\chi_{u_k} - \chi_d, 0] \in \mathcal{P}(b)$. Next, we will find out whether

there is an element e such that $z = [q, x|_G + 2\chi_{u_k} - 2\chi_e]$ satisfies (52). Clearly, all these vectors are even. It is easy to see that $\text{def}_b([q, x|_G + 2\chi_{u_k}, y])$ is 1 or 2. If $\text{def}_b([q, x|_G + 2\chi_{u_k}, y]) = 1$, then for an arbitrary element e in its fundamental circuit we obtain $[q, x|_G + 2\chi_{u_k} - 2\chi_e, y]$, and we are done. If $\text{def}_b([q, x|_G + 2\chi_{u_k}, y]) = 2$, then let W denote the unique 2-deficient set. If there is an element $e \in W$ such that all the 1-deficient sets contain e , then $[q, x|_G + 2\chi_{u_k} - 2\chi_e, y] \in \mathcal{P}(b)$, and we are done. Otherwise, if for every element e there is a 1-deficient set $e \notin W_e$, then $[x|_W, 0]$ is a cdc. Notice that B and D are circuits in $[q, x|_G + 2\chi_{u_k}, y]$, thus $W - B \in \pi$ and $W - D \in \pi$. Since $d \in B \cap D \neq \emptyset$, $|\pi| \geq 3$. \square

A semi-strongly polynomial time algorithm

We construct a semi-strongly polynomial time algorithm which either returns a ntcde, or returns a matching x and a partition π such that $x(S)/2 = \text{val}(\pi)$. The algorithm maintains a matching, and iteratively augments its size by one, until it either finds a certifying partition, or a ntcde. We may initiate x as a basis of $\mathcal{P}(b)$, rounded down to the closest even vector. This initialization may be performed in semi-strongly polynomial time, where “semi-” comes only from the fact that we have to take lower integer part to detect parity. The remaining part of the algorithm may be performed in strongly polynomial time.

The idea behind the algorithm is the following. If our matching x is a basis in the polymatroid, then we are done. Otherwise, we will find an element $u \in S$ where we may add 1 to x and stay within the polymatroid. If 2 may be added to u , that gives a larger matching, and we are done. Thus, we may assume that adding 1 is feasible, but adding 2 is infeasible. In this case, u induces a circuit in x , which can be used building blossoms. Then, these blossoms are contracted. If we find a larger matching in the contraction, then we use Lemma 2.21 to expand blossoms and retain a larger matching over the original groundset. This idea is developed in detail below.

Consider a matching x . First we define $\mathcal{C} := \emptyset$. Then, in a general step of the algorithm, $\mathcal{C} = \{B_1, \dots, B_k\}$ is a family of disjoint x -blossoms. This implies that $x|_{S-\cup\mathcal{C}} \in \mathcal{P}(b^\mathcal{C})$. We distinguish three cases on how close $x|_{S-\cup\mathcal{C}}$ is to a basis of $\mathcal{P}(b^\mathcal{C})$.

Case 1. Suppose that $x(S - \cup\mathcal{C}) = b^\mathcal{C}(S - \cup\mathcal{C})$. Then, by claim (48), there is a set $\mathcal{C}' \subseteq \mathcal{C}$ such that $b(S - \cup\mathcal{C} + \cup\mathcal{C}') = x(S - \cup\mathcal{C}' + \cup\mathcal{C}')$. Then $\mathcal{C}' = \emptyset$, since for all blossoms $B_i \in \mathcal{C}$ there is an element $t \in B_i$ such that $x + \chi_t \in \mathcal{P}(b)$. We conclude that $x(S)/2 = \text{val}(\mathcal{C} \cup \{S - \cup\mathcal{C}\})$.

Case 2. Suppose that $x|_{S-\cup\mathcal{C}} + \chi_u \in \mathcal{P}(b^{\mathcal{C}})$, but $x|_{S-\cup\mathcal{C}} + 2\chi_u \notin \mathcal{P}(b^{\mathcal{C}})$. Then there is a set $u \in Z \subseteq S - \cup\mathcal{C}$ such that u induces a circuit on Z in $x|_{S-\cup\mathcal{C}}$ with respect to $b^{\mathcal{C}}$. By claim (48) there is a set $\mathcal{C}' \subseteq \mathcal{C}$ such that $b(Z \cup \cup\mathcal{C}') = x(Z \cup \cup\mathcal{C}') + 1$. Thus, $\mathcal{C} - \mathcal{C}' + \{Z \cup \cup\mathcal{C}'\}$ is a blossom family.

Case 3. Suppose that $x|_{S-\cup\mathcal{C}} + 2\chi_u \in \mathcal{P}(b^{\mathcal{C}})$. In this case, by applying Lemma 2.21 for members of \mathcal{C} , we construct either a matching larger than x , or a ntcdc. This is done as follows. By assertion (47), there is a (not necessarily even) vector $z \in \mathbb{N}^{\cup\mathcal{C}}$ such that $x' := [x|_{S-\cup\mathcal{C}} + 2\chi_u, z] \in \mathcal{P}(b)$, and $z(B_i) = b(B_i) - 1$ for $i = 1, \dots, k$. Thus, for an arbitrary index $i \in \{1, \dots, k\}$ we get that $x'|_{S-B_i} \in \mathcal{P}(b^{B_i})$. By applying Lemma 2.21 for B_i , we either construct a ntcdc, or we may replace entries of x' in B_i with even numbers, and retain the above properties. By repeating this procedure for $i = 1, \dots, k$ we retain a matching x' that is larger than x .

2.4 Matroid fractional matching

The main result in this section will be Vande Vate's formula for a fractional version of matroid matching. Matroid matching is known as a famous common generalization of graph matching and matroid intersection. Similarly, matroid fractional matching is a common generalization of matroid intersection and graph fractional matching. One could argue that graph fractional matching reduces to matroid intersection, and there is no need for such a common generalization. However, there is more about matroid fractional matching than being this common generalization. We will exhibit another special case of matroid fractional matching, which is the fractional packing of \mathcal{A} -paths. It is unknown whether this problem reduces to matroid intersection, hence it seems matroid fractional matching has a strictly wider range of applications, than matroid intersection has.

We will show that the half-integrality property of graph fractional matching is shared by matroid fractional matching in general. Also note that matroid fractional matching is shown tractable for arbitrary matroids given by an independence oracle, as opposed to matroid matching, which is only known tractable for special classes of matroids such as representable matroids.

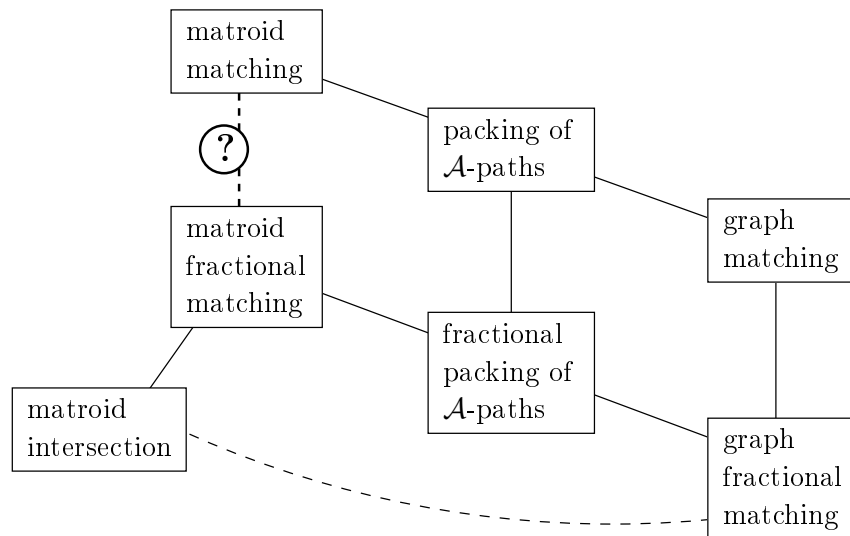


Figure 7: Problems in relation with matroid fractional matching.

The main result on matroid fractional matching is Vande Vate's Theorem 2.22, which provides a min-max formula to determine the maximum value of a fractional matching. In some sense, this min-max formula is inbetween Lovász' Theorem 2.6 on linear matroid matching, and Edmonds' Matroid Intersection Formula. The second result is that the optimum is always attained by some fractional matching which is half-integral. One can show, in fact, that the polytope of all fractional matchings is

half-integer, and it is given by a totally dual half-integral description. A short proof of total dual half-integrality has recently been given by D. Gijswijt.

An instance of matroid matching is given by a matroid $\mathcal{M} = (S, \mathcal{I})$ and a set E of **pairs**, i.e. (disjoint) two-element subsets. A **matching** is a set of disjoint pairs the union of which is independent. As for the matroid matching problem, we are looking for a maximum cardinality matching. In this section we are concerned with a fractional relaxation of matroid matching, which is tractable in arbitrary matroids, as observed by Vande Vate [7]. It is interesting that – as opposed to matroid matching, where we can only deal with special kinds of matroids – we can solve fractional matching in arbitrary matroids.

An instance of matroid fractional matching is given by a matroid $\mathcal{M} = (S, \mathcal{I})$ and a set E of **pairs**. To define fractional matchings, we consider vectors in which we assign non-negative real numbers to each pair, that is, $x \in \mathbb{R}_+^E$. Those vectors x obeying some upper bound constraint will be considered fractional matchings.

There are several alternative possibilities to impose upper bound constraints on such vectors. The first possible reasonable (but unuseful) definition that comes into our mind is the following. We require that the vector $\sum x(e)\chi_e$ is in the independent set polytope of \mathcal{M} . By Edmonds' description of the independent set polytope, this could be formulated as a family of upper bound constraints on x . However, this definition is not worthy of long investigation, since the set of these vectors x is equal to a polymatroid on the ground-set E . Thus, we will not put our effort into this proposed “possible” definition. We should rather be concerned with the following, inequivalent definition.

Consider a finite or infinite matroid $\mathcal{M} = (S, \mathcal{I})$ and a finite set E of pairs (i.e. two-element subsets of S). For any subset $K \subseteq S$, we define its degree-vector $d_K \in \{0, 1, 2\}^E$ by $d_K(e) := r(K \cap sp(e))$ for $e \in E$. Let $\mathcal{P} = \mathcal{P}(\mathcal{M}, E)$ denote the set of **fractional matchings**, i.e. those vectors $x \in \mathbb{R}^E$ fulfilling the following inequalities.

$$(53) \quad x \geq 0$$

$$(54) \quad d_K \cdot x \leq r(K) \quad \text{for } K \subseteq S.$$

Some remarks on this polytope:

- $x \leq \mathbf{1}$ follows from inequality (54).
- For any given set K , inequality (54) follows from replacing K by $sp(K)$ – hence it suffices to substitute K by flats.
- On the other hand, (54) follows from replacing K by a subset K' of K such that $sp(K' \cap sp(e)) = sp(K \cap sp(e))$ for all $e \in E$. Such a set K' is obtained by

picking 0, 1, or 2 elements from $sp(e)$. Hence it suffices to impose inequalities (54) for sets K of cardinality at most $2|E|$.

- It is important that we allow that matroid \mathcal{M} be infinite (for example the full linear matroid over a vectorspace), and that $\bigcup E$ may be a proper subspace of S . Thus, we consider inequalities (54) for sets K which are not spanned by elements in $\bigcup E$, and some of these inequalities may be irredundant in the description!
- If \mathcal{M} is infinite, then we consider an infinite number of inequalities in the above description. And even if \mathcal{M} is finite, the number of inequalities may be exponential in $|S|$. So it is not easy (unsolved, in fact) to test membership in \mathcal{P} of an arbitrary rational vector.
- However, there is a finite number of distinct vectors $d_K \in \{0, 1, 2\}^E$, hence, independent of the nature of the matroid, \mathcal{P} is a nonempty polytope.
- In mathematical folklore, “fractional relaxation of a discrete problem” stands for a continuous problem in which integer feasible solutions correspond one-to-one with feasible solutions of the discrete problem. Now, consider an integral fractional matching x , i.e. a vector $x \in \mathcal{P} \cap \mathbb{Z}^E$. Then x has 0,1 entries, say $x = \chi_F$ for a set $F \subseteq E$ of pairs. Substituting $K = \bigcup F$ in inequality (54), we get that $2|F| \leq r(\bigcup F)$, and thus, F a matching. Conversely, consider a matching M and a set $K \subseteq S$. Then $\{K \cap sp(e) : e \in M\}$ is a direct sum, thus $d_K \cdot \chi_M = \sum_{e \in M} r(K \cap sp(e)) = r(\bigcup \{K \cap sp(e) : e \in M\}) \leq r(K)$. This proves that, in the above sense, matroid fractional matching is a fractional relaxation of matroid matching.

Vande Vate’s min-max formula

The **size** of a vector x is defined by $\mathbf{1} \cdot x$. Let $\nu^*(\mathcal{M}, E)$ denote the maximum size of a fractional matching. In the matroid fractional matching problem, we have to find a maximum size fractional matching. To provide a good characterization, or a min-max formula, we have to find an upper bound on the size of fractional matchings.

Consider an arbitrary set $K \subseteq S$. Let $E_K := \{e \in E : r(K \cap sp(e)) = 0\}$ and $L := K \cup \bigcup E_K$. We claim that

$$val^*(K) := \frac{1}{2}r(K) + \frac{1}{2}r(L) = r(K) + \frac{1}{2}r_{\mathcal{M}/K}(\bigcup E_K)$$

is an upper bound on the size of any fractional matching x . To prove this, notice that $d_K(e) = 0$ implies $d_L(e) = 2$. Moreover, $d_K \leq d_L$, and thus we get that $\frac{1}{2}d_K(e) + \frac{1}{2}d_L(e) \geq 1$ holds for any pair $e \in E$. Thus,

$$\mathbf{1} \cdot x \leq \frac{1}{2}d_K \cdot x + \frac{1}{2}d_L \cdot x \leq \frac{1}{2}r(K) + \frac{1}{2}r(L) = \text{val}^*(K).$$

The main theorem on fractional matchings is the min-max formula of Vande Vate, which claims that the upper bound $\text{val}^*(K)$ is tight for some set K .

Theorem 2.22 (Vande Vate [7]) *For any finite or infinite matroid \mathcal{M} , and any finite set E of pairs, the following equality holds.*

$$(55) \quad \nu^*(\mathcal{M}, E) = \min_{K \subseteq S} \text{val}^*(K).$$

Primal and dual half-integrality

Vande Vate proved the following half-integrality result, which is a generalization of the well-known half-integrality result on fractional matching in undirected graphs. A polytope is called **half-integral**, if all of its vertices are half-integral vectors. A description of a polytope is called **totally dual half-integral**, if for any integer cost-function, the dual system has a half-integral optimum solution. (For more on integrality, see Schrijver, [77].)

Theorem 2.23 (Vande Vate [8]) $\mathcal{P}(\mathcal{M}, E)$ is a half-integral polytope.

Recently, D. Gijswijt [36] elegantly proved the following strengthening of Vande Vate's half-integrality result which implies Vande Vate's min-max formula in Theorem 2.22, as well.

Theorem 2.24 (Gijswijt [36]) (53)-(54) is a totally dual half-integer description of $\mathcal{P}(\mathcal{M}, E)$.

Open questions

- Can matroid fractional matching be reduced to an instance of matroid matching? (Fractional matching in a non-bipartite graph reduces to matching in an auxiliary bipartite graph. The above question asks for a generalization of this reduction.)

- Is there a polynomial time algorithm to find a maximum weighted fractional matching with respect to an arbitrary objective w ? (Vande Vate [7] constructed a polynomial time algorithm to find a maximum size fractional matching, that is, maximizing with respect to objective $w = 1$.)
- Is there a polynomial time separation algorithm for $\mathcal{P}(\mathcal{M}, E)$? (Vande Vate [7] constructed a polynomial time algorithm to test membership of a half-integer vector in $\mathcal{P}(\mathcal{M}, E)$.)

Special cases

- Graph fractional matching. Consider an undirected graph $G = (V, E)$. $\mathcal{M} = (S, \mathcal{I})$ is defined as the free matroid over groundset $S := V$, and E is also considered as the set of pairs in the matroid. It is easy to see that a vector over E is a fractional matching in G if and only if it is a fractional matching in \mathcal{M} . We remark that the well-known half-integrality result of graph fractional matching generalizes to matroid fractional matching, see below.
- Matroid intersection. Consider two matroids $\mathcal{M}_1, \mathcal{M}_2$ on the same groundset S . Let \mathcal{M} be defined as the direct sum of \mathcal{M}_1 and a copy of \mathcal{M}_2 on groundset S' . E is defined as the set of pairs of an element and its copy. The maximum size of a fractional matching is equal to the maximum cardinality of a common independent set. We remark that $\mathcal{P}(\mathcal{M}, E)$ is isomorphic to the common independent set polytope.
- Fractional packings of \mathcal{A} -paths. It is well-known that the problem of packing fully node-disjoint \mathcal{A} -paths reduces to linear matroid matching. Fractional matching with respect to the same set of pairs in the same matroid is equivalent with fractional packing of \mathcal{A} -paths. The half-integrality results below are related to the half-integrality results on \mathcal{A} -path, which will be presented in a later section.

Proofs

Proof of Theorem 2.24. (This proof is due to D. Gijswijt [36].) Let \mathcal{L} denote the set of flats of \mathcal{M} . Then $\mathcal{P}(\mathcal{M}, E)$ is equal to the set of solutions of

$$(56) \quad x \geq 0$$

$$(57) \quad d_K \cdot x \leq r(K) \quad \text{for } K \in \mathcal{L}.$$

Consider the linear program of maximizing $w \cdot x$ over system (56)-(57) for some $w \in \mathbb{N}^E$. The dual to this program, concerning vectors $y \in \mathbb{R}^{\mathcal{L}}$, is maximizing $\sum_{K \in \mathcal{L}} r(K)y(K)$ over

$$(58) \quad y \geq 0$$

$$(59) \quad d_e \cdot y \leq w(e) \quad \text{for } e \in E,$$

where $d_e \in \{0, 1, 2\}^{\mathcal{L}}$ such that $d_e(K) := r(sp(e) \cap K)$ for $K \in \mathcal{L}$. Consider a dual optimum y such that $\sum_{K \in \mathcal{L}} r(K)y(K)^2$ is minimum.

We claim that $\mathcal{L}^+ := \{K \in \mathcal{L} : y(K) > 0\}$ is a chain. Suppose for contradiction that $y(K_1), y(K_2) > 0$, and $K_1 - K_2, K_2 - K_1 \neq \emptyset$. Then, decreasing $y(K_1)$ and $y(K_2)$ by some small positive ϵ , and increasing $y(K_1 \cap K_2)$ and $y(sp(K_1 \cup K_2))$ by ϵ provides a dual optimum y' . By submodularity, $\sum_{K \in \mathcal{L}} r(K)y'(K)^2 < \sum_{K \in \mathcal{L}} r(K)y(K)^2$, a contradiction.

Thus, \mathcal{L}^+ is a chain, say $\mathcal{L}^+ = \{K_1, K_2, \dots, K_k\}$ such that $K_i \subseteq K_j$, whenever $i \leq j$. This implies that $D := [d_{K_1}, d_{K_2}, \dots, d_{K_k}]$ is a matrix of all entries 0, 1, 2, and such that $d_{K_i} \leq d_{K_j}$ whenever $i \leq j$. To prove that there is a half-integer dual optimum y , it suffices to show that every non-singular square submatrix of D has a half-integral inverse. This follows from the below lemma.

Lemma 2.25 *Suppose $b \in \mathbb{Z}^n$, and $A \in \{0, 1, 2\}^{n \times n}$ is non-singular such that its column sums are at most 2. Then the unique solution of $Ax = b$ is half-integer.*

Proof. The lemma is proved by induction on n . If some entry x_i of x is integer, then we are done by deleting the i th column and some row, and applying the inductive hypothesis. Henceforth, assume that all x_i are non-integer. This implies that all column sums are equal to 2. If an entry of A is 2, then we are done by the induction, applied for the deletion of the column and row of that entry 2. Henceforth, we may assume that all columns contain two 1's. Thus A is the edge-node incidence matrix of a simple undirected graph $G = (V, E)$, and x is the unique(!) perfect b -matching in G . Note that $|V| = |E| = n$. Now suppose that $d_G(v) = 1$ for some $v \in V$, $vz \in E$. Then $x(vz) = b(v)$ is integer, which contradicts our assumption. If all degrees are at least 2, then, since $|V| = |E|$, G must be the union of cycles. The incidence matrix of only odd cycles is non-singular. The half-integrality of x is then straightforward. \square

Proof of Theorem 2.22. The maximum size of a fractional matching is equal to the maximum of $\mathbf{1} \cdot x$ subject to $x \in \mathcal{P}(\mathcal{M}, E)$. Consider a half-integral optimum y of the dual program. As in the proof of Theorem 2.24, we may assume that the family $\mathcal{L}^+ := \{K \in \mathcal{L} : y(K) > 0\}$ is a chain. Since y is a half-integral vector, this implies that y either has one positive value equal to 1, or two positive values equal

to $\frac{1}{2}$. Thus, there are two flats $K \leq L$ of the matroid such that $y = \frac{1}{2}\chi_K + \frac{1}{2}\chi_L$ holds. Now, y must satisfy (57), which implies that $L \geq K \cup \{e : r(K \cap sp(e)) = 0\}$, and hence $1 \cdot y \geq val^*(K)$. \square

Fractional matching in linear matroids

Fractional matching makes sense, and the above proof of half-integrality work out for infinite matroids, as well. Even though in the description of $\mathcal{P}(\mathcal{M}, E)$ we consider an infinite number of inequalities, the total dual half-integrality implies that $\mathcal{P}(\mathcal{M}, E)$ is always a polytope. The question is, in whether we should expect a polynomial time algorithm in the infinite case. If, for example, \mathcal{M} is a full linear matroid, and the pairs in E are given by a representation, then a positive answer may be given, and a proof is sketched in the next paragraph. Putting \mathcal{M} as a full linear matroid provides important special cases: for example fractional packing of \mathcal{A} -paths. Thus, a polynomial time algorithm for fractional packing follows from a polynomial time algorithm for fractional matching in linear matroids. Such an algorithm may be constructed by first applying Orlin and Vande Vate's linear matroid matching algorithm, and then performing some adjustments in the end. A sketch of this algorithm goes as follows.

Suppose E consists of k pairs represented by vectors in \mathbb{Q}^n , and \mathcal{M} is defined as the full linear matroid of \mathbb{Q}^n . A maximum fractional matching in E may be constructed as follows. Apply Orlin and Vande Vate's linear matroid matching algorithm for E , and perform the following adjustments in the end. Suppose a vector q is in the span of a K -hypo-matchable component A_i , but not in the span of our maximum matching. Then we will replace the hypo-matching by a perfect fractional matching in A_i/K . A perfect fractional matching in a A_i/K may be constructed as follows. Pick a line e in A_i/K , and let r_1, r_2 be two distinct vectors in e . Let N_1, N_2 be K -hypo-matchings such that $r_i \notin sp_{V/K}(N_i)$. Then $\frac{1}{2}(\chi_{N_1} + \chi_{N_2} + \chi_e)$ is a perfect fractional matching in A_i/K .

A remark on Theorem 2.22

A major question is, whether matroid fractional matching reduces to matroid matching. To the authors best knowledge, this question is unsolved. However, there is a great analogy between Vande Vate's min-max formula and results on graph (fractional) matching; here we make a remark on this.

Consider a simple undirected graph $G = (V, E)$. For a set $X \subseteq V$, let $\mathcal{K}(X)$ denote the family of components of the subgraph $G - X$. An alternative to the

Berge-Tutte Formula is that the maximum cardinality of a matching in G is equal to

$$(60) \quad \min_{X \subseteq V} |X| + \sum_{K \in \mathcal{K}(X)} f(|V(K)|),$$

where $f(a) := \lfloor \frac{a}{2} \rfloor$.

Fractional matching in a simple undirected graph $G = (V, E)$ is a vector $x \in \mathbb{R}_+^E$ such that $\delta_x(v) \leq 1$ for every node $v \in V$. There is a well-known min-max formula that relates the maximum size of a fractional matching to a dual expression that is equivalent with expression (60), by substituting the function f by the following function:

$$f^*(a) := \begin{cases} 0 & \text{for } a = 1 \\ a/2 & \text{otherwise.} \end{cases}$$

Lovász' linear matroid matching formula claims that, the maximum cardinality of a matching is equal to the minimum value of a cover K, π . Recall that a cover is a pair of a set $K \subseteq S$ and a partition π is of E . Notice that the value of a cover may be expressed by using function f , and one should not hesitate replacing the function f by f^* , and see whether this makes sense with fractional matching. Hence, for a cover K, π we define

$$val^*(K, \pi) := r(K) + \sum_{A \in \pi} f^*(r_{\mathcal{M}/K}(A)).$$

$val^*(K, \pi)$ is easily seen as an upper bound on the value of fractional matchings. We claim, that this upper bound is equivalent with the upper bound in Vande Vate's Theorem 2.22, that is,

$$\min_K val^*(K) = \min_{K, \pi} val^*(K, \pi).$$

3 Packing A -paths

The expression “path-packing” stands for a family of pairwise fully node-disjoint paths in an undirected graph $G = (V, E)$. A -paths are those paths joining two distinct nodes of a pre-specified set $A \subseteq V$ of terminals. The subject of this section is packing a maximum number of A -paths, or more generally, packing a maximum number of allowed A -paths. Thus, a path-packing problem is specified by a subset \mathcal{F} of all A -paths as the set of allowed A -paths, while all other paths are considered forbidden in a packing. We discuss those sets \mathcal{F} which imply a tractable path-packing problem. We remark that many results have been proposed for edge-disjoint families of A -paths, instead of node-disjoint. Since edge-disjoint paths reduce to node-disjoint paths in the line-graph, and most of the edge-disjoint results generalize to node-disjoint results, we will only work on the node-disjoint setting. Also note that there is a wide range of path-packing problems that are intractable, however, a comprehensive description of these intractability results goes beyond the goals of this section, and we will rather focus on tractable path-packing problems.

Path-packing, as it is a common generalization of non-bipartite matching and packing disjoint S – T -paths, has always been a prominent topic in combinatorial optimization, and has witnessed some of the most astonishing results in the 60’s and 70’s. Let us mention Gallai’s Formula [33] on node-disjoint A -paths in 1961, and Mader’s Formula [58] on node-disjoint \mathcal{A} -paths in 1978. In the early 80’s, path-packing turned out to be an attractive special case of Lovász’ theory of matroid matching, which brought new light onto the topic of path-packing. Recently, Chudnovsky et al. [10] claimed a deep generalization of Mader’s result, concerning so-called non-zero A -paths.

In a subsequent section, we will survey well-known results on path-packing frameworks, including those mentioned above. Packing \mathcal{A} -paths occupies a central position in Figure 8, depicting relations between with other frameworks. Hence it will be useful to discuss all those results in relation with Mader’s remarkable formula, which will be in the center of our attention throughout the section.

Mader’s Theorem on node-disjoint \mathcal{A} -paths

Consider an undirected graph $G = (V, E)$, and a set $A \subseteq V$ of so-called **terminals**. A path is called an **A -path** if its ends are in two distinct terminals, and its internal nodes are non-terminals. Let $\widehat{\nu}(G, A)$ denote the maximum cardinality of a packing of A -paths. Gallai [33] proved a min-max formula for $\widehat{\nu}(G, A)$, by a reduction to non-bipartite matching in an auxiliary graph.

Next, suppose A is a partitioned into a family $\mathcal{A} = \{A_1, \dots, A_k\}$ of disjoint

node-sets A_i called **terminal sets**. An A -path is called an **\mathcal{A} -path** if its ends are in two distinct terminal sets. Let $\nu(G, \mathcal{A})$ denote the maximum cardinality of a packing of \mathcal{A} -paths. (Thus, if \mathcal{A} is the partition of A into singletons, then $\nu(G, \mathcal{A}) = \widehat{\nu}(G, A)$.) It is unknown whether packing \mathcal{A} -paths reduces to non-bipartite matching. A reduction to linear matroid matching has been given by Schrijver [77]. Thus, packing \mathcal{A} -paths seems more difficult than packing A -paths.

Mader's Formula is attractive since it generalizes in a natural way results of Berge and Tutte on non-bipartite matching, Gallai's result on node-disjoint A -paths, and Menger's theorem on S - T -paths. As a common generalization of these results, we have to carry over some of their features to Mader's Formula. Thus, in Mader's Formula we take into consideration on the one hand, nodesets partially separating some terminals, and on the other hand, the parity of blocking nodesets. There are equivalent ways to define the dual expression in Mader's Formula, here we use the notation proposed by Sebő and Szegő [80].

A family $\mathcal{X} = \{X_0; X_1, \dots, X_k\}$ of disjoint subsets of V is called an **\mathcal{A} -partition** if $A_i \subseteq X_0 \cup X_i$ for $i = 1, \dots, k$. Let $X := \bigcup_{i=1}^k X_i$. Let $\mathcal{K}(\mathcal{X})$ denote the family of **components** of the subgraph $G(\mathcal{X}) = G - X_0 - \bigcup_{i=1}^k E[X_i]$. The **value** of an \mathcal{A} -partition is defined by

$$(61) \quad \text{val}(G, \mathcal{X}) := |X_0| + \sum_{K \in \mathcal{K}(\mathcal{X})} \left\lfloor \frac{|V(K) \cap X|}{2} \right\rfloor$$

Theorem 3.1 (Mader [58]) *Let $G = (V, E)$ be an undirected graph, and let $\mathcal{A} := \{A_1, \dots, A_k\}$ be a partition of $A \subseteq V$. Then*

$$(62) \quad \nu(G, \mathcal{A}) = \min \text{val}(G, \mathcal{X})$$

where the minimum is taken over \mathcal{A} -partitions \mathcal{X} .

A survey of results on path-packing

In this section we survey path-packing frameworks, and we cite results papers on min-max formulae and algorithms. Our list includes most of those frameworks depicted in Figure 8, where we connect two boxes by a line if the framework in the higher standing box generalizes the framework in the lower standing box.

- **Node-disjoint S - T -paths.** For two disjoint sets $S, T \subseteq V$, setting $\mathcal{A} := \{S, T\}$ returns the node-disjoint S - T -paths problem. In 1927, Menger [61] related the maximum number of disjoint S - T -paths to S - T cuts. A polynomial time algorithm can be constructed by a reduction to network flow, or bipartite matching.

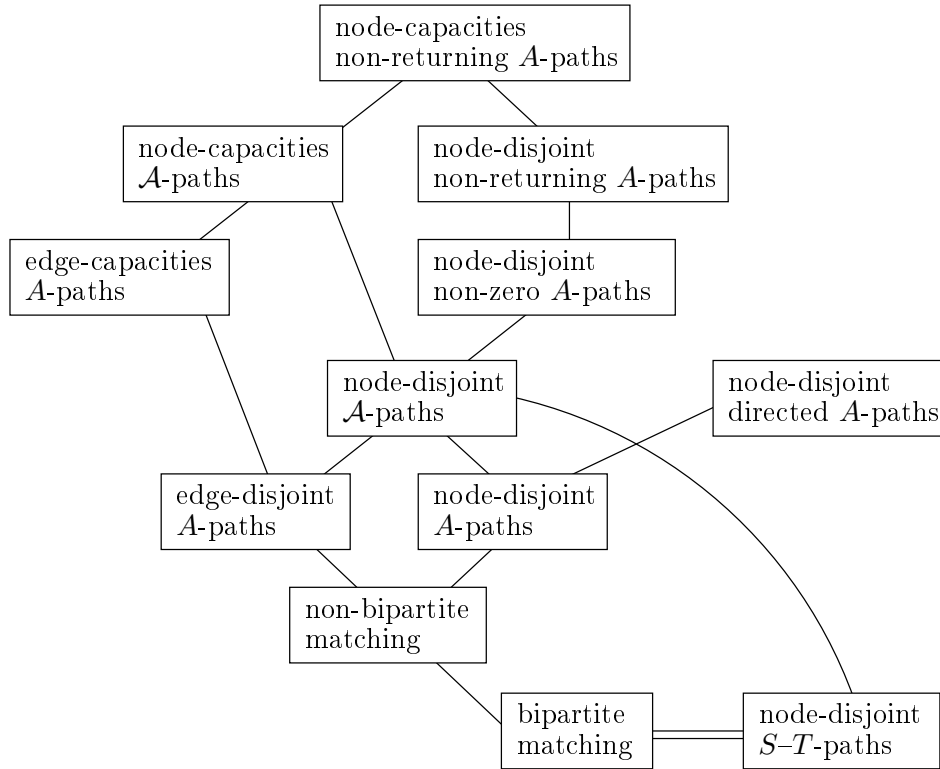


Figure 8: Relations between tractable path-packing problems.

- Non-bipartite matching.** Setting $\mathcal{A} := \{\{v\} : v \in V\}$ returns the maximum matching problem in graph $G = (V, E)$. In 1947, W. Tutte [84] characterized the existence of a perfect matching. In 1958, C. Berge [4] proved a min-max formula on the maximum cardinality of a matching in arbitrary graphs. In 1967, J. Edmonds [21] constructed his famous non-bipartite matching algorithm. Gallai [33] and Edmonds [21] independently proved a structural description known as the Edmonds-Gallai decomposition.
- Node-disjoint A -paths.** Setting $\mathcal{A} := \{\{v\} : v \in A\}$ returns A -paths as a special case. Recall that, in this case, $\widehat{\nu}(G, A) = \nu(G, \mathcal{A})$. In 1961, T. Gallai observed that node-disjoint A -paths reduces to maximum matching by means of the following elementary construction. Define $G' = (V', E')$ by $V' := A \cup \{v', v'' : v \in V - A\}$ and $E' := \{av', av'' : av \in E, a \in A, v \in V - A\} \cup \{v'z'' : vz \in E[V - A]\} \cup \{v'v'' : v \in V - A\}$. Then it is quite easy to prove that $\widehat{\nu}(G, A) = \nu(G') - |V - A|$. This reduction implies a min-max formula, and a polynomial time algorithm for node-disjoint A -paths. In 2003, Kriesell [48] observed that, by using Gallai's reduction to non-bipartite matching, one can also prove a result on packing directed A -paths in directed graphs. This implies a min-max formula, and a polynomial time algorithm, as

well.

- **Edge-disjoint \mathcal{A} -paths** arises a special case by substituting G by its line-graph $L(G)$, and defining $\mathcal{A} := \{\{e : e \in \delta_G(a)\} : a \in A\}$. In 1972, Lovász [53] and Cherkassky [9] independently observed that edge-disjoint \mathcal{A} -paths is easy if the so-called “Euler-condition” holds, i.e. if every non-terminal node has an even degree. They proved that, in this case, the cut condition is necessary and sufficient, thus there is a packing that saturates the minimum cut between $t, A - t$, for every $t \in A$. A polynomial time algorithm is easy to construct by testing whether the deletion of an edge destroys any of the cut-conditions. We remark that, if G is not assumed Eulerian, then non-bipartite matching is a special case. Hence the cut-condition is no more sufficient, and parity has to be taken into consideration. In 1978, W. Mader [57] proved a min-max formula on edge-disjoint \mathcal{A} -paths, which he later generalized to the node-disjoint case.
- **Node-disjoint \mathcal{A} -paths.** In 1978, W. Mader [58] proved a min-max formula on packing node-disjoint \mathcal{A} -paths. Polynomial time algorithms have been given in the following papers. In 1980, Lovász [52] proved that packing \mathcal{A} -paths is equivalent with an instance of matroid matching. Then, Schrijver [77] represented \mathcal{A} -path packing with a linear matroid matching problem, and thus a polynomial time algorithm follows directly, from any linear matroid matching algorithm. Berki [5] applied the linear matroid matching algorithm of Gabow and Stallmann [31] to this special case. In 2004, Chudnovsky et al. [11] constructed a “structural algorithm” for non-zero \mathcal{A} -paths, which provides a new algorithm, and a structural description for Mader’s Formula, as well. Independently, in 2004, Sebő and Szegő [80] described an equivalent structure of the uniquely defined “canonical” optimum \mathcal{A} -partition.
- **Non-zero \mathcal{A} -paths.** In 2004, Chudnovsky et al. [10, 11] proposed the problem to find, in a given group-labelled graph, a maximum number of non-zero \mathcal{A} -paths. A **group-labelled graph** is given by an arbitrary group Γ , and an undirected graph such that we assign a group-element, and a reference-orientation to every edge. The weight of a path is defined as the product of the edge-labels in the order of edges appearing in the path, taking the inverse of the edge-label if the edge appears in reverse with respect to its reference-orientation. A path is called **non-zero** if its weight differs from the unit. The problem of packing non-zero \mathcal{A} -paths contains the problem of packing \mathcal{A} -paths as a special case by putting $\Gamma := (\mathbb{F}_2^+)^k$, and assigning $(0_1, 0_2, \dots, 1_i, \dots, 0_k)$ to edges leaving A_i , and $(0_1, 0_2, \dots, 0_k)$ to all other edges. Chudnovsky et al. [10] proved a min-max formula, and Chudnovsky et al. [11] provided a structural description, and a polynomial time algorithm.

- **Non-returning A -paths.** The author [64] proposed the problem of packing non-returning A -paths in a **permutation-labelled** graph, as a generalization of packing non-zero A -paths. In this setting, we are given G, A as above, and an arbitrary groundset Ω which disjoint from the graph. We assign with edges a reference orientation, and a permutation of Ω . We assign with nodes an element of Ω , which is called the node-potential. Any path induces a permutation of Ω by composing the permutation-labels of the edges, in which we consider the inverse of the permutation if the edge appears backwards with respect to the reference orientation. A path is called non-returning, if its induced permutation maps the node-potential of the first node to an element of Ω other than the node-potential of the last node. This setting generalizes the setting of non-zero paths in group-labelled graphs, just substitute the groundset by Γ , and let the multiplication by the edge-label be the permutation-label. We will discuss results on packing non-returning A -paths.
- **Fractional packing subject to edge- or node-capacities.** In this problem, we assign capacities with edges or nodes of the graph. A fractional packing is a non-negative function on the set of all \mathcal{A} -paths, which is constrained by the edge- or node-capacities. Lovász's [53] and Cherkassky's [9] result on the Euler-condition imply that, if we are given integer edge-capacities, then the maximum fractional packing is attained by a half-integral one. We will prove the analogous half-integrality result for integer node-capacities. We prove that fractional packing (for the all-one node-capacities) reduces to Vande Vate's matroid fractional packing problem. An analogue of the Euler-condition for node-capacities is unknown.
- **Packing A -paths subject to edge-capacities.** The problem is finding a maximum multiset of A -paths such that the number of paths traversing any edge does not exceed its capacity. This problem reduces to edge-disjoint A -paths by replacing an edge by a bunch of parallel edges. This reduction implies a min-max formula, and a weakly polynomial time algorithm. In 2000, Keijsper, Pendavingh and Stougie [43] proved a min-max formula, and constructed a polynomial time algorithm via the ellipsoid method.
- **Node-capacitated \mathcal{A} -path-packing.** The problem is finding a maximum multiset of \mathcal{A} -paths such that the number of paths traversing any node does not exceed its capacity. This problem reduces to packing \mathcal{A} -paths by splitting nodes into a number of copies. This reduction implies a min-max formula, and a weakly polynomial time algorithm. A semi-strongly polynomial time algorithm is given in this thesis.

Contribution of this thesis

- Primal and dual half-integrality of a linear programming description of node-capacitated packing.
- A semi-strongly polynomial time algorithm for node-capacities via the ellipsoid method.
- Existence of a “special” optimum 2-packing.
- A sketch of the proof that fractional packing reduces to matroid fractional matching.
- Introduction of the problem of packing non-returning A -paths, the proof of a simple min-max formula, and the construction of a polynomial time algorithm. We propose two other path-packing problems, and prove that they are equivalent with non-returning A -path-packing.
- A positive answer to Schrijver’s question “Is each Mader matroid a gammoid?”, [79, 77].

3.1 Node-capacitated packings of \mathcal{A} -paths

Using Mader's Theorem, we prove a generalization concerning node-capacities. In the node-capacitated problem, a node may be traversed by as many paths as its given capacity. Mader's Theorem follows from this result by imposing a constant one capacity. Conversely, node-capacitated packing may be solved in weakly polynomial time by splitting nodes into a number of its copies, and considering Mader's Theorem in this expanded graph. This is in fact the way we prove the min-max formula for node-capacities, suggested by Sebő and Szegő. Originally, Sebő and Szegő proved this theorem by exploiting their structural description of packings. Instead, our proof exploits the symmetry of the expanded graph, without using the structural description.

Consider G, \mathcal{A} as above, and a vector $b : V \rightarrow \mathbb{N}$ to specify node-capacities. A multi-set \mathcal{P} of \mathcal{A} -paths is called a **b -packing** if any node $v \in V$ is traversed by at most $b(v)$ of the paths in \mathcal{P} . Let $\nu_b(G, \mathcal{A})$ denote the maximum cardinality of a b -packing. For an \mathcal{A} -partition \mathcal{X} , let $\mathcal{K}_0 = \mathcal{K}_0(\mathcal{X})$ denote the family of components of $G(\mathcal{X})$ having at least two nodes, and let

$$(63) \quad \text{val}_b(G, \mathcal{X}) := b(X_0) + \sum_{K \in \mathcal{K}_0} \left\lfloor \frac{b(V(K) \cap X)}{2} \right\rfloor.$$

Theorem 3.2 (Node-capacitated version of Mader's Theorem – Sebő, Szegő [80]) *Let $G = (V, E)$ be an undirected graph, let $\mathcal{A} := \{A_1, \dots, A_k\}$ be a partition of $A \subseteq V$, and let $b : V \rightarrow \mathbb{N}$. Then*

$$(64) \quad \nu_b(G, \mathcal{A}) = \min \text{val}_b(G, \mathcal{X})$$

where the minimum is taken over \mathcal{A} -partitions \mathcal{X} .

Proof. The easy part of the proof is that $\text{val}_b(G, \mathcal{X})$ gives an upper bound on the cardinality of a b -packing. This follows from the following observation: Any \mathcal{A} -path either traverses at least one node of X_0 , or traverses at least two nodes of the border of some component. Mader's Theorem is a special case of Theorem 3.2 by substituting $b \equiv 1$.

We prove equality in (64) by a reduction to Mader's Theorem. The reduction is very simple. Consider a triple G, \mathcal{A}, b as above. We define the **expanded graph** $G' = (V', E')$ by $V' := \{(v, i) : v \in V, i \in [b(v)]\}$, $E' := \{(v, i)(z, j) : vz \in E, i \in [b(v)], j \in [b(z)]\}$, where $[m] = \{1, 2, \dots, m\}$. We define the terminal sets by $A'_i := \{(v, j) : v \in A_i, j \in [b(v)]\}$, and let $\mathcal{A}' := \{A'_1, \dots, A'_k\}$. It is straightforward that $\nu(G', \mathcal{A}') = \nu_b(G, \mathcal{A})$. By Mader's Theorem, there is an \mathcal{A}' -partition \mathcal{X}' such that $\text{val}(G', \mathcal{X}') = \nu(G', \mathcal{A}')$ holds. Let $\mathcal{X}' = \{X'_0; X'_1, \dots, X'_k\}$ be an \mathcal{A}' -partition minimizing $\text{val}(G', \mathcal{X}')$ which minimizes $|\bigcup \mathcal{X}'|$.

Let us call $\{(v, i) : i \in [b(v)]\}$ a **bunch**. A subset of V' is called **symmetric** if it is the union of bunches. Next we want to prove the following:

(65) Every set in \mathcal{X} is symmetric.

If, for contradiction, (65) does not hold, then we may assume without loss of generality that at least one of the following cases 1–4 holds.

Case 1: $(v, 1) \in X'_0$ and $(v, 2) \notin \bigcup \mathcal{X}'$. Then we define $\mathcal{X}'' := \{X'_0 - (v, 1); X'_1, \dots, X'_k\}$. Let $G'(\mathcal{X}') := G' - X'_0 - \bigcup_{i \geq 1} X'_i$, and $G'(\mathcal{X}'') := G'' - X''_0 - \bigcup_{i \geq 1} X''_i$. Since all the neighbors of $(v, 1)$ are neighbors of $(v, 2)$ as well, we get that the components of $G'(\mathcal{X}'')$ are the same as the components of $G'(\mathcal{X}')$ – except that $(v, 1)$ is added to the component containing $(v, 2)$. Moreover, observe that $\bigcup \mathcal{X}'' = \bigcup \mathcal{X}' - (v, 1)$. These observations imply $val(G', \mathcal{X}'') < val(G', \mathcal{X}')$, a contradiction.

Case 2: $(v, 1) \in X'_0$ and $(v, 2) \in X'_1$. Let $K \in \mathcal{K}(\mathcal{X}')$ be the component with $(v, 2) \in V(K)$. Firstly, suppose that $V(K) = \{(v, 2)\}$, i.e. $(v, 2)$ is an isolated node of $G'(\mathcal{X}')$. Then we define $\mathcal{X}'' := \{X'_0 - (v, 1); X'_1 + (v, 1), \dots, X'_k\}$. It is easy to see that $G'(\mathcal{X}'')$ is obtained by adding $(v, 2)$ as an isolated node. This implies $val(G', \mathcal{X}'') < val(G', \mathcal{X}')$ immediately, a contradiction. Secondly, suppose $V(K) \neq \{(v, 2)\}$, that is $(v, 2)$ has a neighbor in $V' - X'_1 - X'_0$. Then that neighbor is a neighbor of $(v, 1)$ as well. We define $\mathcal{X}'' := \{X'_0 - (v, 1); X'_1, \dots, X'_k\}$. Then $G'(\mathcal{X}'')$ is obtained by adding $(v, 1)$ to the component containing $(v, 2)$. Since $(v, 1) \notin \mathcal{X}''$, we get that $val(G', \mathcal{X}'') < val(G', \mathcal{X}')$. A contradiction.

Case 3: $(v, 1) \in X'_1$ and $(v, 2) \notin \bigcup \mathcal{X}'$. Then we define $\mathcal{X}'' := \{X'_0; X'_1 - (v, 1), \dots, X'_k\}$. Since all the neighbors of $(v, 1)$ are neighbors of $(v, 2)$ as well, we get that the components of $G'(\mathcal{X}'')$ are the same as the components of $G'(\mathcal{X}')$. Moreover, observe that $\mathcal{X}'' \subseteq \mathcal{X}'$. We get $val(G', \mathcal{X}'') \leq val(G', \mathcal{X}')$, which is a contradiction.

Case 4: $(v, 1) \in X'_1$ and $(v, 2) \in X'_2$. Then we define $\mathcal{X}'' := \{X'_0; X'_1 - (v, 1), \dots, X'_k\}$. I.e. $X''_i := X'_i$ for $i \neq 1$ and $X''_1 := X'_1 - (v, 1)$. Let $\mathcal{X}' := \bigcup_{i \geq 1} X'_i$ and $\mathcal{X}'' := \bigcup_{i \geq 1} X''_i$. We make the following case distinction. Firstly, suppose there is a component $K \in G'(\mathcal{X}')$ such that $(v, 1), (v, 2) \in V(K)$. Then all the neighbors of $(v, 1)$ and $(v, 2)$ are in $V(K) \cup X'_0$. This implies that the family of components of $G'(\mathcal{X}')$ and $G'(\mathcal{X}'')$ is the same – i.e. $\{V(L) : L \in \mathcal{K}(\mathcal{X}')\} = \{V(L) : L \in \mathcal{K}(\mathcal{X}'')\}$. Since $\mathcal{X}'' \subseteq \mathcal{X}'$, we immediately get that $val(G', \mathcal{X}'') \leq val(G', \mathcal{X}')$, a contradiction. Secondly, suppose there are two distinct components $K_1, K_2 \in G'(\mathcal{X}')$, $K_1 \neq K_2$ such that $(v, 1) \in V(K_1)$ and $(v, 2) \in V(K_2)$. We claim that $\Gamma_{G'}((v, 1)) \subseteq X'_1 \cup X'_2 \cup X'_0$. To see this: If $(v, 1)$ had a neighbor in $V' - (X'_1 \cup X'_2 \cup X'_0)$, then that would be a neighbor of $(v, 2)$ as well. Thus $(v, 1)$ and $(v, 2)$ were in the same component of $G'(\mathcal{X}')$, a contradiction. Thus we get that $\Gamma_{G'}((v, 1)) \subseteq X'_1 \cup X'_2 \cup X'_0$ holds

indeed. This implies that either the family of components of $G'(\mathcal{X}')$ and $G'(\mathcal{X}'')$ is the same – that is $\{V(L) : L \in \mathcal{K}(\mathcal{X}'')\} = \{V(L) : L \in \mathcal{K}(\mathcal{X}')\}$ – or we get $G'(\mathcal{X}'')$ by merging the two components K_1, K_2 – that is $\{V(L) : L \in \mathcal{K}(\mathcal{X}'')\} = \{V(L) : L \in \mathcal{K}(\mathcal{X}')\} - \{V(K_1), V(K_2)\} + \{V(K_1) \cup V(K_2)\}$. In the former case, since $X'' \subseteq X'$, we get that $val(G', \mathcal{X}'') \leq val(G', \mathcal{X}')$, a contradiction. In the latter case, notice that $val(G', \mathcal{X}') - val(G', \mathcal{X}'') = \lfloor |\mathcal{X}' \cap V(K_1)|/2 \rfloor + \lfloor |\mathcal{X}' \cap V(K_2)|/2 \rfloor - \lfloor |\mathcal{X}'' \cap V(K_1) \cup V(K_2)|/2 \rfloor$. Observe that $(v, 1) \in (\mathcal{X}' \cap V(K_1)) \cup (\mathcal{X}' \cap V(K_2)) - (\mathcal{X}'' \cap V(K_1) \cup V(K_2))$. Hence we are done by the following elementary inequality $\lfloor (q + r - 1)/2 \rfloor \leq \lfloor q/2 \rfloor + \lfloor r/2 \rfloor$ (for positive integers q, r).

This completes the proof of (65).

We define $X_i := \{v \in V : (v, 1) \in X'_i\}$ for all $i = 0, 1, \dots, k$. By the construction of G' it is easy to see that $G'(\mathcal{X}')$ consists of symmetric components, and bunches of isolated nodes. It is easy to see that $G(\mathcal{X})$ consists of the pre-images of the symmetric components, and isolated nodes which are the pre-images of the bunches of isolated nodes in $G'(\mathcal{X}')$. This observation, and (65) implies that $val_b(G, \mathcal{X}) = val(G', \mathcal{X}')$. \square

Weakly-polynomial time algorithm for node-capacitated packing

A weakly-polynomial time algorithm for b -packing follows, by applying a packing algorithm in the expanded graph. (“Weakly-” means that the running time is bounded by an expression that is polynomial in the size of the graph and the value of b .) Since the size of the expanded graph is polynomial in $|V| + |E| + b(V)$, we obtain the following theorem.

Theorem 3.3 *Given G, \mathcal{A}, b as above, one can find a maximum b -packing and an optimum \mathcal{A} -partition in time polynomial in $|V| + |E| + b(V)$.*

3.2 Node-capacitated fractional packing of \mathcal{A} -paths

The motivation for fractional packing is on one hand, just to extend our knowledge of tractable problems, and on the other hand, it will be useful to construct a semi-strongly polynomial time algorithm for integral packing, too. (“Semi-” means that, in addition to basic arithmetics, we may also round-down rational numbers. This is necessary to detect the parity of an integer, which is inevitable to find an optimum integral packing.) Our method dealing with fractional packing is a polyhedral description, which is shown to admit half-integral primal and dual optima. The dual system admits a strongly polynomial time separation oracle via shortest path algorithms, which via ellipsoid method implies the evaluation of the optimum value.

Above we have already shown that there is a weakly-polynomial time algorithm for node-capacitated packing, but the question remains whether there is a semi-strongly polynomial time algorithm, as well. We follow a principle of Gerards [35], who constructed a semi-strongly polynomial time algorithm for b -matching in graphs. Gerards’ algorithm first solves the fractional relaxation of b -matching by a reduction to network flow, and then cites a “proximity” theorem asserting that there is a maximum integral b -matching within a close neighborhood of any maximum fractional b -matching. To apply Gerards’ method to node-capacitated packing of \mathcal{A} -paths, we have to get acquainted with its fractional relaxation, which is our main goal in the following considerations.

Consider the special case of node-capacitated packing when all node-capacities are even, that is, we want to determine $\nu_{2b}(G, \mathcal{A})$ for some $b \in \mathbb{N}^V$. It turns out that there is a slightly simpler characterization for this special case, than without the assumption of even capacities. We will see that the “parity constraint” in the min-max formula – which is manifested by taking the integer part of a half-integer number – can be dropped, and still we get a tight min-max result. We will see that the maximum cardinality of a $2b$ -packing is in fact equal to the maximum of the fractional relaxation of the problem. Thus – by taking halves – we will get a half-integrality result saying that for an arbitrary b , the maximum value of a fractional packing is attained by a half-integral packing. This half-integrality result generalizes the well-known half-integrality result on b -matching in graphs.

A solution of the $2b$ -packing problem is given by the following Theorem 3.4. A vector $w \in \{0, 1, 2\}^V$ is called a **cover** if $w(V(P)) \geq 2$ holds for all \mathcal{A} -paths P . The upper bound $\nu_{2b}(G, \mathcal{A}) \leq bw$ is straightforward. The Theorem says that the minimum of these upper bounds is tight, thus providing a min-max formula for $2b$ -packing. We observe that, as opposed to formula (64), the dual in this min-max formula does not use parity. This suggests that the $2b$ -packing problem might be easier in some sense than the b -packing problem in general.

Theorem 3.4 (Mader’s Theorem with even node-capacities) *Consider an undirected graph $G = (V, E)$, and let $\mathcal{A} := \{A_1, \dots, A_k\}$ be a partition of $A \subseteq V$, and let $b : V \rightarrow \mathbb{N}$ be an even vector. Then*

$$(66) \quad \nu_{2b}(G, \mathcal{A}) = \min bw,$$

where the minimum is taken over covers w .

First proof – reduction to Mader’s Theorem

Theorem 3.2 implies that the maximum cardinality of a $2b$ -packing is determined by

$$\begin{aligned} \nu_{2b}(G, \mathcal{A}) &= \min \text{val}_{2b}(G, \mathcal{X}) = \min \left(2b(X_0) + \sum_{K \in \mathcal{K}_0} \left\lfloor \frac{2b(V(K) \cap X)}{2} \right\rfloor \right) = \\ &= \min (2b(X_0) + b(V(\bigcup \mathcal{K}_0) \cap X)), \end{aligned}$$

where the minimum is taken over \mathcal{A} -partitions \mathcal{X} . Consider a minimizing \mathcal{A} -partition \mathcal{X} . We define a vector $w \in \mathbb{R}^V$ by $w(v) := 2$ for $v \in X_0$, $w(v) := 1$ for $v \in V(\bigcup \mathcal{K}_0) \cap X$, and $w(v) := 0$ otherwise. w is a cover, since any \mathcal{A} -path traverses a node in X_0 or at least two nodes in $V(\bigcup \mathcal{K}_0) \cap X$.

Second proof – half-integrality of a linear programming description

In this section we consider the linear programming relaxation of b -packings, and prove that the primal and dual optima are attained at half-integer solutions. These half-integrality results of Theorem 3.7, Theorem 3.5 imply – in fact are equivalent with – Theorem 3.4. We will prove these half-integrality results directly, and thus we provide an alternative proof of Theorem 3.4, not using Mader’s Theorem.

A vector x : “family of all \mathcal{A} -paths” $\rightarrow \mathbb{R}$ is called a **fractional b -packing** if it fulfills the conditions (68)-(69). The fractional packing problem is the following.

$$(67) \quad \text{maximize } \mathbf{1}x \text{ subject to (68)-(69)}$$

$$(68) \quad x(P) \geq 0 \quad \text{for all } \mathcal{A}\text{-paths } P$$

$$(69) \quad x(\{P : v \in V(P)\}) \leq b(v) \quad \text{for all } v \in V$$

The linear programming dual of (67)-(69) is given by the system (70)-(72). Feasible solutions w of (71)-(72) are called **fractional covers**.

$$(70) \quad \text{minimize } bw \text{ subject to (71)-(72)}$$

$$(71) \quad w(v) \geq 0 \quad \text{for all } v \in V$$

$$(72) \quad w(V(P)) \geq 1 \quad \text{for all } \mathcal{A}\text{-paths } P$$

We remark that the polytope (71)-(72) of covers is “easy” in the sense that a separation algorithm is given by $\binom{k}{2}$ shortest path algorithms. Linear programming duality implies that the maximum in (67) is equal to the minimum in (70). Let $\nu^*(G, \mathcal{A}, b)$ denote the optimum value of the linear program, which is equal to the maximum size $\mathbf{1} \cdot x$ of a fraction packing x , and equal to the minimum value $b \cdot w$ of a cover w . Let $\nu^*(G, \mathcal{A}) := \nu^*(G, \mathcal{A}, 1)$. To prove Theorem 3.4 we will prove that these optima are attained at half-integral vectors x and w , which is stated in the following theorems.

Theorem 3.5 *There is a minimum fractional cover which is half-integer.*

Proof. (This proof was given in Vazirani [85].) Consider an arbitrary optimum cover w . Let W_i denote the set of nodes which are reachable from A_i by a path on nodes of w -weight 0, including end-points. We define sets $Q_i := (A_i - W_i) \cup \Gamma(W_i)$, which are called “borders”. Any \mathcal{A} -path crosses at least two borders, hence $w'(v) := \min\{1, \sum \chi_{Q_i}/2\}$ defines a cover. Moreover, notice, nodes v in any border have $w(v) > 0$. Consider a maximum fractional b -packing x of \mathcal{A} -paths. The following slackness conditions hold.

$$(73) \quad \text{If } w(v) > 0, \text{ then } x(\{P : v \in V(P)\}) = b(v).$$

$$(74) \quad \text{If } x(P) > 0, \text{ then } w(V(P)) = 1.$$

We prove that both of these slackness conditions also hold with w' in place of w . $w(v) > 0$ holds for any $v \in Q_i$, hence (73) also holds with w' in place of w . For (74), consider an \mathcal{A} -path P with $x(P) > 0$. By (74), $w(V(P)) = 1$. Suppose for contradiction that $w'(V(P)) > 1$. Then either

$$(75) \quad \text{there are two distinct nodes } v_1, v_2 \in V(P) \text{ with } w'(v_1) \geq \frac{1}{2} \text{ and } w'(v_2) = 1,$$

or

$$(76) \quad \text{there are three distinct nodes } v_1, v_2, v_3 \in V(P) \text{ with } w'(v_1) = w'(v_2) = w'(v_3) = \frac{1}{2}.$$

In case (75), let P_1 and P_2 be the disjoint paths in P starting in some terminal and ending in v_1 and v_2 , respectively. v_1 is in a border, hence $w(v_1) > 0$. v_2 is in the

intersection of two distinct borders, hence $w(v_1) \geq 1$ holds for any j . This gives $w(V(P)) > 1$, a contradiction. In case (76), suppose v_1, v_2, v_3 appear on path P in this order. v_2 is in some border, say R is a path starting in some terminal, ending in v , and weight $w(V(R) - v) = 0$. $w(v_1), w(v_3) > 0$, hence $v_1, v_3 \notin R$. Thus $P \cup R$ contains an \mathcal{A} -path R' disjoint from v_1 or v_3 . Then $w(R') < w(V(P \cup R)) = 1$, a contradiction. \square

We will prove the half-integrality result on fractional packings (Theorem 3.7) using three kinds of reduction principles. We show Claim 3.6 on these reduction principles, which will also be useful in the proof of Theorem 3.9. Our first reduction is the deletion of an edge of G . It is easy to see that, for any edge $uv \in E$, the inequality $\nu^*(G - uv, \mathcal{A}) \leq \nu^*(G, \mathcal{A})$ holds. We call the edge uv “**superfluous**” if this inequality holds with equality. Our second reduction is the deletion of the two end-points of an edge connecting two distinct terminal sets. Consider an edge $uv \in E$ with $u \in A_i, v \in A_j, i \neq j$. For simplicity, let us denote by \mathcal{A} also its image in $G - u - v$, which is in fact given by $\mathcal{A} - \{A_i, A_j\} + \{A_i - u, A_j - v\}$. Then the inequality $\nu^*(G - u - v, \mathcal{A}) = \nu^*(G, \mathcal{A}) - 1$ holds. We call uv “**solid**” if this inequality holds with equality. Our third reduction is the contraction of an edge of G . Note that we restrict to the contraction of edges having at least one non-terminal end-point. Consider an edge $uv \in E$ such that $u \in V - A$. Let $q := \{u, v\}$ denote the contracted node in the contracted graph G/uv . For simplicity, let us denote by \mathcal{A} denote the sub-partition in G/uv obtained as the image of sub-partition \mathcal{A} – i.e. replace v by q , if necessary. It is easy to see that $\nu^*(G/uv, \mathcal{A}) \leq \nu^*(G, \mathcal{A})$ holds. We call the edge uv “**contractible**” if this inequality holds with equality.

Claim 3.6 *The following assertions holds.*

- (77) *If $u \in A, v \in V - A, uv \in E$ and $d_G(u) = 1$, then uv is contractible.*
- (78) *If $u \in A_i, v \in A_j, i \neq j, uv \in E$ and $d_G(u) = 1$, then uv is solid.*
- (79) *If $u \in A$ and $d_G(u) \geq 3$, then there is a superfluous edge incident with u .*
- (80) *If $u \in V - A$ and $d_G(u) \geq 5$, then there is a superfluous edge incident with u .*
- (81) *If $u \in V - A$ and $1 \leq d_G(u) \leq 3$, then there is a contractible edge incident with u .*

Proof. Assertion (77) follows from the observations that the contraction does not increase the maximum fractional packing, and does not decrease the minimum fractional cover.

Assertion (78) can be proved as follows. If w is a fractional cover of $G - u - v, \mathcal{A}$, then $w + \chi_v$ is a fractional cover of G, \mathcal{A} . Moreover, a fractional packing in $G - u - v, \mathcal{A}$

can be extended by the \mathcal{A} -path uv . This implies $\nu^*(G - u - v, \mathcal{A}) = \nu^*(G, \mathcal{A}) - 1$, and we are done.

Assertion (79) can be proved as follows. Suppose $uz_i \in E$, for $i = 1, 2, 3$, are three distinct edges. Clearly, $\nu^*(G - uz_i, \mathcal{A}) \leq \nu^*(G, \mathcal{A})$ for $i = 1, 2, 3$. We claim that at least one of these three inequalities holds with equality. By Theorem 3.5, all these optima are half-integers, hence we suppose for contradiction that $\nu^*(G - uz_i, \mathcal{A}) \leq \nu^*(G, \mathcal{A}) - \frac{1}{2}$ holds for all $i = 1, 2, 3$. Let w'_i denote a minimum fractional cover with respect to $G - uz_i, \mathcal{A}$. Then $w := \frac{1}{3}(w_1 + w_2 + w_3 + \chi_u)$ is a fractional cover with respect to G, \mathcal{A} . $\mathbf{1} \cdot w \leq \nu^*(G, \mathcal{A}) - \frac{1}{2} + \frac{1}{3}$, a contradiction.

Assertion (80) can be proved as follows. Suppose $uz_i \in E$, for $i = 1, 2, 3, 4, 5$, are five distinct edges. Clearly, $\nu^*(G - uz_i, \mathcal{A}) \leq \nu^*(G, \mathcal{A})$ for $i = 1, 2, 3, 4, 5$. We claim that at least one of these three inequalities holds with equality. By Theorem 3.5, all these optima are half-integers, hence we suppose for contradiction that $\nu^*(G - uz_i, \mathcal{A}) \leq \nu^*(G, \mathcal{A}) - \frac{1}{2}$ holds for all $i = 1, 2, 3, 4, 5$. Let w'_i denote a minimum fractional cover with respect to $G - uz_i, \mathcal{A}$. Then $w := \frac{1}{5}(w_1 + w_2 + w_3 + w_4 + w_5 + 2\chi_u)$ is a fractional cover with respect to G, \mathcal{A} . $\mathbf{1} \cdot w \leq \nu^*(G, \mathcal{A}) - \frac{1}{2} + \frac{2}{5}$, a contradiction.

We prove assertion (81) by distinguishing the following two cases. If $d_G(u) = 1$ or 2 , then consider an arbitrary edge $uv \in E$. Clearly, the contraction of uv does not increase the maximum fractional packing, and does not decrease the minimum fractional cover. We are done. If $d_G(u) = 3$, then suppose $uz_i \in E$, for $i = 1, 2, 3$, are three distinct edges. The inequalities $\nu^*(G/uz_i, \mathcal{A}) \leq \nu^*(G, \mathcal{A})$ are easy to see. We claim that at least one of these three inequalities holds with equality. Suppose for contradiction that $\nu^*(G/uz_i, \mathcal{A}) \leq \nu^*(G, \mathcal{A}) - \frac{1}{2}$ for all $i = 1, 2, 3$. Let w'_i denote a minimum fractional cover with respect to $G/uz_i, \mathcal{A}$. We define $w''_i(a) := w'_i(a)$ for $a \in V - u - z_i$, $w''_i(z_i) := w'_i(q)$ (where q is the contracted node), and $w''_i(u) := 0$. Then for any \mathcal{A} -path P traversing uz_i , we get $w''_i(V(P)) \geq 1$. Since $d_G(u) = 3$, this implies that $w := \frac{1}{3}(w_1 + w_2 + w_3 + \chi_u)$ is a fractional cover with respect to G, \mathcal{A} . $\mathbf{1} \cdot w \leq \nu^*(G, \mathcal{A}) - \frac{1}{2} + \frac{1}{3}$, a contradiction. \square

Theorem 3.7 *There is a maximum fractional b -packing which is half-integer.*

Proof. We split nodes $v \in V$ into $b(v)$ copies, and obtain the expanded graph G_b . Clearly, $\nu^*(G_b, \mathcal{A}) = \nu_b^*(G, \mathcal{A})$. Thus, it suffices to prove the half-integrality result for $b \equiv \mathbf{1}$. We assume without loss of generality that all nodes have positive degree. Theorem 3.5 implies that $\nu^*(G, \mathcal{A})$ is half-integer number. We have to prove that this is attained at a half-integer ($\mathbf{1}$ -)packing. We prove this by induction on $|V| + |E|$. Notice that our reductions defined above can be used in an inductive manner. Thus, by Claim 3.6, we may assume that none of those reductions can be

performed, which implies that all the nodes in A have degree 2, and all the nodes in $V - A$ have degree 4.

Suppose there is a non-terminal $v \in V - A$ of degree 4. Say $vz_i \in E$, for $i = 1, 2, 3, 4$, are its four incident edges. Let x be a maximum fractional packing, and let us decompose x into $x = x_0 + x_1 + x_2 + x_3 + x_4$, where x_i ($i \geq 1$) consists of those components of x which correspond to paths traversing the edge vz_i . Clearly, $\mathbf{1} \cdot (x_1 + x_2 + x_3 + x_4) \leq 2$. If $\mathbf{1} \cdot x_i < \frac{1}{2}$ for some $i \geq 1$, then $\nu^*(G - vz_i, \mathcal{A}) > \nu^*(G, \mathcal{A}) - \frac{1}{2}$. Hence $\nu^*(G - vz_i, \mathcal{A}) = \nu^*(G, \mathcal{A})$, and we are done by induction. Henceforth we may assume that $\mathbf{1} \cdot x_i = \frac{1}{2}$ for all $i \geq 1$. Now let $\mathcal{P}_i := \{P : x_i(P) > 0\}$, and suppose paths $P \in \mathcal{P}_i$ are oriented such that z_i, v appear in this order. Suppose that there are paths $P, R \in \mathcal{P}_1$ such that P, R do not start in the same A_i , or do not end in the same terminal. Say $x_1(P), x_1(R) > \varepsilon > 0$. Notice that $P \cup R$ contains an \mathcal{A} -path Q disjoint from vz_1 . Then $x' := x - x_1 + \varepsilon\chi_Q$ is a fractional packing, and $\nu^*(G - vz_1, \mathcal{A}) \geq \mathbf{1} \cdot x' > \nu^*(G, \mathcal{A}) - \frac{1}{2}$. Hence $\nu^*(G - vz_1, \mathcal{A}) = \nu^*(G, \mathcal{A})$, and we are done by induction. Otherwise the following property holds.

(82) The support of x_i is a flow of value $\frac{1}{2}$ from some A_j to some A_m .

Since all the nodes in $V - A$ have degree 4, we may assume that this property holds for every node in $V - A$ and all $i \geq 1$. Since all the nodes in A have degree 2, we can decompose the respective flows into a half-integral combination of \mathcal{A} -paths, which provides a half-integral maximum fractional packing. \square

The half-integrality results imply that the maximum size of a fractional packing is exactly half the maximum size of a 2-packing. Thus we obtain $\nu(G, \mathcal{A}) = \frac{1}{2}\nu_2(G, \mathcal{A})$. Let us introduce the following notation.

$$\text{val}^*(G, \mathcal{X}) := \frac{1}{2}\text{val}_2(G, \mathcal{X}) = |X_0| + \sum_{K \in \mathcal{K}_0} \frac{1}{2}|V(K) \cap X|.$$

Then it is easy to see the following formula.

Corollary 3.8 $\nu^*(G, \mathcal{A}) = \min \text{val}^*(G, \mathcal{X})$.

Special half-integral packings

We prove a strengthening of Theorem 3.7 for $b \equiv 1$, asserting the existence of a maximum half-integral packing of a “special” kind, defined below. The point is that paths in the support of a half-integral packing may be terribly entangled, and difficult to go on proving other results. Hence we show that the optimum is attained at an esthetic half-integral packing, and this will be useful in the following section, relating fractional packing to matroid fractional matching.

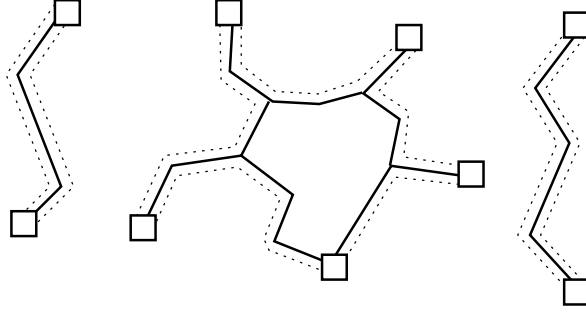


Figure 9: A special half-integral packing.

Next we define what we mean about “special” packings, demonstrated in Figure 9, where the squares are terminals, and a dotted line stands for a path of multiplicity one-half. Suppose C is a cycle, and $P_1, P_2, \dots, P_{2m+1}$ are fully node-disjoint s_i-t_i paths such that $t_i \in V(C)$. Suppose t_i appear in this cyclic order on cycle C , but note that C may traverse other nodes, as well. Let $Q_1, Q_2, \dots, Q_{2m+1}$ denote the unique s_i-s_{i+1} path in $P_i \cup P_{i+1} \cup C$ which is disjoint from all other paths P_j . These paths Q_i are called **induced paths**. If every induced path Q_i is an \mathcal{A} -path, then the configuration C, P_1, \dots, P_{2m+1} is called an **odd \mathcal{A} -cycle**. Given a fully node-disjoint family of \mathcal{A} -paths and odd \mathcal{A} -cycles, a half-integral packing x is defined by assigning $x(P) := 1$ with all the paths in the family, and assigning $x(Q_i) := \frac{1}{2}$ with all induced paths of all odd \mathcal{A} -cycles in the family. Half-integral packings x arising in this manner are called **special**.

Theorem 3.9 *There is a special maximum half-integral packing.*

Proof. We prove this theorem by induction on $|V| + |E|$. By Theorem 3.7 there is a half-integral maximum packing x . If there was an edge which is not traversed by any path P with $x(P) > 0$, then we are done by deleting this edge and applying the inductive hypothesis. Hence we assume that E is the union of paths of positive x -value. This implies that the degree of terminals is 1 or 2, and the degree of non-terminals is 2,3, or 4. Suppose there is a terminal u of degree 1, and its unique neighbour v is a non-terminal. Then $\nu^*(G, \mathcal{A}) = \nu^*(G/uv, \mathcal{A})$, and we are done by induction. Suppose there is a terminal u of degree 1, and its unique neighbor v is another terminal. Then $\nu^*(G, \mathcal{A}) = \nu^*(G - u - v, \mathcal{A}) + 1$, and we are done by induction. Suppose there is a non-terminal of degree 2. Then we are done by applying the inductive hypothesis after the contraction of one of its incident edges. Suppose there is a non-terminal u of degree 3. Then x contains two paths P, R with $x(P), x(R) = \frac{1}{2}$ which traverse u . Moreover, let us denote by uv_1, uv_2, uv_3 are the three edges incident with u such that $uv_1, uv_2 \in E(P)$ and $uv_1, uv_3 \in E(R)$. Then we are done by applying the inductive hypothesis for G/uv_1 .

Thus we may assume all the following conditions:

(83) The degree of terminals is 2, the degree of non-terminals is 4, and

(84) E is the union of all the paths P with $x(P) > 0$, and

Let \mathcal{P} denote the family of all paths P with $x(P) > 0$. The degree-assumption implies that these paths P are edge-disjoint, and $x(P) = \frac{1}{2}$. We prove that in addition to (83) and (84), we may assume the following:

(85) For any non-terminal $v \in V - A$, the two paths of \mathcal{P} traversing v are connecting the same pair A_i, A_j of terminal sets.

Consider a node $v \in V - A$, and the two paths $P, R \in \mathcal{P}$. Say P connects A_i to A_j , and R connects A_q to A_r . Say $vz_1, vz_2 \in E(P)$ and $vz_3, vz_4 \in E(R)$. Suppose $q \notin \{i, j\}$. Then $P \cup R$ contains an A_i - A_q -path R' and an A_j - A_q -path R'' . It is easy to see that $x' := x - \frac{1}{2}(\chi_P + \chi_R) + \frac{1}{4}(\chi_P + \chi_{R'} + \chi_{R''})$ is a fractional packing such that $\mathbf{1} \cdot x' = \mathbf{1} \cdot x - \frac{1}{4}$. This implies that $\nu^*(G - vz_4) = \nu^*(G)$, hence we are done by induction. Thus we may assume (85), as well.

Henceforth, assume that G, \mathcal{A} satisfies conditions (83)–(85). Consider a node $v \in V - A$, and the four-tuple e, f, g, h of edges incident with v . Then there are two paths $P_1, P_2 \in \mathcal{P}$ traversing v , say connecting terminals in A_i to terminals in A_j . Then these edges may be partitioned into **pairs** $\{e, f\}$ and $\{g, h\}$ such that both P_1 and P_2 traverse exactly one of the edges from both pairs, and $\{e, f\}$ appear first when orienting these paths from A_i towards A_j . Given a node $v \in A$, the two edges $\{e, f\}$ incident with v are considered a **pair**. Thus, every edge $e \in E$ has exactly two pairs, namely its pairs with respect to its two endpoints. Now, consider the following system of inequalities on vectors $x \in \mathbb{R}^E$.

$$(86) \quad x \geq 0$$

$$(87) \quad x_e + x_f = 1 \quad \text{for } v \in A, \delta_G(v) = \{e, f\}$$

$$(88) \quad x_e + x_f = x_g + x_h \leq 1 \quad \text{for } v \in V - A, \delta_G(v) \text{ consists of pairs } \\ \{e, f\} \text{ and } \{g, h\}$$

Note that $x \equiv \frac{1}{2}$ is a solution of this system. Moreover, (83)–(85) implies that any solution of this system may be easily transformed into a maximum fractional packing. Thus it suffices to prove the following:

(89) If $V - A \neq \emptyset$, then (86)–(88) has a half-integer solution such that $x \not\equiv \frac{1}{2}$.

Suppose for contradiction that $v \in V - A$, and $\delta_G(v)$ consists of pairs $\{e, f\}$ and $\{g, h\}$. Let $e_1 := e$, and let e_2 be its pair different from f . For $i = 1, 2, \dots$, let e_{i+2} be e_{i+1} 's pair different from e_i . Thus we obtain a cycle e_1, e_2, \dots, e_m on pairwise distinct nodes, where $e_m = f$.

If $m = 2n$, then the following definition provides a half-integral solution of (86)–(88): let $x(e_{2i}) := 1$ and $x(e_{2i-1}) := 0$ for $i = 1, 2, \dots, n$ and let $x(e) := \frac{1}{2}$ otherwise.

If $m = 2n + 1$, then first suppose that $\{g, h\} \subseteq \{e_1, e_2, \dots, e_m\}$. By symmetry we may assume that $e_{2j} = g$ and $e_{2j+1} = h$. Then the following definition provides a half-integral solution of (86)–(88): $x(e_{2i}) := 1$ and $x(e_{2i-1}) := 0$ for $i = 1, 2, \dots, j$ and let $x(e) := \frac{1}{2}$ otherwise.

Finally suppose that $m = 2n + 1$ and $g, h \notin \{e_1, e_2, \dots, e_m\}$. For $g_1 := g$, let $g_1, g_2, \dots, g_{m'}$ denote the sequence defined similarly such that $g_{m'} = h$. By symmetry, we may assume that $m' = 2n' + 1$. In this case, the following definition provides a half-integral solution of (86)–(88): let $x(e) := 1$ for $e \in \{e_2, e_4, \dots, e_{2n}\} \cup \{g_2, g_4, \dots, g_{2n'}\}$, let $x(e) := 0$ for $e \in \{e_1, e_3, \dots, e_{2n+1}\} \cup \{g_1, g_3, \dots, g_{2n'+1}\}$, and let $x(e) := \frac{1}{2}$ otherwise. \square

3.3 Fractional packing of \mathcal{A} -paths reduces to matroid fractional matching

We show that the problem of fractionally packing \mathcal{A} -paths is equivalent with an instance of fractional matching in a linearly representable matroid. Here we will use a construction given by Schrijver [77], which he used as a linear matroid matching representation of (integral) packing \mathcal{A} -paths. We remark that Lovász' represented this problem by matroid matching in a different matroid, but fractional matching in that matroid does not seem to be related we fractional packings. Hence it is important that we use Schrijver's linear representation, and this implies a fractional packing algorithm by exploiting a fractional matching algorithm in this linear matroid.

Consider a graph $G = (V, E)$, a family $\mathcal{A} = \{A_1, \dots, A_k\}$ of disjoint subsets of V , and let $A := \bigcup \mathcal{A}$. For $i = 1, \dots, k$, let l_i denote distinct 1-dimensional subspaces of \mathbb{R}^2 . Let \mathcal{Z} denote the $2|V|$ -dimensional vectorspace of functions $V \rightarrow \mathbb{R}^2$. We assign to every edge $e = ab \in E$ a subspace $L_e < \mathcal{Z}$ such that

$$L_e := \{x \in \mathcal{Z} : x(a) = x(b), \text{ and } x(c) = \mathbf{0} \text{ for all } c \in V - a - b\}.$$

We define a subspace by

$$Q := \{x \in \mathcal{Z} : x(a) \in l_i \text{ for all } a \in A_i, \text{ and } x(c) = \mathbf{0} \text{ for all } c \in V - A\}.$$

We define $\mathcal{E} := \{L_e/Q : e \in E\}$, which is – assuming w.l.o.g. that A_i are stable sets – a set of lines in vectorspace \mathcal{Z}/Q .

Recall that $\nu(G, \mathcal{A})$ denotes the maximum cardinality of a packing of \mathcal{A} -paths in G , and $\nu(\mathcal{E})$ denotes the maximum cardinality of a matching of lines \mathcal{E} . Schrijver proved the following relation between these optima, which is useful to find a maximum packing of \mathcal{A} -paths by applying a linear matroid matching algorithm.

Theorem 3.10 (Schrijver, [77]) *If G is connected, then $\nu(G, \mathcal{A}) = \nu(\mathcal{E}) - |V - A|$.*

Recall that $\nu^*(G, \mathcal{A})$ denotes the maximum size of a fractional packing of \mathcal{A} -paths in G , and $\nu^*(\mathcal{E})$ denotes the maximum size of a fractional of a matching of lines \mathcal{E} . We will prove the following relation between these optima, and conclude that fractional packing of \mathcal{A} -paths reduces to a linearly representable instance of matroid fractional matching. We remark that – incorporating methods from Orlin, Vande Vate [63] and Chang, Llewellyn, Vande Vate [7, 8] – one can construct a polynomial time algorithm to find a maximum fractional matching in a set of lines in a full linear matroid. By the following theorem, this implies a(nother) polynomial time algorithm for fractional packing of \mathcal{A} -paths.

Theorem 3.11 *If G is connected, then $\nu^*(G, \mathcal{A}) = \nu^*(\mathcal{E}) - |V - A|$.*

Proof. Let $\mathcal{X} = \{X_0; X_1, \dots, X_k\}$ an optimum \mathcal{A} -partition such that $\text{val}^*(G, \mathcal{X}) = \nu^*(G, \mathcal{A})$. Then we define

$$K := \{x \in \mathcal{Z} : x(a) \in l_i \text{ for } a \in X_i, i \geq 1 \text{ and } x(a) = 1 \text{ for } a \in X_0\}/Q.$$

It is straightforward that $\text{val}^*(K) \leq \text{val}^*(G, \mathcal{X})$. This implies that $\nu^*(G, \mathcal{A}) \geq \nu^*(\mathcal{E}) - |V - A|$.

We show the reverse inequality by induction on V . We make use of the following observation.

- (90) Suppose we are given a vector $x \in \{0, \frac{1}{2}, 1\}^{\mathcal{E}}$ in vectorspace \mathcal{V} , and $x(e) = 1$. Then x is a fractional matching with respect to \mathcal{V} if and only if $x|_{\mathcal{E}-e}$ is a fractional matching with respect to \mathcal{V}/e .

Now, consider a maximum special packing, given by a fully node-disjoint family of \mathcal{A} -paths and odd \mathcal{A} -cycles. Now, if there is any node not traversed by this special packing, then apply observation (90) for an edge e incident with this node, and we are done by induction. Moreover, if there is an \mathcal{A} -path P of length more than 1, then apply observation (90) for an edge $e \in E(P)$, and we are done by induction. If in an odd \mathcal{A} -cycle there is a path P_i of length at least one, then apply observation (90) for an edge $e \in E(P_i)$, and we are done by induction. If in an odd \mathcal{A} -cycle, the cycle C traverses a non-terminal node, then apply observation (90) for an edge $e \in E(C)$ incident with a non-terminal, and we are done by induction. Hence we may assume that $V = A$, and in that case, $\nu^*(G, \mathcal{A}) \leq \nu^*(\mathcal{E}) - |V - A|$ is straightforward. \square

3.4 A semi-strongly polynomial time algorithm for node-capacitated packing of \mathcal{A} -paths

The min-max formula for integral and fractional b -packing suggests that there should be a polynomial time algorithm to construct the optimum. We will in fact construct a semi-strongly polynomial time algorithm, that is, an algorithm of time complexity polynomial in the size of the graph, where any arithmetic operation, or the rounding-down of a rational is considered a single step. Following the common definition, a strongly polynomial time algorithm may only use arithmetic operations. The construction of a strongly polynomial time algorithm for integer b -packing is not possible, since the optimum might depend on the parity of entries of b , which in general is inaccessible through arithmetic operations only. Thus, the most one should hope for is a semi-strongly polynomial time algorithm.

The idea behind the algorithm dates back to B. Gerards' [35] strongly polynomial time algorithm for b -matching in undirected graphs, which is based on a so-called "Proximity Lemma". A sketch of our b -packing algorithm is the following. First, by exploiting the equivalence of separation and optimization for the polytope of fractional covers, we construct a maximum fractional b -packing. (The maximum fractional packing is given by a decomposition to $\binom{k}{2}$ flows between the possible pairs A_i-A_j of terminal sets. This flow-decomposition is useful to avoid dealing with the exponential groundset of all possible A -paths. In the end we convert these flows into the linear combination of a polynomial number of paths. Thus, we obtain an optimum fractional b -packing the support of which consists of at most $|E|\binom{k}{2}$ paths.) Second, we take the integer part of this maximum fractional b -packing, and thus obtain a near-maximum integral b -packing. Taking the integer part, we lose less than 1 on every path in the packing, thus the gap between this integral b -packing and an optimum b -packing is less than $|E|\binom{k}{2}$. Third, the goal now is to close this gap by iteratively augmenting the b -packing as long as we reach an optimum, and then exhibit a certificate of optimality. At this point we will exploit the analogue of Gerards' "Proximity Lemma", claiming that, given an arbitrary sub-optimal b -packing, there is a larger b -packing in its close surrounding. Closeness here is measured in terms of l_1 -distance, thus the Proximity Lemma implies that we only have to consider a polynomial size residual capacity to exchange the near-optimum b -packing for an optimum one.

We will use the following theorem based on the ellipsoid method. Using simultaneous Diophantine approximation, Frank, Tardos [28] proved the strongly polynomial extension of a theorem of Grötschel, Lovász, Schrijver [37] on separation and optimization over polyhedra.

Theorem 3.12 *Suppose we are given a polynomial time solvable class of polyhedra.*

Then there is strongly polynomial time algorithm for separation if and only if there is a strongly polynomial time algorithm for optimization.

Fractional packing in strongly polynomial time

A fractional packing of \mathcal{A} -paths may be decomposed into $\binom{k}{2}$ flows in an auxiliary digraph, connecting pairs of distinct terminal sets. This allows us to represent fractional packing by a solvable system of inequalities on $\binom{k}{2}2|E|$ variables. A rigorous discussion of this representation is the following.

Let $D = (V, \overline{E})$ denote the auxiliary digraph obtained from G by replacing edges by a pair of back and forth arcs. (Hence $|\overline{E}| = 2|E|$.) We introduce $\binom{k}{2}$ variables for all arcs, that is, $f = (f_1, \dots, f_k)$, where $f_{ij} \in \mathbb{R}^{\overline{E}}$, $1 \leq i < j \leq k$. We consider the set of solutions x of system (91)-(94), that is, f_{ij} are A_i - A_j -flows satisfying the node-capacities in total.

$$(91) \quad f \geq 0$$

$$(92) \quad f_{ij}(\delta_D(u)) - f_{ij}(\varrho_D(u)) = 0 \quad \text{for } 1 \leq i < j \leq k, u \notin A_i \cup A_j.$$

$$(93) \quad f_{ij}(\delta_D(u)) = 0 \quad \text{for } u \in A_k, k \neq i, j.$$

$$(94) \quad f(\varrho_D(u)) \leq b(u) \quad \text{for } u \in V.$$

A straightforward observation is that $\sum_{u \in A} f(\delta_D(u))$ is equal to the size of the corresponding fractional b -packing. Thus, by Theorem 3.12, we can determine a maximum size fractional b -packing, given by its decomposition into $\binom{k}{2}$ flows. Then, these flows may be decomposed into flows along single paths, and thus we obtain a maximum fractional b -packing x such that $|\text{supp}(x)| \leq \binom{k}{2}|E|$.

Integral packing in semi-strongly polynomial time

We construct a maximum b -packing as follows. First, we construct a maximum fractional b -packing using the linear programming formulation above. Let x denote the lower integer part of this fractional optimum. Thus, the size of x is near to the maximum size of a b -packing: the gap is smaller than the cardinality of its support, which is at most $\binom{k}{2}|E|$. To close this gap, we could apply a packing algorithm to augment x , and optimality will be achieved after performing at most $\binom{k}{2}|E|$ augmentations. To show that one augmentation may be performed in strongly polynomial time, we have to reduce the problem of augmenting x to a smaller augmentation problem. For this, we show a ‘‘proximity theorem’’ Theorem 3.13, implying that in the search for an augmentation we may restrict to a close neighborhood of x .

For nodes $v_1, v_2, v_3, v_4 \in V$, let $\mathcal{P}_{v_1, v_2, v_3, v_4}$ denote the family of those \mathcal{A} -paths traversing each of these four nodes, and having its ends in A_i, A_j . For a packing x with respect to G, \mathcal{A} , let b_x denote the smallest capacities enough to support x – i.e. $b_x(v) := x(\{P : v \in V(P)\})$. We claim that a b -packing x' may be constructed in strongly polynomial time fulfilling the following conditions.

$$(95) \quad x' \leq x.$$

$$(96) \quad x'(\mathcal{P}_{v_1, v_2, v_3, v_4}) \geq \min\{2, x(\mathcal{P}_{v_1, v_2, v_3, v_4})\} \text{ for quadruples } v_1, v_2, v_3, v_4 \in V.$$

$$(97) \quad 1 \cdot x' \leq 2|V|^4.$$

Indeed, we first initialize $x' := 0$. If some x' does not fulfill (96) for some quadruple, then we increase x' by one or two, such that we maintain (95). Here we take all the quadruples only once into account, which implies (97). For $v \in V$ we define

$$(98) \quad b'(v) := \min\{b_{x'}(v) + 2, b(v) + b_{x'}(v) - b_x(v)\}.$$

Notice that, if y' is a b' -packing, then $y := x - x' + y'$ is a b -packing. Thus, if there is a b' -packing larger than x' , then there is a b -packing larger than y , as well. The following theorem claims that the converse holds, too.

Theorem 3.13 (Proximity of b -packings) *If x' is a maximum b' -packing, then x is a maximum b -packing. Moreover, in this case, an optimum \mathcal{A} -partition with respect to b', y' is optimum with respect to b, y , as well.*

Proof. Suppose x' is a maximum b' -packing. Then by Mader's Theorem there is an \mathcal{A} -partition \mathcal{X} such that $1 \cdot x' = \text{val}_{b'}(G, \mathcal{X})$. Suppose for contradiction that $1 \cdot x < \text{val}_b(G, \mathcal{X})$. Considering “slackness conditions” with respect to \mathcal{X} and x , it is easy to see that at least one of the following assertions holds.

- There is a node $v \in X_0$ such that $b(v) - 1 \geq b_x(v)$.
- There is a component $K \in \mathcal{K}_0(\mathcal{X})$ and a node $v \in V(K)$ s. t. $b(v) - 2 \geq b_x(v)$.
- There is a component $K \in \mathcal{K}_0(\mathcal{X})$ and two distinct nodes $v, v' \in V(K) \cap X$ such that $b(v) - 1 \geq b_x(v)$ and $b(v') - 1 \geq b_x(v')$.
- There is a component $K \in \mathcal{K}_0(\mathcal{X})$ and a path P such that $|V(K) \cap V(P) \cap X| \geq 4$, and $x(P) \geq 1$.
- There is a component $K \in \mathcal{K}_0(\mathcal{X})$ and two distinct paths P, P' such that $|V(P) \cap X|, |V(P') \cap X| \in \{1, 3\}$, and $x(P), x(P') \geq 1$.
- There is a path P such that $|V(P) \cap X_0| \geq 2$, and $x(P) \geq 1$.

- There is a path P , and two distinct components $K, K' \in \mathcal{K}_0(\mathcal{X})$ such that $|V(P) \cap V(K) \cap X|, |V(P) \cap V(K') \cap X| \geq 2$, and $x(P), x(P') \geq 1$.
- There is a path P , a node $v \in X_0$, and a component $K \in \mathcal{K}_0(\mathcal{X})$ such that $v \in V(P)$, $|V(P) \cap V(K) \cap X| \geq 2$, and $x(P) \geq 1$.

It is easy to see that each of these assertions may be determined from the value of $\min\{2, b_x(v) - b(v)\}$ for some nodes $v \in V$, and from the value of $\min\{2, x(\mathcal{P}_{v_1, v_2, v_3, v_4})\}$ for some triples of nodes $v_1, v_2, v_3, v_4 \in V$. Notice that, by (98), $\min\{2, b_{x'}(v) - b(v)\} = \min\{2, b_x(v) - b(v)\}$, and moreover, by (97), $\min\{2, x'(\mathcal{P}_{v_1, v_2, v_3, v_4})\} = \min\{2, x(\mathcal{P}_{v_1, v_2, v_3, v_4})\}$. Thus we conclude that at least one of the above assertions holds for b', x' , in place of b, x . A contradiction. \square

Corollary 3.14 *A maximum b -packing and an optimum \mathcal{A} -partition can be constructed in strongly polynomial time.*

Proof. By applying the linear programming formulation above, one can construct a maximum fractional b -packing such that its support has cardinality at most $\binom{k}{2}|E|$. Let x be the lower integer part, and let x' be constructed as above, fulfilling (96)-(97). Now apply the weakly-polynomial time subroutine to decide whether x' is a maximum b' -packing, or not. If we find a larger b' -packing y' , then $x - x' + y'$ is a larger b -packing; then apply this subroutine for $x - x' + y'$ iteratively. If we find out that x' is a maximum b' -packing, then by Theorem 3.13, x is a maximum b -packing, and an optimum \mathcal{A} -partition follows. The algorithm has polynomial running time, since we have to apply the subroutine at most $k \binom{|E|}{2}$ times, and one iteration has polynomial running time, since $b'(V) \leq 2|V|^4$. \square

3.5 Mader matroids are gammoids

In this section we will use some notation, and results from the previous sections. Recall that $G = (V, E)$ is an undirected graph, $A \subseteq V$ is a set of terminals, and $\mathcal{A} = \{A_1, \dots, A_k\}$ is a partition of A into disjoint terminal sets. For a packing \mathcal{P} of \mathcal{A} -paths, let $A(\mathcal{P})$ (resp. $V(\mathcal{P})$) denote the set of terminals (resp. nodes) covered by \mathcal{P} . A set $Q \subseteq A$ is called **coverable** if there is a packing \mathcal{P} of \mathcal{A} -paths such that $Q \subseteq A(\mathcal{P})$. The family of coverable sets is the family of independent sets of a matroid, as was shown in [77] by Schrijver. This matroid is called the **Mader matroid** for G, \mathcal{A} .

A well-known class of matroids is the class of gammoids, constructed as follows. Consider a digraph $D = (U, B)$ with nodeset U and arcset B . Let S, A be disjoint subsets of U . We say a set $Q \subseteq A$ is **linked to** S if there is a packing of $S \rightarrow Q$ directed paths such that S is the set of starting nodes, and Q is the set of ending nodes of paths in the family. The bases of the gammoid are the subsets of A linked to S .

Schrijver [77, 79] asks the following question, for which we give a positive answer.

Question 3.15 (Schrijver [77, 79]) *Is each Mader matroid a gammoid?*

Theorem 3.16 *Mader matroids are gammoids.*

The construction of the gammoid representation of a Mader matroid is in fact implicitly contained in Mader's original proof [58] of his min-max formula. We reconstruct a similar auxiliary digraph from a specially chosen dual optimum in Mader's formula.

Proof

We introduce some more notation. Consider an \mathcal{A} -partition \mathcal{X} . For a component $K \in \mathcal{K}$, let $A^K := V(K) \cap X$ denote the **border of** K . Thus X is in fact equal to the union of all the borders. Let us call $K \in \mathcal{K}$ **even/odd** depending on the cardinality of its border. We define a partition $\mathcal{A}^K = \{A_1^K, \dots, A_k^K\}$ of the border by $A_i^K := X_i \cap V(K)$.

Recall the following observation: Any \mathcal{A} -path either uses at least one node of X_0 , or uses at least two nodes of the border of some component. This implies that, if \mathcal{P} is a maximum packing of \mathcal{A} -paths and \mathcal{X} is a minimum value \mathcal{A} -partition, then

(99) the nodes in $A - V(\mathcal{P})$ are in the border of odd components, and

(100) if $A^K = \emptyset$ for some $K \in \mathcal{K}$, then $V(\mathcal{P}) \cap V(K) = \emptyset$.

Such optimality criteria are called slackness conditions, we discuss them in more detail in the sequel.

A special optimal \mathcal{A} -partition

Mader's theorem determines the optimum by means of an \mathcal{A} -partition. Sebő and Szegő [80], and independently Chudnovsky et al. [10] gave a structural description, generalizing the Gallai-Edmonds decomposition. Their results imply that there is an optimum \mathcal{A} -partition having some specific properties. For our purposes we need an optimum \mathcal{A} -partition such that K, \mathcal{A}^K is critical (defined below) for each component $K \in \mathcal{K}$. This follows from those structural descriptions mentioned above.

We recall some notation introduced in [80] – analogous notation was introduced in [10], too. A path P is called an A_i - v path if one of its ends is v , its other end is $a \in A_i$, and $\{a\} = V(P) \cap A$. (Here the nodes a and v are not necessarily distinct, i.e. P may be a zero-length path starting and ending in $a = v \in A_i$.) We call a node $v \in V$ **i -rooted** if there is an A_i - v path P such that $\nu(G, \mathcal{A}) = \nu(G - V(P), \mathcal{A})$. A node is called **rooted** if it is i -rooted for some i , and is called **multi-rooted** if it is i -rooted and j -rooted for some $i \neq j$. Notice, a terminal in A_i may not be j -rooted for $j \neq i$, thus no terminal is multi-rooted. The instance G, \mathcal{A} is called **critical** if $|A| = 2\nu(G, \mathcal{A}) + 1$, all the nodes are rooted and all the nodes in $V - A$ are multi-rooted.

Following results of Sebő and Szegő [80], we may assume without loss of generality that there is an optimal \mathcal{A} -partition \mathcal{X} such that

$$(101) \quad X_0 \subseteq V - A.$$

$$(102) \quad K, \mathcal{A}^K \text{ is critical for each component } K \in \mathcal{K}.$$

The gammoid representation of the Mader matroid

Consider an optimal \mathcal{A} -partition \mathcal{X} having properties (101) and (102). Let $\beta_K := |X \cap V(K)| - 1$. By (102), β_K is even. Let $[\beta_K] := \{1, 2, \dots, \beta_K\}$. We define an auxiliary directed graph $D = (U, B)$. See (103), (104) for the formal definition. For the nodeset U , we keep the old nodes of X , and delete all other nodes. For any node $x \in X_0$ we introduce $k + 2$ new nodes $x', x'', (x, 1), \dots, (x, k)$. For any component K we introduce β_K new nodes $(K, 1), \dots, (K, \beta_K)$.

$$(103) \quad U := \{x', x'' : x \in X_0\} \cup \{(K, m) : K \in \mathcal{K}, m \in [\beta_K]\} \cup \\ \cup \{(x, i) : x \in X_0, i \in [k]\} \cup X$$

and arcset is defined by

$$\begin{aligned}
(104) \quad B := & \{x'(x, i), x''(x, i) : x \in X_0, i \in [k]\} \cup \\
& \cup \{(x, i)v : xv \in E, v \in X_i\} \cup \\
& \cup \{(x, i)v : \exists K \in \mathcal{K}, v \in V(K) \cap X, \exists xu \in E, u \in V(K) - X\} \cup \\
& \cup \{(K, m)v : v \in V(K) \cap X, m \in [\beta_K]\} \cup \bigcup_{i=1}^k \overleftarrow{E}[X_i],
\end{aligned}$$

where $\overleftarrow{E}[\cdot]$ is obtained by replacing all edges $uv \in E[\cdot]$ by the two arcs uv, vu . Define

$$S := \{x', x'' : x \in X_0\} \cup \{(K, m) : K \in \mathcal{K}, m \in [\beta_K]\}$$

Notice that $|S| = 2 \text{val}(G, \mathcal{X})$ holds. The following claim completes the proof of Theorem 3.16.

Claim 3.17 *A set $Q \subseteq T$ is a maximum coverable set if and only if it is linked to S in D .*

Proof. To prove this claim we need to formulate slackness conditions in more detail. As already observed, any \mathcal{A} -path uses at least one node of X_0 , or at least two nodes of the border of some component. This observation, and our choice of \mathcal{X} imply that if \mathcal{P} is a maximum packing, and $P \in \mathcal{P}$, then P can be decomposed into segments defined by (105) or (106).

$$(105) \quad P = a_i - P_i - r_i - R - r_j - P_j - a_j, \text{ where } a_i \in A_i, r_i \in X_i \cap V(K), r_j \in X_j \cap V(K), a_j \in A_j, \text{ and these nodes split } P \text{ into segments } P_i, R, P_j \text{ such that } V(P_i) \subseteq X_i, V(R) \subseteq V(K), V(P_j) \subseteq X_j.$$

The notation in (105) implies that $i \neq j$, $r_i \neq r_j$, thus R has positive length. However P_i, P_j may have zero length, so a_i, r_i and a_j, r_j may coincide.

$$(106) \quad P = a_i - P_i - y_i - R_i - z_i - x - z_j - R_j - y_j - P_j - a_j, \text{ where } a_i \in A_i, y_i \in V(K) \cap X_i, z_i \in V(K), x \in X_0, z_i \in V(K'), y_j \in V(K') \cap X_j, a_j \in A_j \text{ for some } K \neq K', K, K' \in \mathcal{K}. \text{ Moreover, these nodes split } P \text{ into segments } P_i, R_i, z_i x, x z_j, R_j, P_j \text{ such that } V(P_i) \subseteq X_i, V(R_i) \subseteq V(K), z_i x, x z_j \in E, V(R_j) \subseteq V(K'), V(P_j) \subseteq X_j.$$

The notation in (106) implies $i \neq j$, $x \neq a_i, y_i, z_i, z_j, y_j, a_j$, and $a_i, y_i, z_i \neq z_j, y_j, a_j$. However, the nodes a_i, y_i, z_i (resp. z_j, y_j, a_j) are not necessarily distinct, and P_i, R_i, R_j, P_j may have zero length. The following hold for any maximum packing \mathcal{P} .

$$(107) \quad \text{For any } K \in \mathcal{K} \text{ there are exactly } \beta_K/2 \text{ paths in } \mathcal{P} \text{ of category (105).}$$

$$(108) \quad \text{Every node } x \in X_0 \text{ is traversed by a path in } \mathcal{P} \text{ of category (106).}$$

Suppose a maximum packing \mathcal{P} such that $Q = A(\mathcal{P})$. We replace paths $P \in \mathcal{P}$ by two directed S - Q paths in digraph D as follows. If P is a category (105) path, then we replace P by the two directed paths $(K, m) \text{--} r_i \text{--} \vec{P}_i \text{--} a_i$ and $(K, m') \text{--} r_j \text{--} \vec{P}_j \text{--} a_j$. (107) implies that we can choose m 's such that all (K, m) 's are used by exactly one such directed path. Now suppose P is of category (106). Then the two paths in exchange for P will be $x' \text{--} (x, i) \text{--} z_i \text{--} \vec{P}_i \text{--} a_i$ and $x'' \text{--} (x, j) \text{--} z_j \text{--} \vec{P}_j \text{--} a_j$. By (108), all then nodes $x', x'' \in S$ are use by exactly one such directed path.

Consider a family \mathcal{R} of directed paths linking Q to S . Consider a directed path $R \in \mathcal{R}$. It follows from the construction of D that there is a unique component K such that R traverses an arc $(K, m)v$ or $(x, i)v$ with $v \in V(K) \cap X$. We say R **enters** K **via** $(K, m)v$, or via $(x, i)v$, respectively. The paths in \mathcal{R} entering K are the following: exactly β_K paths starting in the nodes $\{(K, m) : m \in [\beta_K]\}$, and at most one additional path starting in some x' or x'' . So all but one of the nodes in the border of K are traversed by the paths entering K via some $(K, i)v$. The remaining border node z may be traversed by a path entering K via some $(x, i)v$, or a path not entering K .

Let $R^{(1)}, \dots, R^{(\beta_K)}$ denote the paths entering K via some $(K, i)v$. Let $P^{(i)}$ (for $i \in [\beta_K]$) denote the undirected path in G which we get from $R^{(i)}$ by deleting its starting node (K, m) . If there is a path entering K via some $(x, i)v$, then let $P^{(0)}$ denote the path we get from it by deleting its starting segment $x' \text{--} (x, i) \text{--} v$ (or $x'' \text{--} (x, i) \text{--} v$). Thus we have either β_K or $\beta_K + 1$ paths joining nodes in the border A^K to terminals in A such that nodes in A_i^K are joined to nodes in A_i .

Consider a component for which there are β_K paths joining A^K to A (i.e. $P^{(i)}$ for $i \in [\beta_K]$). Choose a packing \mathcal{P}_K in K, \mathcal{A}^K (i.e. of $\beta_K/2$ paths) such that $A^K(\mathcal{P}_K)$ is equal to the set of starting nodes of the paths $P^{(i)}$. We compose $\beta_K/2$ \mathcal{A} -paths from the union of the paths $\{R^{(1)}, \dots, R^{(\beta_K)}\} + \mathcal{P}_K$. Clearly, this composition consist of \mathcal{A} -paths.

Consider components K for which there are $\beta_K + 1$ paths joining A^K to A (i.e. $P^{(i)}$ for $i \in \{0\} \cup [\beta_K]$). These components come up in pairs K, K' corresponding to x' and x'' , for some $x \in X_0$. Suppose K is entered by the path $x' \text{--} (x, i) \text{--} v \text{--} R$ and K' is entered by the path $x'' \text{--} (x, j) \text{--} v' \text{--} R'$. Let us recall the definition (104) of the arcset B . The arc $(x, i)v \in B$ implies that either

$$(109) \quad xv \in E, v \in X_i, \text{ or}$$

$$(110) \quad \exists xu \in E, v \in V(K) \cap X, u \in V(K) - X \text{ holds.}$$

If arc $(x, i)v$ is of case (109), then consider a maximum packing \mathcal{P}_K in K, \mathcal{A}^K (i.e. of $\beta_K/2$ paths) such that it is disjoint from v . Such a \mathcal{P}_K exists because of the criticality of K, \mathcal{A}^K . Then we can compose a family of disjoint paths from $\{R^{(0)}, \dots, R^{(\beta_K)}\} + \mathcal{P}_K + vx$, and this composition consists of $\beta_K/2$ \mathcal{A} -paths and

one additional A_i-x path.

If both of the arcs $(x, i)v$ and $(x, j)v'$ are of case (109), then merge the additional A_i-x path and the additional A_j-x path. This gives a family of $\beta_K + \beta_{K'} + 1$ disjoint \mathcal{A} -paths. The terminals covered by these paths are those nodes which were linked to S by some paths entering K and K' .

If arc $(x, i)v'$ is of case (109), but $(x, j)v$ is of case (110), then we construct the family of $\beta_K + \beta_{K'} + 1$ disjoint \mathcal{A} -paths as follows. We do the above construction for $(x, i)v'$, thus we get a family of fully node-disjoint paths which consists of $\beta_K/2$ \mathcal{A} -paths and one additional A_i-x path. We do a different construction for $(x, j)v$, which is the following. Since $v \in V(K) - X = V(K) - A^K$, and K, \mathcal{A}^K is critical, v is multi-reachable in K, \mathcal{A}^K . Thus there is a packing $\mathcal{P}_{K'}$ of $\mathcal{A}^{K'}$ -paths such that there is an $A_m^{K'}-u$ path Q disjoint from all the paths in $\mathcal{P}_{K'}$, and $m \neq i$. Then we can compose a packing of $\beta_K + \beta_{K'} + 1$ \mathcal{A} -paths from the union of $\{R^{(0)}, \dots, R^{(\beta_K)}\} + \mathcal{P}_K + Q + ux$. This composition covers the nodes which were linked to S by some paths entering K and K' .

Suppose both of the arcs $(x, i)v$ and $(x, j)v'$ are of case (110). We do the second of the above constructions for both arcs. There it is possible to choose $m \neq m'$ such that x is reached by an additional A_m-x path (through K) and an additional $A_{m'}-x$ path (through K'). From the union of these two constructions we compose a family of \mathcal{A} -paths as desired. \square

3.6 Packing non-returning A -paths

Chudnovsky et al. gave a min-max formula for the maximum number of node-disjoint non-zero A -paths in group-labeled graphs [10], which is a generalization of Mader’s theorem on node-disjoint A -paths [58]. Here we present a further generalization with a shorter proof. The main feature of Theorem 3.18 is that parity is “hidden” inside $\widehat{\nu}$, which is given by an oracle for non-bipartite matching.

W. Mader [58] gave a theorem on packing A -paths, which was re-stated in a more transparent form in two papers, in [80] by A. Sebő and L. Szegő, and independently in [10] by M. Chudnovsky, J. Geelen, B. Gerards, L. Goddyn, M. Lohman and P. Seymour. In [10] a min-max formula is given for the packing of non-zero A -paths in group-labeled graphs, which contains Mader’s theorem as a special case. In this paper we show Theorem 3.18 on non-returning A -paths in “permutation-labeled” graphs, which contains the result on non-zero A -paths. The method of proof in this paper is also related to the short proof of Mader’s theorem given by A. Schrijver [78]. We use an analogue of Berge’s alternating paths’ lemma in the proof of the main theorem of this paper.

Let $G = (V, E)$ be an undirected graph with node-set V , edge-set E and a fixed set $A \subseteq V$ of **terminals**. Suppose every edge is assigned with a reference-orientation. Let Ω be an arbitrary **set of “potentials”** and let $\omega : A \rightarrow \Omega$ define the **potential of origin** for the terminals. Let $\pi : E \rightarrow S(\Omega)$ where $S(\Omega)$ is the set of all permutations of Ω . For an edge $ab = e \in E$, let $\pi(e, a) := \pi(e)$ and $\pi(e, b) := \pi^{-1}(e)$ be the **mapping of potential** on edge ab . A path is called an **A -path** if it joins two distinct nodes of A , not using any other node in A . For an A -path $P = (v_0, e_0, v_1, e_1, \dots, e_{k-1}, v_k)$, let $\pi(P) := \pi(e_0, v_0) \circ \pi(e_1, v_1) \circ \dots \circ \pi(e_{k-1}, v_{k-1})$ define the **mapping of potentials** on P . Let P be called **non-returning** if $\pi(P)(\omega(v_0)) \neq \omega(v_k)$. In other words, an A -path is returning if it maps the potentials of origin onto each other. Notice that an A -path is non-returning if and only if its reverse is non-returning. A family of fully node-disjoint non-returning A -paths will be called a **non-returning family**. Let $\nu(G, A, \omega, \pi)$ denote the maximum cardinality of a non-returning family.

The problem of finding a maximum non-returning family is a slight generalization of the problem of packing non-zero A -paths in group-labeled graphs. To see this, we define the set of potentials as the set of elements of the group, the potential of origin as zero for each terminal, and we define $\pi(e)$ as the multiplication by the group-label on edge e .

¹This section is set to appear as a short note in *Combinatorica*, and has been published as TR-2005-12 in the Technical Reports series of the Egerváry Research Group.

The min-max formula

Consider a graph $G = (V, E)$ with a set $A \subseteq V$ of terminals. Let $\widehat{\nu}(G, A)$ denote the maximum number of fully node-disjoint A -paths. This is in fact a special case of packing non-returning A -paths, since it is easy to construct mappings of potentials such that every A -path is non-returning. T. Gallai [33] determined $\widehat{\nu}(G, A)$ by a reduction to non-bipartite matching.

Next we define a notion which we use in the main theorem to determine $\nu(G, A, \omega, \pi)$. Consider a set $F \subseteq E$ of edges. Let $A' := A \cup V(F)$. F is called **A -balanced** if ω can be extended to a function $\omega' : A' \rightarrow \Omega$ such that $\pi(ab)(\omega'(a)) = \omega'(b)$ for each edge $ab = e \in F$. (Or equivalently, each edge in F gives a one-edge returning A' -path with respect to ω' . Notice that an A -path P is returning if and only if $E(P)$ is A -balanced.)

Theorem 3.18 *If G, A, ω, π is given as above then the equation*

$$(111) \quad \nu(G, A, \omega, \pi) = \min \widehat{\nu}(G - F, A \cup V(F))$$

holds, where the minimum is taken over A -balanced edge-sets F .

We observe that in Theorem 3.18 the dual is based on two well separated “ingredients”. Firstly, checking whether F is indeed A -balanced may be done by using a depth-first search. Secondly, determining $\widehat{\nu}(G - F, A \cup V(F))$ reduces to maximum matching in an auxiliary graph, see Gallai [33]. However, we will not refer to details of this reduction or to Gallai’s result.

Let us point out that there is only a formal difference between the right hand side in (111) and those given in [80, 10]. It is quite easy to transform the “dual solutions” into each other by using Gallai’s result. So, when we apply Theorem 3.18 to special cases, we obtain some of the results of [80, 10].

Proof of Theorem 3.18

First we prove the easy inequality, i.e. for an A -balanced edge-set F we show that $\nu(G, A, \omega, \pi) \leq \widehat{\nu}(G - F, A \cup V(F))$. Consider a non-returning A -path P . One can easily see that if $E[P] \subseteq F$ then P would be returning. Hence some section of P must be an $A \cup V(F)$ -path in $G - F$. Given any non-returning family, the family of these sections gives a same-size family of node-disjoint $A \cup V(F)$ -paths in $G - F$.

To show equality in (111) we use the following notion. For a non-returning family \mathcal{P} let $A(\mathcal{P})$ denote the set of terminals covered by the paths in \mathcal{P} . A set $Z \subseteq A$ is called **exactly coverable** if there is a non-returning family \mathcal{P} with $Z = A(\mathcal{P})$.

Lemma 3.19 *If Z is exactly coverable with $|Z| < 2\nu(G, A, \omega, \pi)$ (i.e. it is not maximum), then there is an exactly coverable set $Z + s + t$ with $s, t \notin Z$.*

Proof. We prove this by induction on $|V|$. Let \mathcal{P} be a non-returning family with $Z = A(\mathcal{P})$. Consider a non-returning family \mathcal{R} with $|\mathcal{R}| = |\mathcal{P}| + 1$. If $Z \subseteq A(\mathcal{R})$, then we are done. Otherwise there is a node $r \in Z - A(\mathcal{R})$.

Case I. Suppose that r is covered by a one-edge path rr' in \mathcal{P} . Then $Z' := Z - r - r'$ is exactly coverable for $G - r - r', A - r - r', \omega, \pi$. To see that Z' is not maximal, delete from \mathcal{R} a path incident to r' , if any. So, by induction, there is an exactly coverable set $Z' + s + t \subseteq A - r - r'$. It is easy to see that $Z + s + t$ is exactly coverable for G, A, ω, π .

Case II. Suppose that r is covered by a path with first edge $rq \in E$, $q \in V - A$. Define $\omega' : A - r + q \rightarrow \Omega$ by ω on $A - r$ and by $\omega'(q) := \pi(rq, r)(\omega(r))$. Then $Z' := Z - r + q$ is exactly coverable for $G - r, A - r + q, \omega', \pi$. We claim that Z' is not maximal, which can be seen as follows. If the paths in \mathcal{R} are disjoint from q , then $A(\mathcal{R})$ is exactly coverable for $G - r, A - r + q, \omega', \pi$. Otherwise, if there is a path $R \in \mathcal{R}$ with $q \in V(R)$ joining nodes $q_1, q_2 \in A(\mathcal{R})$, then some $q - q_i$ section of R must be non-returning. Thus $A(\mathcal{R}) - q_{3-i} + q$ is exactly coverable for $G - r, A - r + q, \omega', \pi$. So, by induction, there is an exactly coverable set $Z' + s + t \subseteq A - r + q$. It is easy to see that $Z + s + t$ is exactly coverable for G, A, ω, π . \square

Let $\alpha = \alpha(G, A, \omega, \pi)$ denote the number of terminals where at least one returning A -path starts. Consider a counterexample with $|V| + |E|$ minimal, and then α minimal.

Claim 3.20 *There is a node-disjoint family of $\nu(G, A, \omega, \pi) + 1$ A -paths, $\nu(G, A, \omega, \pi)$ of which are non-returning.*

Proof. In case of $\nu(G, A, \omega, \pi) = \widehat{\nu}(G, A)$ the formula (111) obviously holds with $F = \emptyset$. Otherwise, suppose some returning A -path starts in a terminal $t \in A$. Let $\Omega' := \Omega + \bullet$ for some $\bullet \notin \Omega$. Let π' be identical to π , let \bullet be always mapped onto \bullet , and let us redefine $\omega'(t) := \bullet$. So all paths starting in t will be non-returning, and the status of paths disjoint from t does not change. We define $\alpha' := \alpha(G, A, \omega', \pi')$. Clearly, $\alpha' < \alpha$, hence formula (111) holds for the instance G, A, ω', π' . Consider a minimal A -balanced set F with respect to ω', π' with $\nu(G, A, \omega', \pi') = \widehat{\nu}(G - F, A \cup V(F))$.

We claim that F is A -balanced with respect to ω, π . This follows from $t \notin V(F)$, which can be seen as follows. By the definition of ω', π' , there is no A -path in (V, F) starting in t , thus there is a component C of (V, F) with $\{t\} = V(C) \cap A$. It is easy to see that for $F' := F - E(C)$ we have $\widehat{\nu}(G - F', A \cup V(F')) \leq \widehat{\nu}(G - F, A \cup V(F))$, which by the minimality of F implies that $E(C) = \emptyset$.

So, by the choice of G, A, ω', π' we must have $\nu(G, A, \omega', \pi') > \nu(G, A, \omega, \pi)$. Consider $\nu(G, A, \omega, \pi) + 1$ non-returning A -paths with respect to ω', π' . Only one path incident with t can be returning with respect to ω, π . \square

Consider a family \mathcal{P} of A -paths given by Claim 3.20, and let $P \in \mathcal{P}$ be the returning path. Here $E(P)$ is A -balanced. Let ω' be the $A \cup V(P) \rightarrow \Omega$ function from the definition. Let $G' := G - E(P)$ and $A' := A \cup V(P)$. Since the paths in $\mathcal{P} - P$ are non-returning we have

$$(112) \quad \nu(G', A', \omega', \pi') \geq \nu(G, A, \omega, \pi) = \nu.$$

By the choice of G, A, ω, π , there is an A' -balanced edge set F' with respect to ω', π' , with $\nu(G', A', \omega', \pi') = \widehat{\nu}(G' - F', A' \cup V(F'))$. It is easy to see that $F := F' \cup E(P)$ is A -balanced with respect to ω, π . This gives $\widehat{\nu}(G - F, A \cup V(F)) = \widehat{\nu}(G' - F', A' \cup V(F'))$. If in (112) we have equality, then we are done.

Otherwise, there is a non-returning family \mathcal{P}' in G', A', ω', π' with $|\mathcal{P}'| > \nu$. By Lemma 3.19 we may choose \mathcal{P}' with $A(\mathcal{P}') = A(\mathcal{P} - P) + s + t$ for some s, t . We will get to a contradiction by constructing a $|\mathcal{P}'|$ -size non-returning family. To this end, a path ending in a node in $V(P)$ can be extended by a section of P to get a non-returning A -path. Since \mathcal{P}' covers at most two nodes in $V(P)$ we need at most two such extending sections, and clearly, two extending sections fit into P . This contradiction completes the proof. \square

Equivalent models

In this section we discuss other definitions of allowed A -paths, which ultimately turned out to be equivalent with the above model of packing non-returning A -paths. The subgroup model below is a straightforward generalization of Chudnovsky et al. [10]'s non-zero model, which is obtained by substituting the trivial subgroup. In one of our discussions, Laci Szegő recalled that J.F. Geelen independently investigated the subgroup model, and mentioned that their methods may be extended to the subgroup model, as well.

- **The subgroup problem.** Let $G = (V, E)$ be an undirected graph with a reference orientation. Let $A \subseteq V$ be a fixed set of terminals. Let Γ be a group with a subgroup Γ' . Let $\gamma : E \rightarrow \Gamma$ be the group-labels on edges. Let $\gamma(ab, a) := \gamma(ab)$, $\gamma(ab, b) := \gamma(ab)^{-1}$ – here we use the reference orientation. The weight of an arbitrary walk $W = (v_0, e_0, \dots, e_{k-1}, v_k)$ is defined by $\gamma(W) := \gamma(e_0, v_0) \cdot \gamma(e_1, v_1) \cdots \gamma(e_{k-1}, v_{k-1})$. An A -path is allowed if it is non- Γ' , i.e. its weight is not in the subgroup Γ' . Note that setting $\Gamma' := \{\mathbf{1}\}$

returns the problem of packing non-zero A -paths, introduced by Chudnovsky et al. [10].

- **The axiomatic model.** We denote by \mathcal{F} the family of allowed A -walks, which is subject to the below axioms. Given such a family \mathcal{F} , the problem is to pack A -paths each of which must be a member of \mathcal{F} . We remark that there may be A -walks in \mathcal{F} which are not A -paths, but these walks must be included in order to satisfy the axioms. It is unclear, however, which classes of A -paths can be extended to nice classes of A -walks. A walk is called a half- A -walk if its first node is in A . A path is called a half- A -path if it is a half- A -walk. A family \mathcal{F} of A -walks is called **nice** if it satisfies the following two axioms.

(113) (**Axiom of Symmetry**) If W is a half- A -walk, then $W * W^{-1} \notin \mathcal{F}$.

(114) (**Axiom of Triple Exchange**) If W_1, W_2, W_3 are half- A -walks ending in the same node, and $W_1 * W_2^{-1} \in \mathcal{F}$ then at least one of $W_2 * W_3^{-1}$, $W_3 * W_1^{-1}$ is in \mathcal{F} .

Proof of equivalence. Let us call a A -path **proper** if its internal nodes are non-terminals. We claim that the axioms of symmetry and triple exchange imply the following properties, as well. Here, (115) provides another “symmetry” property, whereas (116) implies that a maximum packing of allowed A -paths is achieved by proper A -paths.

(115) $W \in \mathcal{F}$ implies $W^{-1} \in \mathcal{F}$.

(116) If $W \in \mathcal{F}$, and $v_i \in A$ is an internal node of W , then one of W 's segments at v_i also is in \mathcal{F} .

(These two assertions can be proved as follows. For (115), consider an s - t walk W such that $s, t \in A$. Let W_t denote the zero-length walk starting and ending in t . Apply triple exchange for $W_1 := W$, $W_2 := W_t$, $W_3 := W$. Then $W_1 * W_2^{-1} = W \in \mathcal{F}$, by the symmetry axiom $W_3 * W_1^{-1} = W * W^{-1} \notin \mathcal{F}$. Thus $W_2 * W_3^{-1} = W^{-1} \in \mathcal{F}$. (116) follows from triple exchange by using the zero-length half- A -path starting and ending in v_i .)

Claim 3.21 *The subgroup problem fulfills the axioms of symmetry and triple exchange.*

Proof. If W_1, W_2 are half- A -walks ending in the same node, then $\gamma(W_1 * W_2^{-1}) = \gamma(W_1)\gamma(W_2)^{-1}$. Moreover, $ab^{-1} \in \Gamma', bc^{-1} \in \Gamma' \Rightarrow ca^{-1} \in \Gamma'$. These observations imply triple exchange. Symmetry follows from the unit being in Γ' . \square

Claim 3.22 *The non-returning A -path problem reduces to a subgroup problem.*

Proof. Without loss of generality we can assume that there is a fixed element $\bullet \in \Omega$ such that $\omega \equiv \bullet$. (Otherwise, consider a node $a \in A$. Redefine $\omega'(a) := \bullet$. Relabel $\pi'(ab) := \pi(ab) \circ (\bullet \leftrightarrow \omega(a))$, where $\bullet \leftrightarrow \omega(a)$ denotes switching of \bullet and $\omega(a)$. It is easy to see that for any half- A -path W starting in $a \in A$, $\pi(W)(\omega(a))$ does not change. Hence, the non-returning A -walks will be the same for G, A, ω, π as for G, A, ω', π' .)

Let $\Gamma := S(\Omega)$, $\gamma(e) := \pi(e)$, and $\Gamma' := \text{stab}(\bullet)$, i.e. the subgroup of permutations with \bullet a fixpoint. It is quite easy to see that a walk is non-returning if and only if its γ -weight is not in Γ' . \square

Claim 3.23 *The axiomatic problem reduces to non-returning A -paths.*

Proof. We define a relation for half A -walks: half- A -walks W_1, W_2 ending in the same node $t \in V$ we put $W_1 \sim W_2$ if $W_1 * W_2^{-1} \notin \mathcal{F}$. We claim that \sim is an equivalence relation: transitivity follows from triple exchange, reflexivity follows from symmetry. Let Ω be the classes of equivalent half- A -walks with respect to \sim . Let $[W] \in \Omega$ denote the class of half- A -walk W . We define π by

$$\pi(uv)([W]) := \begin{cases} [W] & \text{if } W \text{ ends not in } u \\ [W * uv] & \text{if } W \text{ ends in } u \end{cases}$$

Let $\omega(a) := [(a)]$. We need to show that $\pi(uv)$ is a permutation of Ω .

1. First we show that $\pi(uv)$ is a well-defined function. Consider a half- A -walk W ending in u . $W * uv * vu * W^{-1} \notin \mathcal{F}$ follows from symmetry. Suppose $W \sim W'$. By triple exchange for $W, W', W * uv * vu$ we get $W * uv \sim W' * uv$. Thus $\pi(uv)([W]) = \pi(uv)([W'])$.

2. We show that $\pi(uv)$ is surjective. Consider a half- A -walk W ending in v . $W * vu * uv \sim W$ follows from symmetry. Thus $\pi(uv)([W * vu]) = [W]$.

3. We show that $\pi(uv)$ is injective. Consider two half- A -walks W_1, W_2 ending in u . Suppose $\pi(uv)([W_1]) = \pi(uv)([W_2])$. Then $W_1 * uv * vu * W_2^{-1} \notin \mathcal{F}$. By symmetry we get $W_1 * uv * vu * W_1^{-1} \notin \mathcal{F}$. By triple exchange for $W_1, W_1 * uv * vu, W_2$ we get that $W_1 * W_2^{-1} \notin \mathcal{F}$. Thus $[W_1] = [W_2]$.

The proof will be completed by showing that W is non-returning in the p-graph if and only if $W \in \mathcal{F}$. Say W is a A -walk starting in a and ending in b . By definition we have $\pi(W)(\omega(a)) = [W]$ and $\omega(b) = [(b)]$. So W is non-returning $\Leftrightarrow [W] \neq [(b)] \Leftrightarrow W * (b)^{-1} = W \in \mathcal{F}$. \square

Special cases

- A list of special cases of the non-zero A -paths is given in [10], and these are special cases of non-returning A -paths, of course. Odd length A -paths, S – T paths, Mader’s \mathcal{A} -paths, paths on surfaces which are non-contractible, or non-separating, or orientation-reversing (with respect to a fixed base-point and some paths joining the terminals to the base-point).
- The following problem is a special case of non-returning A -paths, but does not seem to be a special case of non-zero A -paths. Given a graph $G = (V, E)$, a set $A \subseteq V$, a coloring of A , and a partial coloring of the edgeset. So an edge may be assigned one of our colors, but may as well be left uncolored. Allowed A -paths are those traversing at least two distinct colors. I.e. forbidden A -paths have same-color ends in A , and only traverse nodes and edges of this color, or uncolored ones.
- If G, A is fixed, then the union of nice families of A -walks gives a nice family.

The delta-matroid

We observe that a delta-matroid is induced by path-packing, if the family of allowed paths obeys the weak triple exchange axiom. A A -path is called **proper**, if its ends are in A and its internal nodes are in $V - A$. A half- A -path is called **proper**, if only its first node is in A . Fix a family \mathcal{H} of proper A -paths – paths in \mathcal{H} are considered as allowed. Let us call a subset A' of A **exactly coverable** if there is a packing of allowed paths in s.t. A' is exactly the set of terminals covered by the packing. A family \mathcal{H} of proper A -paths is subject to the weak triple exchange axiom.

Weak Triple Exchange Axiom. Consider proper half- A -paths P_1, P_2, P_3 ending in the same node $v \in V - A$ such that $V(P_i) \cap V(P_j) = \{v\}$ (for $i \neq j$). If $P_1 * P_2^{-1} \in \mathcal{H}$ then at least one of $P_2 * P_3^{-1}, P_3 * P_1^{-1}$ is in \mathcal{H} .

We can also interpret the axiom of symmetry with respect to a family \mathcal{H} of proper A -paths. That is, we do not make a distinction between a path and its reverse.

Theorem 3.24 *Suppose a family \mathcal{H} of proper A -paths fulfills the axioms of symmetry and weak triple exchange. Then the family of exactly coverable sets is a delta-matroid.*

Proof. Suppose A_1, A_2 are exactly coverable, say by packings $\mathcal{P}_1, \mathcal{P}_2$. To show the delta-matroid exchange axiom, consider a node $a \in A_1 - A_2$. We claim that there

is a node a' s.t. $A_2\Delta\{a, a'\}$ is exactly coverable. For this, consider the path $P \in \mathcal{P}_1$ starting in a . If $V(P)$ is disjoint from $V(\mathcal{P}_2)$, then we choose a' to be the end of P – we are done. Otherwise, suppose q is the first node of P meeting $V(\mathcal{P}_2)$. Say $q \in V(Q)$, $Q \in \mathcal{P}_2$. Let P_0 denote the a - q segment of P . If $q \in A$, then let a' be the other end of Q – we are done. If $q \in V - A$, then say Q', Q'' are the segments of Q separated by q . The axiom implies that at least one of $Q' * P_0^{-1}$, $Q'' * P_0^{-1}$ is allowed, say for example $Q' * P_0^{-1} \in \mathcal{H}$. We choose a' to be the terminal in Q' – we are done. \square

Open Question

Does the Weak Triple Exchange Axiom imply that the maximum cardinality of a packing of allowed A -path is equal to $\min \widehat{\nu}(G - F, A \cup V(F))$, where the minimum is taken over edge-sets F not containing allowed A -paths?

3.7 Packing non-returning A -paths algorithmically

In this section we construct a polynomial time algorithm for packing non-returning A -paths. For technical reasons in the algorithm, it will be useful to consider an extension of the problem, which is in fact equivalent with packing non-returning A -paths. The definition of this A -path packing problem is almost the same as the definition of packing non-returning A -paths. The difference is that we assign “jolly joker” permutations with some edges, which means that every A -path traversing that edge is non-returning. The precise definition is the following.

Let Ω be an arbitrary set. Let \mathbf{jj} and \mathbf{JJ} be “jolly jokers”, i.e. some “imaginary elements” not in Ω . Elements of $\Omega + \mathbf{jj}$ are called **potentials**, and elements of $S(\Omega) + \mathbf{JJ}$ are called **permutations**. For an arbitrary permutation π and an arbitrary potential ω , we define $\mathbf{JJ}^{-1} := \mathbf{JJ} \circ \pi := \pi \circ \mathbf{JJ} := \mathbf{JJ}$ and $\mathbf{JJ}(\omega) := \pi(\mathbf{jj}) := \mathbf{jj}$. Intuitively, this means that jolly jokers may be substituted “arbitrarily”.

Similar to the problem of packing non-returning A -paths, we are given an undirected graph $G = (V, E)$ with a reference orientation, and a fixed set $A \subseteq V$ of terminals. Let $\omega : A \rightarrow \Omega$. Let $\pi : E \rightarrow S(\Omega) + \mathbf{JJ}$. The quadruple G, A, ω, π is called a **permutation-labeled graph**, or **p-graph**, for short. For an edge $e = ab \in E$, let $\pi(e, a) := \pi(E)$ and $\pi(e, b) := \pi(e)^{-1}$. For a path $P = (v_0, e_0, v_1, e_1, \dots, e_{k-1}, v_k)$, let $\pi(P) := \pi(e_0, v_0) \circ \pi(e_1, v_1) \circ \dots \circ \pi(e_{k-1}, v_{k-1})$. An A -path P is called **non-returning** if $\pi(P)(\omega(v_0)) \neq \omega(v_k)$.

Contraction of dragons

A path P is called a **half- A -path** if it starts in a terminal $s \in A$, ends in a node $t \in V$ and $V(P) \cap A = \{s\}$. We say P **ends in t with potential** $\pi(P)(\omega(s))$.

Consider a node $v \in V$ and a potential $\omega_0 \in \Omega \cup \{\mathbf{jj}\}$. We say a node v is **ω_0 -reachable** (or ω_0 is reachable at v), if there is a pair \mathcal{P}, P_v such that P_v is a half- A -path ending in v with ω_0 , and \mathcal{P} is a packing of ν non-returning A -paths each of which is fully node-disjoint from P_v . We say a node is **reachable** if it is ω_0 -reachable for some $\omega_0 \in \Omega \cup \{\mathbf{jj}\}$. v is called **uniquely reachable** if it is ω_0 -reachable only with a single element $\omega_0 \neq \mathbf{jj}$. Otherwise – if v is **jj-reachable** or there are at least two different elements of Ω which are reachable at v , then v is called **multiply reachable**. The definition implies that a reachable terminal is uniquely reachable.

We call a p-graph G a **dragon** if $|A| = 2\nu + 1$ and every node is reachable. A p-graph is called **critical** if it is a dragon such that every non-terminal is multiply reachable. (The notion of criticals is analogue to the notion used in [10]. The notion of dragons should be considered as a weak version of criticality.)

¹This section is based on Tech. Report TR-2005-15 of the Egerváry Research Group.

Let us use the expression **odd cycle** for p-graphs s.t. $G = (V, E)$ is an odd cycle, $A = V$, and all the edges in E give one-edge non-returning A -walks (which are in fact non-returning A -paths except for 1-edge odd cycles). A p-graph with $V = \{a, b\}$, $E = \{ab\}$, $A = \{a\}$ is called a **rod**.

Claim 3.25 *Odd cycles and rods are dragons.* □

A crucial lemma is the following, saying that the min-max formula holds for dragons. This Lemma will only be proved in the end by using our algorithm. So, we will first give a partial proof of the algorithm and this Lemma, which will be completed in Section 3.7. The Lemma is stated at this point to “filter” the concept of the algorithm.

Lemma 3.26 (A dragon has a special dual) *Suppose a p-graph G is a dragon with exactly its nodes in V_1 being uniquely reachable, say $v \in V_1$ is $\omega'(v)$ -reachable. Let $F := \{e \in E[V_1] : e \text{ is } \omega'\text{-balanced}\}$. Then $2\nu = |V_1| - c(G - F, V_1)$.*

The notion “reachability” is in fact motivated by the goal to define the contraction of dragon subgraphs. Consider a set $Z \subseteq V$ such that $G[Z]$ is dragon. We wish to define the contraction of Z in such a way that the following property holds. The contracted p-graph should be defined on the contracted graph G/Z with terminals $A/Z := A - Z + \{Z\}$, and ω_Z, π_Z should be defined in such a way that any A/Z -path P in G/Z is non-returning if and only if it has an **expansion in G** . An expansion of P is a packing \mathcal{P} s.t. $\mathcal{P}/Z = \{P\}$, and moreover $\mathcal{P} \cap G[Z]$ is a packing in $G[Z]$ of size $\nu(G[Z]) = (|A \cap Z| - 1)/2$.

Definition 3.27 (Contraction of a dragon) *Consider a set $Z \subseteq V$ such that $G[Z]$ is dragon. We define the contracted p-graph on G/Z as follows.*

1. Let Z_1 be the uniquely reachable nodes in $G[Z]$, say $a \in Z_1$ is ω_a -reachable.
2. Let $A/Z := A - Z + \{Z\}$. Let $\Omega' := \Omega + \bullet$ for some new element $\bullet \notin \Omega$.
3. Let $\omega_Z(s) := \omega(s)$ for all $s \in A/Z - \{Z\}$, and let $\omega_Z(\{Z\}) := \bullet$.
4. We define $\pi_Z(e)$ by the following case splitting.
 - (a) If e is disjoint from Z , then we define $\pi_Z(e)$ by extending $\pi(e)$ to Ω' by mapping \bullet to \bullet .
 - (b) For an edge ab with $a \in Z_1$, $b \notin Z$ we label its image $\{Z\}b$ such that $\pi_Z(\{Z\}b)(\bullet) = \pi(ab)(\omega_a)$.
 - (c) For an edge ab with $a \in Z - Z_1$, $b \notin Z$ we define let $\pi_Z(\{Z\}b) := \mathbf{JJ}$.

We remark that the contraction of a dragon Z does not have an effect on the part of G disjoint from Z . That is, the p-graph $G[V - Z]$ is the same as $G/Z[V - Z]$. Thus, dragons disjoint from Z remain dragons in the contraction, as well. This implies that the contraction of a node-disjoint family \mathcal{Z} of dragons does not depend on the order in which those dragons are contracted.

In the following claims we formulate precisely the property of expansion we wished the contraction of a node-disjoint family of dragons would have. Their proofs are immediate from the definition.

Claim 3.28 (Expansion of a path) *Consider a non-returning A/\mathcal{Z} -path P in G/\mathcal{Z} . Then P has an **expansion** in G , i.e. there is a packing \mathcal{P} in G with the following properties. $\mathcal{P}/\mathcal{Z} = P$, and the paths in \mathcal{P} cover all the terminals in the pre-image of P . \square*

Claim 3.29 (Expansion of a packing) *From any packing in G/\mathcal{Z} one can construct a packing in G which exposes the same number of terminals. \square*

Claim 3.30 (Pre-image of a dragon) *Consider a dragon Z_1 in G/\mathcal{Z} . Then the pre-image of Z_1 is dragon. Moreover, $\mathcal{Z}/Z_1 := \{Z : Z \in \mathcal{Z}, \{Z\} \notin Z_1\} \cup \{\text{the pre-image of } Z_1\}$, is a finer node-disjoint family of dragons. \square*

Sequences of contractions and the 3-Way Lemma

The following definition concerns the sequence of contractions arising in our algorithm, and we pick those properties which are important for the algorithmic purposes. Such a sequence arises from a recursive application of the 3-Way Lemma, which will be discussed later.

Consider a packing \mathcal{P} in G and a dragon Z in G . We say \mathcal{P} is **equipped with Z** if \mathcal{P} consists of some paths disjoint from $V(Z)$ and exactly $\nu(G[Z]) = (|A \cap V(Z)| - 1)/2$ paths inside Z . A **sequence of contractions** is a sequence

$$(117) \quad \begin{aligned} & (\mathcal{Z}_1, G_1, \mathcal{P}_1, \mathcal{R}_1, S_1) \\ & (\mathcal{Z}_2, G_2, \mathcal{P}_2, \mathcal{R}_2, S_2) \\ & \dots \\ & (\mathcal{Z}_m, G_m, \mathcal{P}_m, \mathcal{R}_m, S_m) \\ & (\mathcal{Z}_{m+1}, G_{m+1}, \mathcal{P}_{m+1}) \end{aligned}$$

with $m \geq 0$, and the below properties 1,2,3,4. These are the key properties of a sequence of contractions maintained by our algorithm.

1. $\mathcal{Z}_0 = \emptyset$, and \mathcal{Z}_i is a node-disjoint family of dragons in G . $G_i = (V_i, E_i) := G/\mathcal{Z}_i$.
2. $G_i[S_i]$ is an odd cycle or a rod, where $S_i \subseteq V_i$. \mathcal{R}_i is a packing in G_i which is equipped with S_i .
3. $\mathcal{P}_{i+1} := \mathcal{R}_i/S_i$, $\mathcal{Z}_{i+1} := \mathcal{Z}_i/S_i$ for $i = 1, \dots, m$.
4. Each $\mathcal{P}_i, \mathcal{R}_i$ leaves the same number of terminals uncovered.

Clearly, \mathcal{P}_i is a packing in G_i . Notice that in the $m+1$ 'st line we have triples, in all other lines we have 5-tuples. Notice, the $\bigcup \mathcal{Z}_i$ is laminar. There is a trivial sequence of contractions with $m = 0$ and $\mathcal{Z}_0 = \emptyset$. A vector $c : V \rightarrow \{0, 1, 2\}$ is called a **2-cover** if $c(V(P)) \geq 2$ holds for every non-returning A -path P .

Lemma 3.31 (Constructing a verifying pair) *Suppose we have a sequence of contractions, and a 2-cover c in G_{m+1} with $2|\mathcal{P}_{m+1}| = c(V)$. Then for all i , \mathcal{P}_i is a maximum packing in G_i and one can construct a verifying pair for \mathcal{P}_i .*

The 3-Way Lemma is the crucial tool in our algorithm, which either provides an augmentation, or a certificate of optimality, or provides a rod or an odd cycle to extend the sequence of contractions.

Lemma 3.32 (The 3-way Lemma) *Consider a p -graph with a packing \mathcal{P} . Then at least one of the following alternatives holds:*

1. *There is a packing \mathcal{R} with $|\mathcal{R}| = |\mathcal{P}| + 1$.*
2. *There is a packing \mathcal{R} s.t. $|\mathcal{R}| = |\mathcal{P}|$, and is equipped with a rod or an odd cycle.*
3. *There is a 2-cover c such that $2|\mathcal{P}| = c(V)$.*

These two lemmas imply a constructive proof of the main result as follows.

The algorithmic proof of Theorem 1.6. We construct an algorithm ‘‘A’’ with the input of a p -graph, and a packing \mathcal{P} . The output is either a larger packing, or a verifying pair for \mathcal{P} . Iterative application of ‘‘A’’ provides a constructive proof of Theorem 1.6.

The algorithm ‘‘A’’ starts off with initiating the trivial sequence of contractions, $m = 0$. In a general step, apply Lemma 3.32 to $G_{m+1}, \mathcal{P}_{m+1}$! If alternative 1 holds, then by Claim 3.29 one can construct a packing in G larger than \mathcal{P} . If alternative 2 holds, then by Claim 3.30 one can construct a longer sequence of contractions. If alternative 3 holds, then by Claim 3.31 \mathcal{P} is maximum, and a verifying pair can be

constructed. The proof of correctness and polynomiality is completed by observing that the length of a sequence of contractions is bounded by $|E| + |V|$. \square

The above proof will only be algorithmic if we can be deal with the following tasks algorithmically.

- We need to maintain unique/multiple reachability of nodes in a dragon. We will show that reachability can be maintained for those dragons which appear in a sequence of contractions. In fact dragons in a sequence of contractions have a special decomposition called the “dragon decomposition”, Lemma 3.26 will in fact be proved by showing that the dragon decomposition can be maintained throughout a sequence of contractions. A polynomial time algorithm is constructed by maintaining a dragon decomposition for dragons in \mathcal{Z}_i .
- To construct an algorithm we also need an oracle to deal with the p-graph, the permutations. It turns out that it is enough to have an oracle to check whether some A -walk is non-returning. Given such an oracle for $G = G_0$, an oracle for G_i can be constructed by calling the G -oracle. We skip this construction.

Proof of the 3-way Lemma

The proof goes by induction on $|V(G)| + |E(G)|$.

First we check whether $c := \chi_{A(\mathcal{P})}$ is a 2-cover. If not, then there exists a non-returning A -walk W starting in $a \in A - A(\mathcal{P})$. If W consists of a single edge e which is disjoint from $V(\mathcal{P})$, then alternative 2 holds if e is a loop, or alternative 1 holds if e is a non-loop edge. If W consists of more than one edge, and its first edge is disjoint from $V(\mathcal{P})$, then clearly, alternative 2 holds.

It remains that there is an edge $ab \in E$ such that

$$(118) \quad a \in A - A(\mathcal{P}), b \in V(\mathcal{P}), \text{ and if } b \in A \text{ then } ab \text{ is a non-returning path.}$$

Say $b \in V(P)$ for some $P \in \mathcal{P}$. If $b \notin A$, then there are 3 A -paths inside $P \cup ab$. By the definition of non-returning A -paths either 2 or 3 of them are non-returning. These are the two cases to be distinguished below. Say edge ab meets path $P \in \mathcal{P}$, b splits P into sections $P = (s_1, P_1, b, P_2, s_2)$

Case 1. $b \in V - A$, $ab + P_1$ is a returning A -path and $ab + P_2$ is a non-returning A -path.

Suppose P_1 has more than one edge, say its first edge is $s_1s'_1$. Then $\mathcal{R} := \mathcal{P} - P + (ab + P_2)$ and rod $s_1s'_1$ solves alternative 2. So, suppose P_1 has exactly one edge, namely s_1b . We define a contracted graph G', A', ω', π' as follows. Let

$G' := (G - ab - s_1b) / \{s_1, a\}$, let p_1 denote the new node, let $A' := A - a - s_1 + b + p_1$ and

$$\omega'(v) := \begin{cases} \omega(v) & \text{for } v \in A' - b - p_1 \\ \pi(ab)(\omega(a)) & \text{for } v = b \\ \bullet & \text{for } v = p_1 \end{cases}$$

and

$$\pi'(uv) := \begin{cases} \pi(uv) & \text{if } u, v \neq p_1 \\ \bullet \mapsto \pi(av)(\omega(a)) & \text{if } uv \text{ is the image of an edge } av \\ \bullet \mapsto \pi(s_1v)(\omega(s_1)) & \text{if } uv \text{ is the image of an edge } s_1v. \end{cases}$$

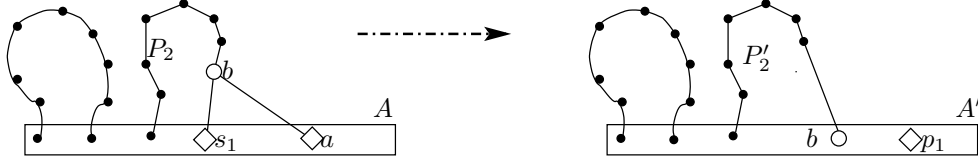


Figure 10: The reduction in **Case 1**.

If alternative 3 holds with c' , then one can easily see that $c'(p_1) = 0$. Hence alternative 3 is solved by $c(u) := c'(u)$ (for $u \notin \{s_1, a\}$) and $c(a) := c(s_1) := 0$. We prove that if one of alternatives 1,2 holds for G' , then we find a solution also for G . Consider \mathcal{R}' , \mathcal{R}' and S' , \mathcal{R}' and $u'v'$ in the respective alternative for G' . We find one of the alternatives for G by considering the pre-image of the solution (i.e. \mathcal{R}' , \mathcal{R}' and C' , \mathcal{R}' and $u'v'$) plus ab, s_1b . To check these we only need to check components incident with p_1, b . So at most two components we need to think about, each of which may be a non-returning A -path in \mathcal{R}' , the odd cycle C , or the edge $u'v'$. Subcases are depicted in Figure 11.

Case 2. $b \in A$.

If P had more than one edge, then $\mathcal{P} - P + ab$ and the first edge of P would solve alternative 2. So, suppose $P = s_1b$. Then $a, b, s_1 \in A$. We define a contracted p-graph G', A', ω', π' . Let $G' := (G - b) / \{s_1, a\}$, let s denote the contracted node $A' := A - a - b - s_1 + s$ and

$$\omega'(v) := \begin{cases} \omega(v) & \text{for } v \in A' - s \\ \bullet & \text{for } v = s \end{cases}$$

and

$$\pi'(uv) := \begin{cases} \pi(uv) & \text{if } u, v \neq s \\ \bullet \mapsto \pi(s_1v)(\omega(s_1)) & \text{if } uv \text{ is the image of an edge } s_1v \\ \bullet \mapsto \pi(av)(\omega(a)) & \text{if } uv \text{ is the image of an edge } av. \end{cases}$$

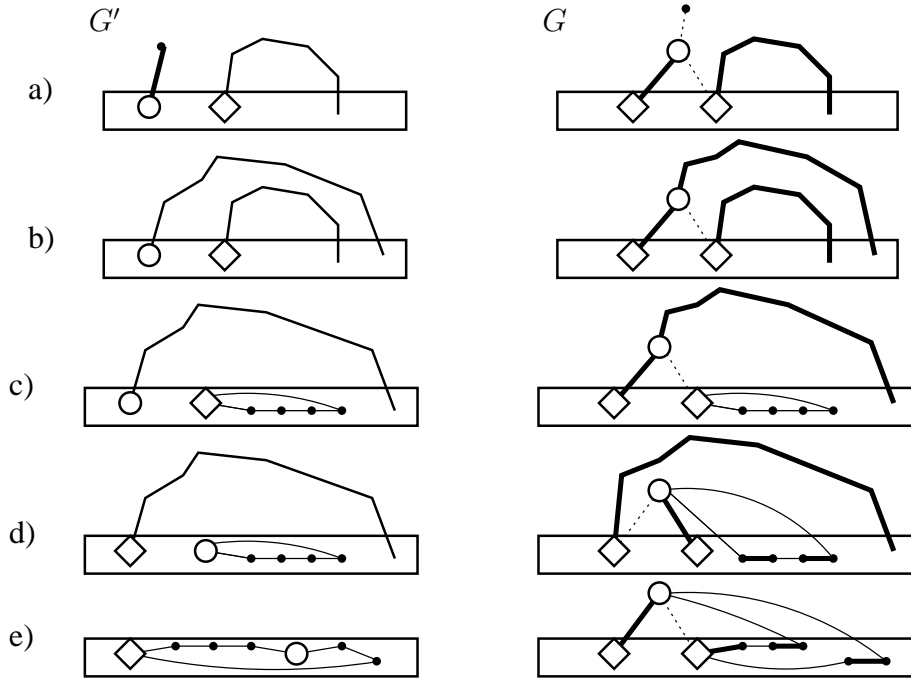


Figure 11: Subcases of Case 1.

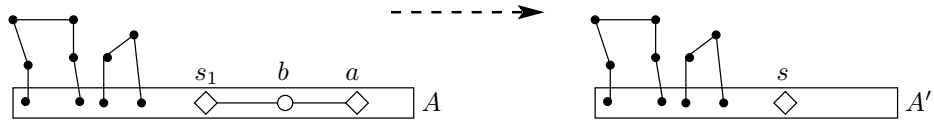


Figure 12: The reduction in Case 2.

If alternative 3 holds with c' , then one can easily see that $c'(s) = 0$. Hence alternative 3 is solved by $c(u) := c'(u)$ (for $u \notin \{a, b, s_1\}$), $c(a) := c(s_1) := 0$ and $c(b) := 2$. We claim that if one of alternatives 1,2 holds for G' , then we find a solution also for G . Subcases are depicted in Figure 13.

Case 3. $b \notin A$ and $ab + P_1$, $ab + P_2$ are both non-returning A -paths.

If P_1 or P_2 has more than one edge, then one can see by a similar argument that alternative 3 holds. So, suppose $P \cup ab$ is a claw with center $b \in V - A$ and tips $a, s_1, s_2 \in A$ such that all 3 of the A -paths in the claw are non-returning. Let $G' := (G - E(P) - ab - b)/(V(P) - b + a)$, let s denote the contracted node $A' := A - s_1 - s_2 - a + s$ and

$$\omega'(v) := \begin{cases} \omega(v) & \text{for } v \in A' - s \\ \bullet & \text{for } v = s \end{cases}$$

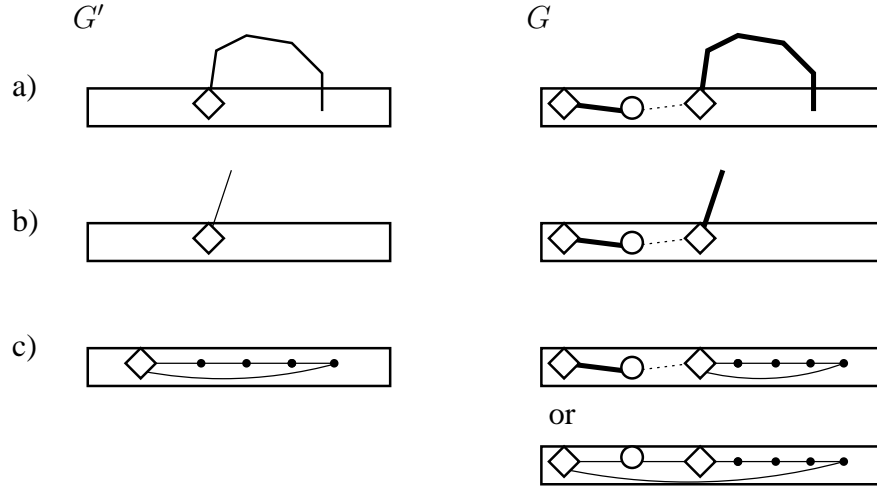


Figure 13: Subcases of Case 2.

and

$$\pi'(uv) := \begin{cases} \pi(uv) & \text{if } u, v \neq s \\ \bullet \mapsto \pi(s_1v)(\omega(s_1)) & \text{if } uv \text{ is the image of an edge } s_1v \\ \bullet \mapsto \pi(s_2v)(\omega(s_2)) & \text{if } uv \text{ is the image of an edge } s_2v \\ \bullet \mapsto \pi(av)(\omega(a)) & \text{if } uv \text{ is the image of an edge } av. \end{cases}$$

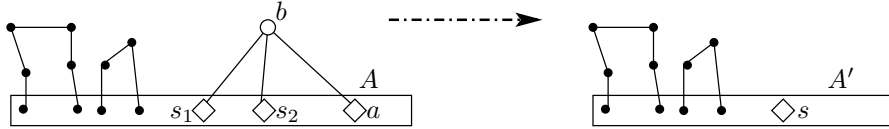


Figure 14: The reduction in **Case 3**.

If alternative 3 holds with c' , then one can easily see that $c'(s) = 0$. Hence alternative 3 is solved by $c(u) := c'(u)$ (for $u \notin \{a, b, s_1, s_2\}$), $c(a) := c(s_1) := c(s_2) := 0$ and $c(b) := 2$. We claim that if one of alternatives 1,2 holds for G' , then we find a solution also for G . Subcases are depicted in Figure 15.

2-covers and verifying pairs

Consider a graph G and a set A of nodes. A node is called an **isolated node of G, A** if it is in A , and it is an isolated node of G . Let $c_1(G, A)$. Thus, $c_{\text{odd}}(G, A) \leq c_1(G, A)$.

Consider a 2-cover c and a packing \mathcal{P} such that $c(V) = 2|\mathcal{P}|$. Let $X := \{v \in V : c(v) = 2\}$, $Y := \{v \in V : c(v) = 1\}$. Let B be the union of components of $G - X - Y$

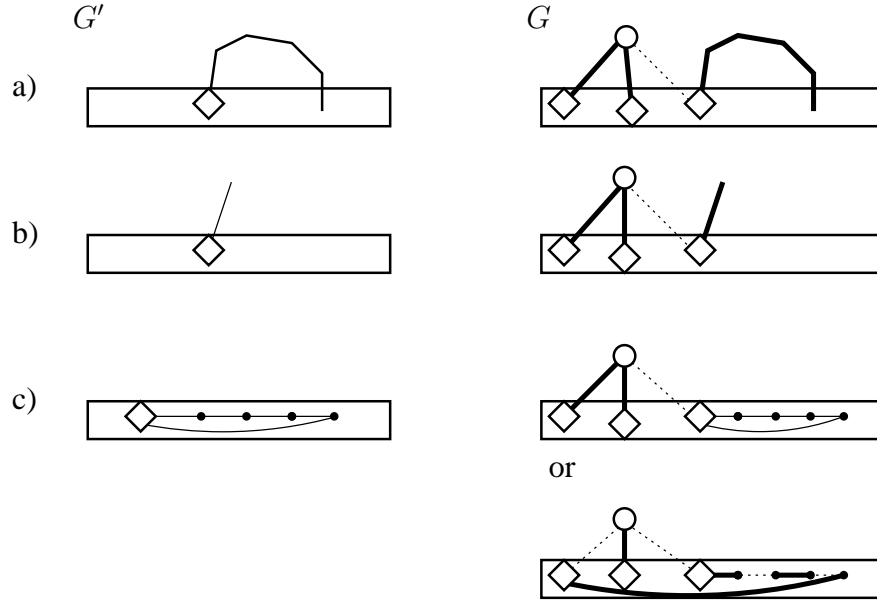


Figure 15: Subcases of Case 3.

which have at least one node in A . Let $F := \{ab \in E[V - X] : a \in B \text{ or } b \in B\}$, which is easily seen to be A -balanced. We claim that $\{v \in A^F - X : v \text{ is not isolated in } G - F - X\} \subseteq Y$. Thus $c(V) = 2|X| + |A^F - X| - c_1(G - F - X, A^F - X)$. We conclude that if there is a 2-cover such that $c(V) = 2|\mathcal{P}|$, then there is a verifying pair X, F such that all odd components are isolated nodes.

Proof of Lemma 3.31

It follows from the previous section that there is a verifying pair X, F_{m+1} for \mathcal{P}_{m+1} such that all odd components are isolated nodes. For $i < m+1$ let $\{S_i\}^{(m+1)} \in V_{m+1}$ denote the image of S_i in G_{m+1} . We will construct a pair F_1, X below, and to prove that it is a verifying pair with respect to \mathcal{P}_1 we are going to use that for all $i' \geq 1$ the following holds:

$$(119) \quad \begin{aligned} \{S_{i'}\}^{(m+1)} &\in A_{m+1}^{F_{m+1}} - X \text{ and} \\ \text{moreover } \{S_{i'}\}^{(m+1)} &\text{ is an isolated node of } G_{m+1} - X - F_{m+1}. \end{aligned}$$

Consider the smallest positive integer i such that for all $i' \geq i$ (119) holds. (If (119) holds for no i' , then we consider $i = m+1$.)

Let $\{Z_{1,m+1}, Z_{2,m+1}, \dots, Z_{q,m+1}\}$ be the family of dragons in \mathcal{Z}_{m+1} which are the images of at least one of $\{\{S_{i'}\}^{(m+1)} : i \leq i'\}$. Let $\mathcal{Z}_{m+1}^{(i)} = \{Z_{1,m+1}^{(i)}, Z_{2,m+1}^{(i)}, \dots, Z_{q,m+1}^{(i)}\}$ denote their pre-images in G_i . We claim that each $G_i[Z_{j,m+1}^{(i)}]$ is a dragon – this follows from Claim 3.30. Let $U_{j,m+1}^{(i)} \subseteq Z_{j,m+1}^{(i)}$ denote the set of uniquely

reachable nodes of $G_i[Z_{j,m+1}^{(i)}]$. Let $U := \bigcup_j U_{j,m+1}^{(i)}$. By definition, there exists a function ω_i'' on U s.t. a node $v \in U$ is uniquely $\omega_i''(v)$ -reachable (in $G_i[U_{j,m+1}^{(i)}]$). Let $F_{ij} := \{uv : \exists j, uv \in E_i[U_{j,m+1}^{(i)}], \text{ and } uv \text{ is } \omega_i''\text{-balanced}\}$

$$(120) \quad F_i := \{\text{pre-image of } F_{m+1} \text{ in } G_i\} \cup \bigcup F_{ij}.$$

Claim 3.33 F_i, X is a verifying pair for \mathcal{P}_i .

Proof. First we show that F_i is A_i -balanced. Consider the set H of nodes in $A_{m+1}^{F_{m+1}} - X$ which are also nodes in V_i – that is, $H = (A_{m+1}^{F_{m+1}} - X) - \{\{S_{i'}\}^{(m+1)} : i \leq i'\}$. To prove that $A_i^{F_i} - X \subseteq U \cup H$, we need to show that there is no edge $ab \in F_i$ with $a \in Z_{j,m+1}^{(i)} - U_{j,m+1}^{(i)} \not\subseteq b$. This follows from the fact that there is no edge in F_{m+1} with a **JJ** label. Hence we have $A_i^{F_i} - X \subseteq U \cup H$, indeed. By definition, there is a function ω'_{m+1} on $A_{m+1}^{F_{m+1}} - X$ s.t. all edges in F_{m+1} are ω'_{m+1} -balanced. Let us define a function ω'_i by the value of ω_i'' on nodes in U , and by the value of ω'_{m+1} on nodes in H . By definition 3.27, all edges in F_i are ω'_i -balanced.

Next we show that the value of F_i, X is equal to $|\mathcal{P}_i|$. By Lemma 3.26 we get

$$(121) \quad |A_i \cap U_{j,m+1}^{(i)}| - 1 = |U_{j,m+1}^{(i)}| - c(G_i[Z_{j,m+1}^{(i)}] - F_{ij}, U_{j,m+1}^{(i)}).$$

It is easy to see that $|A_i^{F_i} - X| \leq |H| + \sum_j |U_{j,m+1}^{(i)}|$, and $|A_{m+1}^{F_{m+1}} - X| = |H| + q$. Let α denote the number of isolated nodes of $G_{m+1} - X - F_{m+1}$ which are in H . Clearly,

$$c_1(G_{m+1} - X - F_{m+1}, A_{m+1}^{F_{m+1}} - X) = \alpha + q.$$

Any odd component of some $G_i[Z_{j,m+1}^{(i)}] - F_{ij}$ is also an odd component of $G_i - X - F_i$. Also, the α isolated nodes of H will be isolated nodes also in $G_i - X - F_i$. So

$$c(G_i - X - F_i, A_i^{F_i} - X) \geq \alpha + \sum_j c(G_i[Z_{j,m+1}^{(i)}] - F_{ij}, U_{j,m+1}^{(i)}).$$

From all these equations/inequalities we get

$$(122) \quad |A_i^{F_i} - X| - c(G_i - X - F_i, A_i^{F_i} - X) \leq |H| - \alpha - q + |A_i \cap U|$$

and

$$(123) \quad |A_{m+1}^{F_{m+1}} - X| - c_1(G_{m+1} - X - F_{m+1}, A_{m+1}^{F_{m+1}} - X) = |H| - \alpha$$

Since $|A_i \cap U| - q = |A_i| - |A_{m+1}| = 2|\mathcal{P}_i| - 2|\mathcal{P}_{m+1}|$, finally we get

$$(124) \quad \begin{aligned} |A_i^{F_i} - X| - c(G_i - X - F_i, A_i^{F_i} - X) &\leq \\ &\leq (|A_{m+1}^{F_{m+1}} - X| - c_1(G_{m+1} - X - F_{m+1}, A_{m+1}^{F_{m+1}} - X)) + 2|\mathcal{P}_i| - 2|\mathcal{P}_{m+1}| \end{aligned}$$

hence we are done. \square

We show that Claim 3.33 implies Lemma 3.31. To see this, suppose $i \geq 2$. So, (119) holds for $i, i + 1, \dots, m$, but not for $i - 1$. Now, notice that $\{S_{i-1}\}$ is a terminal in G_i which is uncovered by \mathcal{P}_i . By Lemma 3.33, F_i, X is a verifying pair for \mathcal{P}_i . $\{S_{i-1}\}$ is exposed by \mathcal{P}_i , so by a slackness condition $\{S_{i-1}\}$ must be in an odd component of $G_i - X - F_i$. If $\{S_{i-1}\} \in H$, then clearly, (119) also holds for $i - 1$. If $\{S_{i-1}\}$ is in one of $G_i[Z_{j,m+1}^{(i)}]$, then for some $i' \geq i$ $\{S_{i-1}\}^{(m+1)} = \{S_{i'}\}^{(m+1)}$. This also implies (119) for $i - 1$, a contradiction.

□□

Maintaining the dragon-decomposition

Definition 3.34 A *dragon-decomposition* is given by a forest $T \subseteq E$ which has the following properties. Let $A^T := A \cup V(T)$.

1. For each $a \in A$ there is a component T_a of the (forest) subgraph (A^T, T) such that $A \cap V(T_a) = \{a\}$. Moreover, $\{T_a : \text{for each } a \in A\}$ are all the components of (A^T, T) .
2. Let $\omega^T : A^T \rightarrow \Omega$ be the (uniquely defined) function s.t. each edge in T is ω^T -balanced. Let F be the set of ω^T -balanced edges. Let \mathcal{K} is the family of components of $G - F$.
3. T/\mathcal{K} is a tree.
4. $K, A^T \cap V(K), \omega^T, \pi$ is critical for every $K \in \mathcal{K}$.

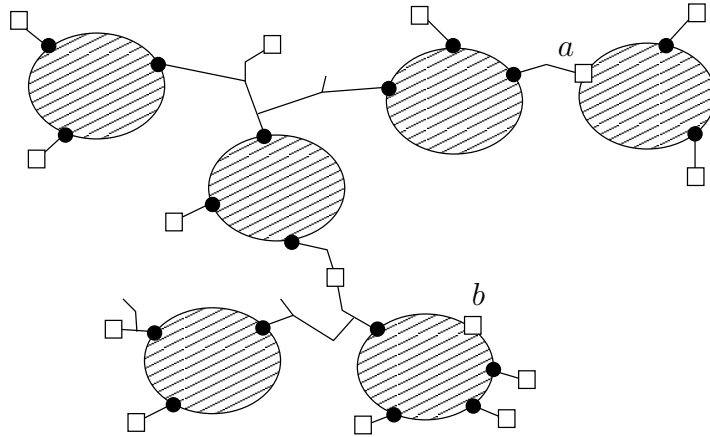


Figure 16: A dragon-decomposition.

Some remarks on the definition of dragon decompositions: The definition implies that $|A|$ is odd. In Figure 16, squares are the terminals, bold dots are so-called

“borders” of components, the shaded areas are components. Edges in $F - T$ are not depicted. In trees T_a (joining components to each other and to the terminals) some nodes may collapse; however, to keep components disjoint, borders must have positive distance. For example in Figure 16, a collapsed with one of the borders in T_a , and so did b in T_b . Note that, T_a may be any kind of tree. However, by shrinking some of the edges in some T_a , in some of the proofs we may assume that T_a bounded by a and the borders in T_a , i.e. T_a has no pendant branches.

Firstly, let us show the main idea about dragon-decompositions, which is showing that this structure implies the dragon property. We will in fact be able to determine precisely which nodes are uniquely-/multiply reachable. These facts are shown in the following claim. It is easy to see that the assertions of Lemma 3.26 follow from this, if G is a dragon having a dragon-decomposition.

Claim 3.35 *A p-graph with a dragon-decomposition is a dragon. A^T is exactly the set of uniquely reachable nodes. A node $x \in A^T$ is $\omega^T(a)$ -reachable by the unique half- A -path inside some T_a . $2\nu = |A| - 1 = |A^T| - c_{\text{odd}}(G - F, A^T)$.*

Proof. We begin with an easy but important observation, which is in fact the one and only tool to construct packings in a p-graph with a dragon-decomposition. Let $\mathcal{K} = \{K_1, K_2, \dots, K_k\}$. Consider a component K_i , let us call nodes in $V(K_i) \cap V(T)$ “borders of K_i ”. Choose one of the borders, say $b \in V(K_i) \cap V(T) := \{b_1, \dots, b_{2m+1}\}$. The following assertion is easy to see from Definition 3.34.

(125) There is a packing of m non-returning paths inside $T \cup K_i$ disjoint from b .

Notice that, a packing given by (125) only uses edges in K_i or edges in a component T_a of T s.t. $V(T_a)$ contains a border of K_i other than b . To prove Claim 3.35, we distinguish two cases:

Case A. $v \in V(T_a)$ for some $a \in A$. Let b_i denote the border of K_i which is “closest to v ” (i.e. b_i is the end of the pre-image of the unique $v - \{K_i\}$ path in tree T/\mathcal{K}). Let \mathcal{P}_i be a packing given by (125) applied for K_i, b_i . It is easy to see that $\mathcal{P} := \bigcup \mathcal{P}_i$ is a packing, and is disjoint from $V(T_a)$. Also notice that, \mathcal{P} covers all terminals in $A - a$. So v is reachable, just use the unique $a - v$ path in T_a .

Case B. Say $v \in V(K_1) - \bigcup V(T_a)$, so v is a non-border node of K_1 . Suppose v is ω -reachable in K_1 (i.e. in the critical p-graph $K_1, A^T \cap V(K_1), \omega^T, \pi$). Next we show that v is also ω -reachable in the whole p-graph G . For this, we define \mathcal{P}_i for $i \neq 1$ analogue to case A. It is easy to see that $\bigcup \mathcal{P}_i$ is a packing which is node-disjoint from K_1 and from the T_a 's meeting K_1 ; moreover, $\bigcup \mathcal{P}_i$ covers all the terminals, except for those terminals a for which T_a meets K_1 . We construct \mathcal{P} by adding a packing “through K_1 and the trees T_a meeting K_1 ” s.t. there will also be

a half- A -path ending in v with potential ω . This is easily constructed by using the unique paths in these T_a 's linking K_1 to a .

So we have proved that G is a dragon, and that all the nodes in $V(K_1) - \bigcup V(T_a)$ are multiply reachable. To prove the remainder of our claim we have to do the following considerations. Firstly, notice that F is A -balanced, and that the components K_i are exactly the odd components we count to determine c_{odd} in the min-max formula. Also notice that, $A^T = A^F$. So $|A^T| - c_{\text{odd}}(G - F, A^T)$ is a lower bound on 2ν (by ‘‘necessity’’ in the min-max formula). We show equality by using the construction given for case A, as follows. Let \mathcal{P} be constructed for some $v \in V(T_a)$. For every K_i , \mathcal{P} has the following property:

- (126) If K_i has $2r + 1$ borders, then there are r paths in \mathcal{P} which use two borders of K_i and each of these paths uses at most one border in any component other than K_i .

So $|\mathcal{P}|$ must be equal to the number of all the borders minus k . Clearly, the set of borders is subset of A^T ; and $k = c_{\text{odd}}(G - F, A^T)$.

We still need to show that the nodes in $\bigcup V(T_a)$ are uniquely reachable. This will be proved using equation $2|\mathcal{P}| = |A| - 1 = |A^T| - c_{\text{odd}}(G - F, A^T)$. Equality here implies ‘‘equality throughout’’, in other words, if $|\mathcal{P}| = (|A| - 1)/2$, then property (126) holds for every K_i . So a maximum packing \mathcal{P} occupies at least all but one of the borders in any component. Thus a half- A -path in $G - V(\mathcal{P})$ either enters some $V(K_i)$ and never comes out of $V(K_i)$, or is disjoint from $\bigcup V(K_i)$. Hence, a node in $\bigcup V(T_a)$ is only reachable by half- A -paths on edges in F . As F is A -balanced, this implies that the nodes in $A^T = \bigcup V(T_a)$ are uniquely reachable. \square

Notice that a dragon-decomposition – if there is one – must be more or less unique. F is unique, since it is determined as the set of edges xy s.t. x, y is uniquely reachable (say ω_x, ω_y -reachable) and $\pi(xy)(\omega_x) = \omega_y$. However, forest T is not defined uniquely, we only know that it is contained in F .

Lemma 3.36 *Dragons have a dragon decomposition.*

Our strategy to prove this lemma is the following. First we only prove that in our algorithm we are able to maintain a dragon-decomposition of dragons appearing in the sequence of contractions. In the end, Lemma 3.36 will follow by observing that a dragon has no other verifying pair than the trivial one. So we first prove the following weakening of Lemma 3.36.

Lemma 3.37 *Dragons in a sequence of contractions have a dragon decomposition.*

It is easy to see that Lemma 3.26 follows from Lemma 3.36 and Claim 3.35. In fact, to have our algorithm working, we in fact only need Lemma 3.37 and Claim

3.35. Lemma 3.36 will be proved in the end from Lemma 3.37 by showing that a dragon has no other decomposition than the trivial decomposition.

The proof of Lemma 3.37 goes by maintaining a dragon-decomposition through the following operations.

1. Given a dragon-decomposition of $G = (V, E)$, find the dragon-decomposition of $G' = G + ab = (V, E + ab)$ for some $a, b \in V$.
2. Given a dragon-decomposition of Z by a forest T . If G/Z is a rod, then find the decomposition of G .
3. Given a node-disjoint family \mathcal{Z} of p-graphs which are subgraphs in a p-graph G . Suppose each member of \mathcal{Z} is given together with a dragon-decomposition, and suppose G/\mathcal{Z} is an odd cycle. Find a dragon-decomposition of G .

Operation 2 is easy to solve. Suppose G is built up by adding an edge ab to Z such that b is a new node. If T gives a dragon-decomposition of Z , then T or $T + ab$ also gives a dragon-decomposition of G , just check the definition.

Let us call a dragon decomposition given by a forest T a **pure decomposition** if $F = T$. So, for a pure decomposition property 3 means that G/\mathcal{K} is a tree. A dragon does not necessarily have a pure decomposition! The proof of the following lemma will be given below.

Lemma 3.38 *Given a node-disjoint family \mathcal{Z} of p-graphs which are subgraphs in a p-graph G . Suppose each member of \mathcal{Z} is given together with a pure decomposition, and suppose G/\mathcal{Z} is an odd cycle. Then one can construct a pure decomposition of G .*

Let us first see how to use this lemma to solve an operation of type 3 in general. We first omit some edges of G to get some family \mathcal{Z} of purely decomposables. Lemma 3.38 produces a decomposition, then, in the end we put back the omitted edges. These are operations of type 1. Notice, however, that most operations of type 1 are in fact special cases of type 3, which will be solved by Lemma 3.38, too. So only those operations of type 1 remain which are not special cases of type 3. Up to this point we have a pure decomposition (by some forest T) of some spanning subgraph (V, E') for some $E' \subseteq E$. Now we have to put back a set $E - E'$ of omitted edges such that none of them maps to a non-returning loop (=odd cycle) in $G/(V, E')$. We claim that T also gives a decomposition of G , since we get $F = T \cup (E - E')$. This follows from the definition and Claim 3.35.

Proof of Lemma 3.38. G is built up by putting members of \mathcal{Z} cyclically next to each other, and joining them by the pre-images of the odd cycle. So each $Z \in \mathcal{Z}$

will be incident with two such edges. Let us introduce some more notation for this. Let $\mathcal{Z} = \{Z_1, \dots, Z_{2k+1}\}$, and let the pre-image of the odd cycle be $\{a_i b_{i+1} : i = 1, \dots, 2k+1\}$ (addition mod $2k+1$), where $a_i, b_i \in Z_i$. Recall that, by definition of an odd cycle and of a contraction, for any i , a_i, b_{i+1} are reachable inside Z_i , resp. Z_{i+1} such that edge $a_i b_{i+1}$ produces a non-returning A -path.

To simplify the proof we will make one more assumption – besides assuming to have pure decompositions for the Z_i 's, say given by forest T^i , with components \mathcal{K}^i . Notice that one can build up a pure decomposition by growing a sub-forest of T^i/\mathcal{K}^i . So in the first place we assume that T^i/\mathcal{K}^i is “path-like”, and a_i, b_i are in the most distant parts of this path structure. Then, in the end, we can glue back the pendant branches – in the next paragraph we will be more precise in how to do this.

Suppose we have a pure decomposition of Z given by forest T , with components \mathcal{K} . Suppose some $K \in \mathcal{K}$ is a “leaf” – that is one of its borders x is a cutting node between K and $\mathcal{K} - K$. Say K is incident with trees $T_{a_1}, \dots, T_{a_{2m+1}}$ with $x \in T_{a_{2m+1}}$. Then it is easy to see that $T' := T - \bigcup_{i=1}^{2m} T_{a_i}$ purely decomposes $Z' := Z - K - \bigcup_{i=1}^{2m} T_{a_i}$. Moreover, it is easy to see that Z/Z' is a dragon with a pure decomposition $\bigcup_{i=1}^{2m} T_{a_i}$. So a pure decomposition can be dismantled by cutting off one of its “leaves”, or, by induction, cutting off some of its “branches”. In the other direction, if some Z' has a pure decomposition, and Z/Z' has a pure decomposition, and the pre-image of $E(Z/Z')$ has exactly one (cutting) node in common with $V(Z')$, then the union of these two pure decompositions gives a pure decomposition of Z .

By the above reasoning, we may assume without loss of generality that each Z_i has the following “path-like” structure, with a_i, b_i being on the opposite ends of it. (We may assume this since “pendant branches” we cut off, and glue back in the end after a pure decomposition of G is found.) We may also assume – by shrinking some edges in the forest – that the forest only has path-components and possibly some “Y”-components, see Figure 17.

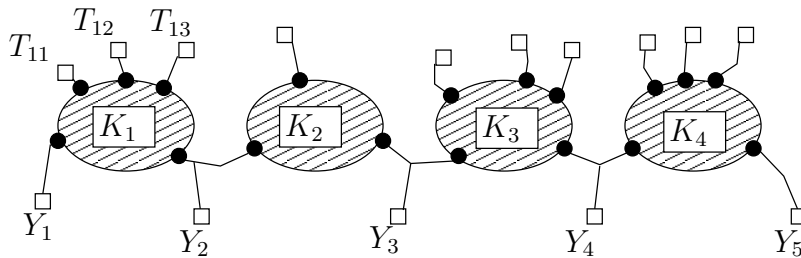


Figure 17: A pure decomposition with a “path-like structure”.

In a path-like pure decomposition, let K_j ($j = 1, \dots, l$) be the components in the order of appearance, let Y_j ($j = 1, \dots, l+1$) be the tree before K_j or after K_{j-1} , and let T_{jq} be the paths incident with K_j (clearly there is an odd number of

T_{jq} 's for a fixed j .) To interpret Figure 17, the shaded parts are the components K_j the inner structure of which is not in discussion here – we only have property 3, i.e. the criticality with respect to some induced potential on the border. The bold dots denote the border of components in the figure, the squares denote the terminals. Our definition allows that some of the paths depicted in the figure vanish as zero-length paths. However the path on the tree Y_{j+1} joining K_j and K_{j+1} may not vanish – this would result in glueing K_j and K_{j+1} to form one bigger component. It is important that the bold dots (the borders) are all pairwise distinct nodes, and any component has an odd number of borders.

So, our assumption (besides having a pure decomposition) is that Z_i has a path-like decomposition with notation $\mathcal{K}^i, K_j^i, Y_j^i, T_{jq}^i$, moreover, $a_i \in K_1^i \cup Y_1^i$ and $b_i \in K_l^i \cup Y_{l+1}^i$. So G has a cyclic structure, Figure 18 shows an example. This structure is built up by putting “path-like” decompositions cyclically next to each other, and joining their leaves $K_1^i \cup Y_1^i$ and $K_l^{i+1} \cup Y_{l+1}^{i+1}$ by some edge $a_i b_{i+1}$. These edges $a_i b_{i+1}$ are drawn as dotted lines.

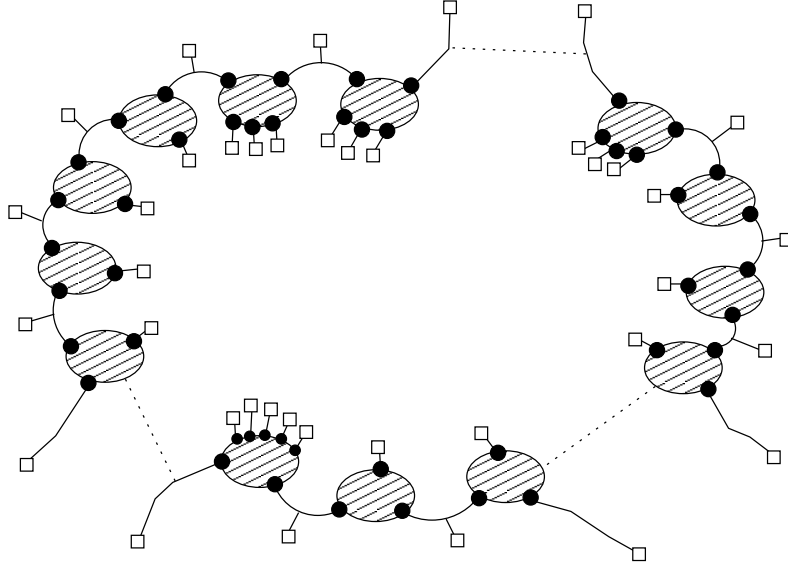


Figure 18: The cyclic structure.

It is easy to see that $G/\bigcup \mathcal{K}^i$ (shrinking the shaded parts) is a graph with a unique cycle Q with an odd number of pairwise node-disjoint pendant paths attached to Q . Let K denote the pre-image of Q . Clearly, K contains all the K_j^i 's, and all the edges $a_i b_{i+1}$. Let $T := E - E(K)$, which is equal to the the union of the pre-image of Q 's pendant paths.

Claim 3.39 T determines a pure decomposition of G .

Proof. Properties 1 and 3 are straightforward, property 2 is just a definition. To

prove property 4 we need to show that if $T = \emptyset$, then G is critical. (This case is exactly the case when pendant paths on Q have zero length.) Since Z_i are dragons by themselves it follows that all nodes in G are reachable. We need to show that nodes in $V - A$ are multiply reachable. Since the K_j^i 's are critical by themselves – a pure decomposition is given inside the Z_i 's – multiple reachability for the non-border nodes of K_j^i 's also follows. We are only concerned with the non-terminal nodes in the paths in G joining two neighboring K_j^i 's.

By shrinking, we may assume that the cyclic structure of K is built up as follows – demonstrated in Figure 20.

1. Put critical K_j^i 's cyclically next to each other.
2. Join two neighboring K_j^i 's by one of the ways in Figure 19 using altogether an odd number of dashed lines (the $a_i b_{i+1}$'s).

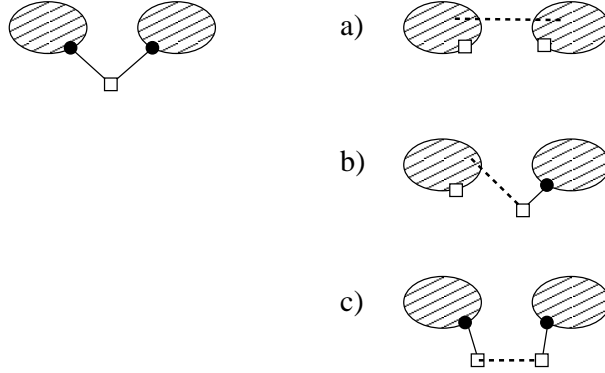


Figure 19: The possible paths connecting neighboring K_j^i 's.

To prove that K is critical, first we prove the following claim which is equivalent with that increasing the capacity of such a node a from 1 to 2 increases ν .

Claim 3.40 *Given an arbitrary terminal $a \notin \bigcup K_j^i$. Then there is a family of $\nu + 1$ non-returning paths such that they are pairwise node-disjoint, except for node a , where there start 2 paths.*

Proof. Such a terminal a has degree 2 in G . Let us split a into 2 copies a' and a'' such that the two incident edges are split 1-1 among them. So G' has one more terminal, $G' / \bigcup K_j^i$ is an a' - a'' path on an even number of nodes, and the pre-images of nodes in $G' / \bigcup K_j^i$ are dragons. Consider the perfect matching M in $G' / \bigcup K_j^i$. By Claim 3.28, there is a \mathcal{P}' packing in G' such that $\mathcal{P}' / \bigcup K_j^i = M$. The pre-image of \mathcal{P}' is a family of non-returning paths in G as required. \square

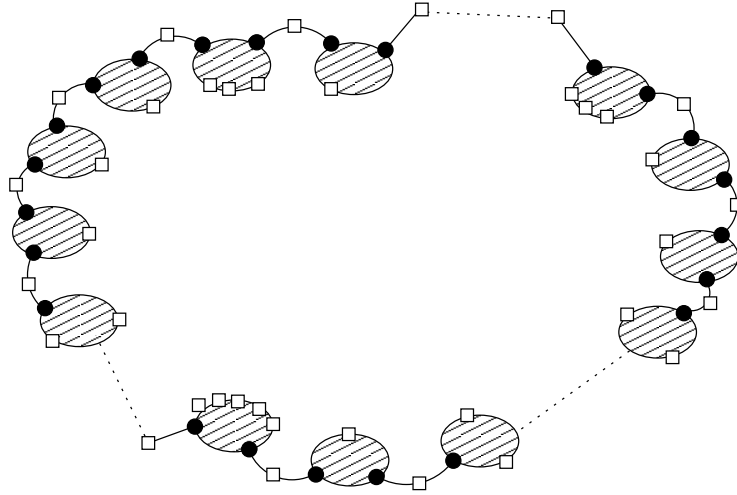


Figure 20: K

To prove that K is critical we need to prove that the bold dots are multiply reachable. Consider the bold dot z , i.e. a border which is not a terminal. We have already shown that z is reachable by the unique a - z path disjoint from $\bigcup K_j^i$. Then a is a terminal not in $\bigcup K_j^i$. Consider a family \mathcal{P} as of Claim 3.40, say with paths $P_1, P_2 \in \mathcal{P}$ containing a . Then one of them, say P_1 contains the (unique) a - z path. Let P'_1 denote the other section of P_1 ending in z . Then – since P_1 was non-returning – P'_1 reaches z with another different potential. Also P'_1 is disjoint from the maximum packing $\mathcal{P} - P_1$. Thus z is multiply reachable, hence K is critical. This proves Claim 3.39, and concludes the proof of Lemma 3.37. \square

Proof of Lemma 3.36. Consider a dragon G , run the algorithm. We find a verifying pair X, F . In any maximum packing, nodes in X must be traversed by some path. So $X = \emptyset$.

4 Open Questions

The below list includes open questions which the author considers for future research, and could have fit into the thesis, if they had been solved. Hereby we omit some well-known and often cited open questions, and rather focus on those related to methods and frameworks used in this thesis. Some of the open questions are accompanied by a short remark on conjectures or related results, while some others are discussed more comprehensively in the respective sections.

1. Is there a simple combinatorial algorithm to decide, given an undirected graph, whether there is a square-free 2-factor? (In this thesis, a simple combinatorial algorithm has been given for bipartite graphs.)
2. Formulate a natural, tractable common generalization of path-matching and packing \mathcal{A} -paths! Sebő and Szegő [80] proposed a conjecture on the existence of a packing of $|A|/2$ \mathcal{A} -paths the deletion of which results in a perfectly matchable subgraph. Is this problem indeed tractable?
3. Is there a strongly polynomial time algorithm to find a maximum half-integral fractional b -packing of \mathcal{A} -paths? (Previously we have constructed a semi-strongly polynomial time algorithm.)
4. Is the problem of finding a maximum weight packing of \mathcal{A} -paths tractable? Is the problem of finding a maximum weight fractional packing of \mathcal{A} -paths tractable? We remark that, in the first place, it is not quite clear what kind of a weight should be assigned with a (fractional) packing of \mathcal{A} -paths. We have shown that fractional packing of \mathcal{A} -paths is related with an instance of matroid fractional matching, which concerns the following questions.
5. Can matroid fractional matching be reduced to an instance of matroid matching? We remark that, if the problem is given on pairs in a full linear matroid, then the problem of finding a maximum size fractional matching may be solved by applying a linear matroid matching algorithm, and performing some adjustments in the end.
6. Is there a polynomial time separation algorithm for the matroid fractional matching polytope $\mathcal{P}(\mathcal{M}, E)$? We remark that Chang, Llewellyn and Vande Vate [7, 8] constructed a polynomial time algorithm to test membership of a half-integer vector in $\mathcal{P}(\mathcal{M}, E)$.
7. Is there a polynomial time algorithm to find a maximum weight fractional matching with respect to an arbitrary objective w ? We remark that Chang, Llewellyn and Vande Vate [7, 8] constructed a polynomial time algorithm to

find a maximum size fractional matching, that is, maximizing with respect to objective $w = 1$.

8. Are Mader delta-matroids representable? (A representation is given by a skew-symmetric matrix.) Our proof on Mader matroids provides no clue on the delta-matroid induced by packings on the terminals.
9. We see great similarity between our non-bipartite matching algorithm and our algorithm for the Partition Formula, which applies for polymatroids without ntedc's. Unluckily, we don't know a reduction of non-bipartite matching to matching in such polymatroids. Is there a framework dealing with the Partition Formula, and including non-bipartite matching as a special case?
10. Does the Weak Triple Exchange Axiom imply that the maximum cardinality of a packing of allowed A -path is equal to $\min \widehat{\nu}(G - F, A \cup V(F))$, where the minimum is taken over edge-sets F not containing allowed A -paths?

References

- [1] E. Balas and W. Pulleyblank, *The perfectly matchable subgraph polytope of an arbitrary graph*, *Combinatorica* **9** (1989) 321-337.
- [2] A. Benczúr, L. Végh, *Primal-dual approach for directed vertex-connectivity augmentation and generalizations*, EGRES Technical Report TR-2005-06, www.cs.elte.hu/egres (to appear in *ACM Transactions on Algorithms*)
- [3] A. Benczúr, L. Végh, *Primal-dual approach for directed vertex-connectivity augmentation and generalizations*, *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005*, Vancouver, British Columbia, Canada, 186-194
- [4] C. Berge, *Sur le couplage maximum d'un graph*, *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences* **247** (1958) 258-259.
- [5] Cs. Berki, A \mathcal{T} -path packing algorithm, masters thesis, 2002.
- [6] A. Björner, L. Lovász, *Pseudomodular lattices and continuous matroids*, *Acta Scientiarum Mathematicarum*, Szeged **51** (1987) 295-308.
- [7] S. Chang, D.C. Llewellyn, J. Vande Vate, *Matching 2-lattice polyhedra: finding a maximum vector*, *Discrete Mathematics* **237** (2001) 29-61.
- [8] S. Chang, D.C. Llewellyn, J. Vande Vate, *Two-lattice polyhedra: duality and extreme points*, *Discrete Mathematics* **237** (2001) 63-95.
- [9] B.V. Cherkassky, *Mnogopolyusnye dvukhproduktovye zadachi*, [Russian: Multi-terminal two commodity problems], in: *Issledovaniya po Diskretnoi Optimizatsii* [Russian: Studies in discrete optimization] (A.A. Friedman, ed.), Izdatel'stvo "Nauka", Moscow, 1976, 261-289.
- [10] M. Chudnovsky, J.F. Geelen, B. Gerards, L. Goddyn, M. Lohman, P. Seymour, *Packing non-zero A-paths in group-labeled graphs*, submitted, (2004)
- [11] M. Chudnovsky, W.H. Cunningham, J.F. Geelen, *An algorithm for packing non-zero A-paths in group-labeled graphs*, submitted, (2004)
- [12] G. Cornuéjols and D. Hartvigsen, *An extension of matching theory*, *Journal of Combinatorial Theory Ser. B* **40** (1986) 285-296.
- [13] G. Cornuéjols, D. Hartvigsen and W. Pulleyblank, *Packing subgraphs in a graph*, *Op. Res. Letters*, (1982) 139-143.

- [14] G. Cornuéjols and W. Pulleyblank, *A matching problem with side conditions*, Discrete Mathematics **29** (1980) 135-159.
- [15] G. Cornuéjols and W. Pulleyblank, *Critical graphs, matchings and tours or a hierarchy of the travelling salesman problem*, Combinatorica **3(1)** (1983) 35-52.
- [16] W.H. Cunningham, *Matching, Matroids and Extensions*, Math. Program. Ser. B **91** (2002) 3, 515-542.
- [17] W.H. Cunningham and J.F. Geelen, *The Optimal Path-Matching Problem*, Combinatorica, **17/3** (1997), 315-336.
- [18] W.H. Cunningham and J.F. Geelen, *Combinatorial Algorithms for Path-Matching*, manuscript, (2000).
- [19] W.H. Cunningham, J.F. Geelen, *Vertex-disjoint directed paths and even circuits*, manuscript
- [20] W.H. Cunningham, J.F. Geelen, *The optimal path-matching problem*, Combinatorica **17** (1997) 315-337.
- [21] J. Edmonds, *Paths, trees, and flowers*, Canadian Journal of Mathematics **17** (1965) 449-467.
- [22] J. Edmonds, *Submodular functions, matroids, and certain polyhedra*, in: Combinatorial Structures and Their Applications (Proceedings Calgary International Conference on Combinatorial Structures and Their Applications, Calgary, Alberta, 1969; R. Guy, H. Hanani, N. Sauer, J. Schönheim, eds.), Gordon and Breach, New York (1970) 67-87.
- [23] A. Dress, L. Lovász, *On some combinatorial properties of algebraic matroids*, Combinatorica **7** (1997) 39-48.
- [24] A. Frank, *Restricted t -matchings in bipartite graphs*, Discrete Applied Mathematics **131/2** (2003) 337-346.
- [25] A. Frank, T. Jordán, *Minimal edge-coverings of pairs of sets*, Journal of Combinatorial Theory Ser. B, **65** (1995) 73-110.
- [26] A. Frank, T. Jordán, Z. Szigeti, *An orientation theorem with parity conditions*, Discrete Applied Mathematics **115** (2001) 37-47.
- [27] A. Frank and L. Szegő, *A Note on the Path-Matching Formula*, J. of Graph Theory **41/2**, (2002) 110-119.

- [28] A. Frank, É. Tardos, *An application of simultaneous diophantine approximation in combinatorial optimization*, *Combinatorica* **7** (1987) 40-65.
- [29] Zs. Fekete, *Source location with rigidity and tree packing requirements*, EGRES Technical Report TR-2005-04, www.cs.elte.hu/egres
- [30] M.L. Furst, J.L. Gross, L.A. McGoech, *Finding a maximum genus graph embedding*, *J. ACM* **35** (3) (1988) 523-534.
- [31] H.N. Gabow, M. Stallmann, *An augmenting path algorithm for linear matroid parity*, *Combinatorica* **6** (1986) 123-150
- [32] T. Gallai, *Maximale Systeme unabhängiger Kanten*, *A Magyar Tudományos Akadémia Matematika Kutatóintézetének Közleményei* **9** (1964) 401-413.
- [33] T. Gallai, *Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen*, *Acta Mathematica Academiae Scientiarum Hungaricae* **12** (1961) 131-137.
- [34] J.F. Geelen, S. Iwata, K. Murota, *The linear delta-matroid parity problem*, *Journal of Combinatorial Theory Ser. B* **88** (2003) 377-398.
- [35] A.M.H. Gerards, *Matching*, in: *Network Models* (M.O. Bal, T.L. Magnanti, C.L. Monma, G.L. Nemhauser, eds.) [Handbooks in Operations Research and Management Science, Vol 7] Elsevier, Amsterdam, 1995
- [36] D. Gijswijt, *On the fractional matroid matching polytope*, manuscript, private communication, 2006
- [37] M. Grötschel, L. Lovász, A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, *Combinatorica* **1** (1981) 169-197.
- [38] D. Hartvigsen, *Extensions of Matching Theory*, PhD thesis, 1984
- [39] D. Hartvigsen, *The square-free 2-factor problem in bipartite graphs*, in: *Integer Programming and Combinatorial Optimization* (eds. G. Cornuéjols, R. Burkard, G.J. Woeginger) Springer Verlag, Lecture Notes in Computer Science, 1610, 1999, pp. 234-240. (Extended abstract.)
- [40] D. Hartvigsen, *Finding maximum square-free 2-matchings in bipartite graphs*, *Journal of Combinatorial Theory, Ser. B*, in press
- [41] W. Hochstättler, W. Kern, *Matroid matching in pseudomodular lattices*, *Combinatorica* **9** (1989) 145-152.

- [42] P.M. Jensen, B. Korte, *Complexity of matroid property algorithms*, SIAM J. Computing **11** (1) (1982) 184-190.
- [43] J.C.M. Keijsper, R.A. Pendavingh, L. Stougie, *A linear programming formulation of Mader's edge-disjoint paths problem*, J. of Combinatorial Theory Ser. B **96** (2006) 159-163.
- [44] T. Király, J. Szabó, *A note on parity constrained orientations*, EGRES Technical Report TR-2003-11, www.cs.elte.hu/egres
- [45] Z. Király, *C_4 -free 2-factors in bipartite graphs*, EGRES Technical Report TR-2001-13, www.cs.elte.hu/egres
- [46] T. Király, M. Makai, *On polyhedra related to even factors*. in: Integer programming and combinatorial optimization, D. Bienstock, G. Nemhauser, eds. 416–430, Lecture Notes in Comput. Sci., 3064, Springer, Berlin, 2004.
- [47] D. König, *Graphok és matrixok*, Matematikai és Fizikai Lapok **38** (1931) 116-119.
- [48] M. Kriesell, *Disjoint A -paths in digraphs*, J. Combin. Theory Ser. B **95** (2005) 168-172.
- [49] E.L. Lawler, *Matroids with parity conditions: a new class of combinatorial optimization problems*, Memorandum ERL-M334, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, California (1971)
- [50] M. Loebl and S. Poljak, *Efficient Subgraph Packing*, Journal of Combinatorial Theory Ser. B, **59** (1993), 106-121.
- [51] L. Lovász, *Matroid matching and some applications*, Journal of Combinatorial Theory Ser. B **28** (1980) 208-236.
- [52] L. Lovász, *Selecting independent lines from a family of lines in a space*, Acta Sci. Math. Szeged, **42**(1-2) (1980), 121-131.
- [53] L. Lovász, *On some connectivity properties of Eulerian graphs*, Acta Mathematica Academiae Scientiarum Hungaricae **28** (1976) 129-138.
- [54] L. Lovász, *On determinants, matchings, and random algorithms*, in: Fundamentals of Computation Theory FCT'79, Proceedings of the Conference on Algebraic, Arithmetic, and Categorical Methods in Computation Theory, Berlin 1979, L. Budach ed., Akademie-Verlag, Berlin

- [55] L. Lovász, *The matroid matching problem*, in: Algebraic Methods in Graph Theory, Vol. II (Colloquium Szeged, 1978; L. Lovász, V.T. Sós, eds.) Colloquia Mathematica Societatis János Bolyai **25**, North-Holland, Amsterdam, 1981, 495-517.
- [56] L. Lovász and M.D. Plummer, *Matching Theory*, Akadémiai Kiadó, Budapest, 1986.
- [57] W. Mader, *Über die Maximalzahl kreuzungsfreier A-Wege*, Archiv der Mathematik **30** (1978) 325-336.
- [58] W. Mader, *Über die Maximalzahl kreuzungsfreier H-Wege*, Archiv der Mathematik **31** (1978) 387-402.
- [59] M. Makai, J. Szabó, *Polymatroids without double circuits*, manuscript, private communication, 2006
- [60] M. Makai, *On maximum cost K_{tt} -free t -matchings of bipartite graphs*, Proceedings of "Graph Theory 2004" in memoriam Claude Berge, Paris
- [61] K. Menger, *Zur allgemeinen Kurventheorie*, Fundamenta Mathematicae **10** (1927)
- [62] Nebeský, *A new characterization of the maximum genus of a graph*, Czechoslovak Math. J. **31** (106) (1981) 604-613.
- [63] J.B. Orlin, J. Vande Vate, *Solving the linear matroid parity problem as a sequence of matroid intersection problems*, Math. Programming **60** (1993) 255-276.
- [64] G. Pap, *Packing non-returning A-paths*, Combinatorica, to appear
- [65] G. Pap, *Packing non-returning A-paths algorithmically*, submitted
- [66] G. Pap, *Alternating paths revisited I: even factors*, EGRES Technical Report, TR-2004-18, www.cs.elte.hu/egres, submitted
- [67] G. Pap, *Négyzetmentes 2-párosítások páros gráfokban*, Matematikai Lapok, submitted
- [68] G. Pap, *Packing non-returning A-paths*, EGRES Technical Report, TR-2005-12, www.cs.elte.hu/egres
- [69] G. Pap, *Alternating paths revisited II: restricted b-matchings in bipartite graphs*, EGRES Technical Report, TR-2005-13, www.cs.elte.hu/egres

- [70] G. Pap, *Alternating paths revisited III: hypo-matchings in directed graphs*, EGRES Technical Report, TR-2005-14, www.cs.elte.hu/egres
- [71] G. Pap, *Alternating paths revisited IV: packings and 2-packings of A-paths*, EGRES Technical Report, TR-2005-15, www.cs.elte.hu/egres
- [72] G. Pap, *Hypo-matchings in Directed Graphs*, Graph Theory, Trends in Mathematics, 2006 Birkhäuser Verlag, Basel, Switzerland, 325-335.
- [73] G. Pap, *Combinatorial algorithms for matchings, even factors and square-free 2-factors*, Mathematical Programming Ser. B, IPCO XI special issue, to appear
- [74] G. Pap and L. Szegő, *On the Maximum Even Factor in Weakly Symmetric Graphs*, Journal of Combinatorial Theory Ser. B, **91/2** (2004) 201-213.
- [75] G. Pap and L. Szegő, *On factorizations of directed graphs by cycles*, Combinatorica, to appear
- [76] R. Rado, *A theorem on independence relations*, The Quarterly Journal of Mathematics, Oxford, **(2) 13** (1942) 83-89
- [77] A. Schrijver, *Combinatorial Optimization – Polyhedra and Efficiency*, Springer-Verlag, Berlin, Heidelberg, 2003
- [78] A. Schrijver, *A short proof of Mader’s \mathcal{S} -paths theorem*, Journal of Combinatorial Theory Ser. B **85** (2001) 319-321.
- [79] A. Schrijver, *Is each Mader matroid a gammoid?*, www.cwi.nl/lex.
- [80] A. Sebő, L. Szegő, *The path-packing structure of graphs*. in: Integer programming and combinatorial optimization, D. Bienstock, G. Nemhauser, eds., 256–270, Lecture Notes in Comput. Sci., 3064, Springer, Berlin, 2004.
- [81] B. Spille and L. Szegő, *A Gallai-Edmonds-type Structure Theorem for Path-Matchings*, Journal of Graph Theory, **46/2** (2004) 93-102.
- [82] B. Spille and R. Weismantel, *A Generalization of Edmonds’ matching and matroid intersection algorithms*, in: Proceedings of the 9th International IPCO Conference, 2002; W.J. Cook, A.S. Schultz, eds.
- [83] Z. Szigeti, *On the graphic matroid parity problem*, Journal of Combinatorial Theory Ser. B **88** (2003) 247-260.
- [84] W.T. Tutte, *The factorization of linear graphs*, The Journal of the London Mathematical Society **22**, (1947) 107-111.

- [85] V.V. Vazirani, *Approximation algorithms*, Springer, Berlin, 2001.
- [86] J. Vande Vate, *Fractional Matroid Matchings*, Journal of Combinatorial Theory Ser. B **55 B** (1992) 133-145.
- [87] J. Vande Vate, *Structural properties of matroid matchings*, Discrete Appl. Math., **39** (1992) 69–85.
- [88] Welsh, D.J.A. *Matroid Theory*, Academic Press, London, 1976.