REFERENCES


---

THE MATROID MATCHING PROBLEM

L. Lovász

0. INTRODUCTION. FORMULATION OF THE MATCHING PROBLEM

A polymatroid is a finite set endowed with a non-negative, monotone, submodular, integral valued set-function called "rank". Matroids are special polymatroids (when every singleton set has rank at most 1). A more general construction yielding polymatroids is the following: we take a set $S$ of flats in a matroid and the "rank" of a subset of $S$ is the matroid-rank of their union. In fact, every polymatroid arises this way [7, see also 5].

We shall call a polymatroid linear, if it is representable by flats in a projective space (or linear space). Not every polymatroid (not even every matroid) is linear, but most polymatroids arising from various applications of the theory are linear.

Let $(S, r)$ be a polymatroid. A set $X \subseteq S$ is called a matching if

$$ r(X) = \sum_{x \in X} r(x). $$

*This paper was written while the author was visiting the Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada.*
The matchings of $S$ do not form the independent sets of a matroid in general. We are concerned with the determination of the maximum cardinality of a matching, which we denote by $\nu = \nu(S, r)$.

**Example 1.** Let $S$ be the set of edges of a hypergraph. Define

$$r(X) = |\bigcup \{E : E \in X\}|$$

for $X \subseteq S$. Then $(S, r)$ is a polymatroid and $\nu(S, r)$ is the maximum number of pairwise disjoint edges in the hypergraph. If the hypergraph is a graph, the polymatroid will be called **graphic**.

**Example 2.** Let $(T, \rho)$ be a matroid and $S = \{T_1, \ldots, T_n\}$ a partition of $T$ into independent sets. Define

$$r(X) = \rho(\bigcup X)$$

for $X \subseteq S$. Then $\nu(S, r)$ is the maximum number of classes of $S$ whose union is independent.

**Example 3.** Let $(S, r_1), \ldots, (S, r_k)$ be matroids without loops. Define $r = r_1 + \ldots + r_k$. Then $\nu(S, r)$ is the largest common independent set of the given matroids.

In both Examples 1 and 3 the calculation of $\nu(S, r)$ is known to be NP-complete. However, if $r(|x|) = 2$ for every $x \subseteq S$, then Example 1 is the matching problem for graphs while Example 3 is the intersection problem for two matroids. Both problems are well-solved (see Edmonds [1], [2], Tutte [9], [10]).

This raises the problem of determining $\nu(S, r)$ for all polymatroids such that all elements have rank at most 2 (see Jensen and Yann [3]). Different formulations of this problem are known as the "matchoid problem" [Edmonds] and the "matroid parity problem" [Lawler [4]].

In this paper we first show that no polynomial-bounded algorithm exists which would solve the matroid parity problem in general. I have learnt recently that Professor B. Korte in Bonn has obtained this same result. Then we give an algorithm which solves the problem, among others, for all linear polymatroids. More precisely, the algorithm works on polymatroids which are represented (and not merely representable) as 2-dimensional subspaces of a linear space.

Besides giving an algorithm, solution of such a problem may mean to find a minimax formula (good characterization) for $\nu(S, r)$. Such a formula was given, for linear polymatroids, in [6]. As usual, our algorithm provides a new proof of the same formula:

**Theorem.** Let $H$ be a set of lines in a projective space, and let $\nu(H)$ denote the maximum number of lines in $H$ which form a matching. Then

$$\nu(H) = \min \left\{ r(A) + \sum_{i=1}^{k} \left\lfloor \frac{r(H_i + A) - r(A)}{2} \right\rfloor \right\}$$

where $A$ ranges over all flats of the projective space and $(H_1, \ldots, H_k)$ over all partitions of $H$.

It may help to understand the arguments which follow if it is pointed out that a flat $A$ which gives equality here must be contained in the span of every maximum matching.

1. **Preliminaries**

**Definition 1.1.** A pair $(S, r)$ where $S$ is a finite set and $r$ is an integral-valued function on its subsets is called a **polymatroid** if

I. $r(\emptyset) = 0$,

II. $X \subseteq Y \Rightarrow r(X) \leq r(Y)$,

III. $r(X \cup Y) + r(X \cap Y) \leq r(X) + r(Y)$.

$(S, r)$ is called a $k$-polymatroid if $r(x) = k$ for all $x \subseteq S$; it is called a $(k \leq 1)$-polymatroid if $r(x) \leq k$ for all $x$. $(k \leq 1)$-polymatroids are just matroids.

**Definition 1.2.** A polymatroid $(S, r)$ is said to be the **direct sum** of polymatroids $(U, r)$ and $(V, r)$ if $U \cup V = S$, $U \cap V = \emptyset$ and

$$r(U) + r(V) = r(S).$$
It is known that in this case \( r(X \cup Y) = r(X) + r(Y) \) whenever \( X \subseteq U \) and \( Y \subseteq V \).

**Example.** If \((S, r)\) is graphic then the decomposition of the graph into connected components corresponds to the decomposition of \((S, r)\) into direct summands.

**Definition 1.3.** Given a polymatroid \((S, r)\) and a set \( X \subseteq S \), the (unique) maximal set \( Y \subseteq S \) such that \( X \subseteq Y \) and \( r(X) = r(Y) \) shall be denoted by \( \text{Span}(S, r)^X \) or \( \text{Span} X \):

**Example.** If \((S, r)\) is graphic then \( \text{Span} X \) is the set of edges spanned by vertices covered by \( X \).

**Definition 1.4.** Let \((S, r)\) be a polymatroid, \( A \subseteq S \) and define

\[
r_A(X) = r(A \cup X) - r(A) \quad (X \subseteq S).
\]

\( r_A(X) \) is easily seen to be a polymatroid function.

**Proposition 1.5.** Let \((S, r)\) be a 2-polymatroid and \( A \) a maximum matching. Then \((S, r_A)\) is a matroid.

The following proposition (whose proof is omitted) seems to be interesting although will not be used in the sequel. It yields an analogue of matching matroids, defined by Edmonds and Fulkerson.

**Proposition 1.6.** Let \((S, r)\) be a \((\leq 2)\)-polymatroid and define

\[
r^*(X) = \max \{ r_A(X) : A \text{ max. matching} \}.
\]

Then \((S, r^*)\) is a matroid.

The following assertion generalizes Gallai’s identity for graphs.

**Proposition 1.7.** Let \((S, r)\) be a 2-polymatroid and \( T \) a minimum subset of \( S \) such that \( r(T) = r(S) \). Then

\[
|T| = r(S) - \nu(S, r).
\]

**Proof.** Let \( A \) be a maximum matching in \( T \) and \( X = T - A \). By Proposition 1.5,

\[
r(S) = r(T) \leq 2|A| + |T - A| = |T| + |A|;
\]

hence

\[
|T| \geq r(S) - |A| = r(S) - \nu(T, r) > r(S) - \nu(S, r).
\]

On the other hand, let \( A \) be a maximum matching in \((S, r)\) and \( X \) a basis of \((S - A, r_A)\). Then

\[
r(A \cup X) = r(A) + r_A(X) = r(A) + r_A(S - A) = r(S)
\]

and so

\[
|T| \leq |A \cup X| = |A| + |X| + r_A(X) = |A| + r_A(S - A) = r(S) - |A| = r(S) - \nu(S, r).
\]

**Definition 1.8.** Let \((C, r)\) be a 2-polymatroid. We call it a circuit if every proper subset of \( C \) is a matching but \( r(C) = 2|C| - 1 \) (note that not every minimal non-matching is a circuit, e.g., 3 skew lines in rank 4 space).

A polymatroid \((F, r)\) is called a \( \nu \)-flower if \( r(F) = 2\nu + 1 \) and \( |F| = \nu + 1 \). The name comes from the role of flowers, which is somewhat similar to the role of blossoms in Edmonds’ matching algorithm, although flowers do not generalize blossoms.

**Proposition 1.9.** Every flower \( F \) contains a unique circuit \( C \). The flower is direct sum of \( C \) and a matching. If \( f \in C \) then \( F - f \) is a matching; if \( f \in F - C \) then \( F - f \) is a flower.

**Definition 1.10.** A 2-polymatroid \((D, r)\) is called a double circuit if for each \( e \in D \), \( D - e \) is a flower. The double circuit is trivial if it is the direct sum of two circuits. A \( \nu \)-double flower is the direct sum of a double circuit and a matching, together having \( \nu + 2 \) lines.

**Example.** If \((S, r)\) is graphic then a circuit of \((S, r)\) is a path of length 2, and a \( \nu \)-flower is the union of a path of length 2 and \( \nu - 1 \) disjoint edges. A double circuit is either a 3-star or the disjoint union of two paths of length 2 (these latter subgraphs are the trivial double circuits).
Proposition 1.11. A \( \nu \)-double flower \( H \) has a partition \( H = H_0 \cup H_1 \cup \ldots \cup H_k \), such that \( H_0 \) is a matching. \( H \) is direct sum of \( H_0 \) and \( H - H_0 \); \( H - H_0 - H_1 \) is a circuit for all \( 1 \leq i \leq k \), and these are all the circuits in \( H \).

Proposition 1.12. Let \( (H, r) \) be a 2-polymatroid such that \[ r(H) = 2|H| - 2. \] Then one of the following possibilities hold.

1. There is a line \( e \in H \) such that \( H - e \) is a matching.
2. \( H \) is a \( \nu \)-double flower.

Let \( P \) be a projective space and \( X \subseteq P \). We denote by \( \bar{X} \) the flat spanned by \( X \).

Let \( A \) be a flat in \( P \). Take a flat \( B \) such that \[ A \cap B = \phi, \quad A \cup B = P. \]

It is well known that this implies \[ r(A) + r(B) = r(P). \]

Define, for each flat \( F \), \[ F / A = (F \cup A) \cap B. \]

Then \[ r(F / A) = r(F \cup A) - r(A). \]

If \( H \) is any collection of flats in \( P \), we set \[ H / A = \{ F / A : F \in H \}. \]

Note that \( H / A \) is, up to a linear isomorphism, independent of the choice of \( B \). This justifies our notation, which does not indicate \( B \).

If \( H \) is a set of flats in \( P \) then we denote by \( \bar{H} \) the least flat containing all members of \( H \) and set \( r(H) = r(\bar{H}). \)

Note that if \( X \subseteq H \) and \( A = \bar{X} \) then \[ H / A = (H; r_X). \]

The following simple lemma was proved in [6].

Lemma 1.13. Let \( H \) be a set of flats in a projective space \( P \) such that \[ \sum_{x \in H} (r(H) - r(x)) < r(H). \]

Then \( \bigcap H \neq \phi. \)

Corollary 1.14. Let \( D \) be a set of lines in a projective space which forms a non-trivial double flower. Let \( \mathcal{C}_1, \ldots, \mathcal{C}_k \) be the circuits in \( D \).

Then \[ K = \mathcal{C}_1 \cap \ldots \cap \mathcal{C}_k \neq \phi. \]

We shall call \( K \) the kernel of \( D \). Note the closely related fact that the 3 edges of a non-trivial double circuit in a graphic polymatroid have a point in common.

2. THE POLYNOMIAL UNSOLVABILITY OF THE MATCHING PROBLEM FOR GENERAL 2-POLYMATROIDS

First we have to make it more precise what kinds of algorithms do we really consider here. Let \( (S, r) \) be a 2-polymatroid. There is, of course, no way to store an arbitrary polymatroid (or even a matroid) on \( n \) elements in \( O(n^{\text{const}}) \) space; there are simply too many of them. So what we assume is that we have an algorithm which calculates the rank of any given set \( X \subseteq S \). The calculation of \( r(X) \) is only counted as one step. But the algorithm computing \( r(X) \) is considered as a "black box", we cannot make use of its internal properties in designing, say, our matching algorithm.

Known matroid algorithms like Edmonds’ intersection algorithm have the above feature.

Theorem 2.1. Every algorithm which decides \( \nu(S, r) \geq k \) for every 2-polymatroid \( (S, r) \) needs, in the worst case, more than \( \binom{n}{k} \) time, where \( n = |S| \).
We need a simple lemma whose proof is straightforward.

**Lemma 2.2.** Let $S$ be a finite set, $\nu \geq 1$ and integer and $f \colon 2^S \to \{0, \ldots, 1\}$ a set function such that

\[
\begin{align*}
&f(X) = 2|X| & \text{if} & & |X| \leq \nu, \\
&f(X) = 2\nu + 2 & \text{if} & & |X| > \nu + 2, \\
&f(X) \in (2\nu + 1, 2\nu + 2) & \text{if} & & |X| = \nu + 1.
\end{align*}
\]

Then $(S, f)$ is a $2$-polymatroid.

Consider now any algorithm which calculates $\nu(S,r)$. Plug in the $2$-polymatroid $(S,f_0)$ where

\[
\begin{align*}
f_0(X) = 2|X| & \text{if} & |X| \leq \nu, \\
2\nu + 1 & \text{if} & |X| = \nu + 1, \\
2\nu + 2 & \text{if} & |X| > \nu + 2.
\end{align*}
\]

We show that the algorithm must ask for $f_0(X)$ for every $X \subseteq S$, $|X| = \nu + 1$. For suppose that there is a subset $X_1 \subseteq S$, $|X_1| = \nu + 1$ whose rank $f_0(X_1)$ is not asked for. Then plug in the polymatroid $(S,f_1)$ where

\[
\begin{align*}
f_1(X) = 2|X| & \text{if} & |X| \leq \nu, \\
2\nu + 1 & \text{if} & |X| = \nu + 1, X \neq X_1, \\
2\nu + 2 & \text{if} & X = X_1 \text{ or } |X| > \nu + 2.
\end{align*}
\]

Then for all sets $X$ for which the algorithm has asked for $f_0(X)$, we have $f_0(X) = f_1(X)$. Therefore the algorithm does the same with $(S,f_1)$ as with $(S,f_0)$ and reaches the same conclusion. This is impossible since $\nu(S,f_0) = \nu$ while $\nu(S,f_1) = \nu + 1$.

So the algorithm must ask for all $\binom{n}{\nu + 1}$ values $f_0(X)$, $|X| = \nu + 1$. If we let here $\nu = \left\lfloor \frac{n-1}{2} \right\rfloor$ then

\[
\binom{n}{\nu + 1} > (2 - \epsilon)^n.
\]

We remark that for $|S| = 4$ and $\nu = 1$, the polymatroid $(S,f_1)$ arises in a natural way from the so-called Vamos–Higgs-matroid, which is one of the best known examples of non-linear matroids.

3. A POLYNOMIAL-BOUNDED ALGORITHM FOR LINEAR 2-POLYMATROIDS

In this paragraph we consider only $2$-polymatroids. Accordingly, whenever a $(\leq 2)$-polymatroid arises, we delete all elements of rank $0$ or $1$. It would not be difficult to extend our considerations to all $(\leq 2)$-polymatroids, but the formulation would become somewhat awkward. We shall call the elements of the polymatroids lines. This may help the reader to visualize our arguments as two special cases: lines in a projective space and edges (lines) of a graph.

We shall assume that we are able to compute ranks fast; also, in the part when we assume that our polymatroid is linear, we shall use simple manipulations with linear subspaces, like forming intersection or span etc. Efficient algorithms of linear algebra are available for this purpose and we shall not go into details.

The general setting is the following. We shall have a polymatroid (in some steps we have to assume it is linear) and a $\nu$-element matching. We shall try to find a $(\nu+1)$-element matching or a proof that no $(\nu+1)$-element matching exists. In some cases, however, we shall be content with finding a non-trivial $\nu$-double flower. Non-trivial $\nu$-double flowers relatively easily produce $(\nu+1)$-element matchings by algorithm (3.3) below.

The presentation of the algorithm is broken down into parts. The first two algorithms are of auxiliary character. The second contains the crucial point where representability enters our discussion. We shall sketch after some steps how they would specialize in the case of graphs.
Algorithm 3.1.

Input:

1° A 2-polymatroid \((S, r)\).

2° A set \(\varphi\) of \(\nu\)-flowers in \(S\). The union of their circuits is denoted by \(V\).

3° A \(\nu\)-element matching \(B\) and another matching \(M \subseteq V\).

Output: One of the following:

(a) A \((\nu + 1)\)-element matching in \((S, r)\).

(b) A \(\nu\)-flower whose circuit meets both \(V\) and \(S - V\).

(c) Two matchings \(B', M'\) such that

\[
\begin{align*}
|B'| &= \nu, \\
|M'| &= |M|, \\
|B' \cap V| &= |B \cap V|, \\
M' &\subseteq V
\end{align*}
\]

and, moreover, \(B' \supseteq M'\).

Description. We show that we can either reach one of the conclusions \((a), (b), (c)\), or

(d) replace \(M\) and \(B\) by two other matchings \(M'\) and \(B'\) satisfying \((*)\) and such that \(|M' \cup B'| < |M \cup B|\).

It is clear that after at most \(\nu\) iterations we achieve one of \((a), (b)\) or \((c)\).

Case 1. \(r(M \cup B) > 2\nu\).

Select a line \(f \in M\) with \(r(B + f) > r(B) + 1\). If \(r(B + f) = r(B) + 2\) then \(B + f\) is a \((\nu + 1)\)-element matching. Suppose that \(r(B + f) = r(B) + 1\). Let \(C\) be the circuit in \(B + f\). If \(C\) meets both \(V\) and \(S - V\) then \((b)\) is achieved. Otherwise, select a line \(g \in C - M\) and replace \(B\) by \(B - g + f\). Thus \((d)\) is achieved.

Case 2. \(r(B \cup M) = r(B)\). If \(M \subseteq B\) then \((c)\) is achieved, so suppose

that there is a line \(e \in M - B\). Since \(e \in V\), there exists a \(\nu\)-flower \(F \in \varphi\) such that the circuit \(C\) of \(F\) contains \(e\). We show that we can achieve one of \((a), (b), (c), (d)\) or

(e) replace \(B\) and \(M\) by two matchings \(B'\) and \(M'\) such that \((<\ast>\) is satisfied, \(|B' \cup M'| = |B \cup M|\) and \(|B' \cup F| < |B \cup F|\).

It is clear that in at most \(\nu\) iterations one of \((a), (b), (c), (d)\) or \((e)\) is achieved.

Since \(r(F) > r(B)\), there is a line \(f \in F\) with \(r(B + f) > r(B) + 1\).

If \(r(B + f) > r(B) + 2\) then \((a)\) is achieved. So suppose \(r(B + f) = r(B) + 1\). Let \(C_1\) denote the circuit in the \(\nu\)-flower \(B + f\). Note that \(e \not\in C_1\), hence \(C_1 \not\subseteq C\) and therefore \(C_1 \not\subseteq F\). Select a line \(h \in C_1 - F\) and set \(B' = B + f - h\).

If \(C_1\) intersects both of \(V\) and \(S - V\) then \((b)\) is achieved, if \(C_1 \subseteq S - V\) then \((e)\) is trivially achieved, so suppose that \(C_1 \subseteq V\).

Also note that by the submodularity of \(r\),

\[
\begin{align*}
& r(M + f) + r(B) = r(M + f) + r(B \cup M) > \\
& > r(M) + r(B \cup M + f) > r(M) + (B + f) > r(M) + r(B),
\end{align*}
\]

and hence

\[
r(M + f) > r(M).
\]

Case 2.1. \(C_1 \subseteq M + f\). Then \(M + f\) is a flower and so \(M' = M + f - h\) is a matching. Set

\[
M' = M + f - h, \quad B' = B + f - h.
\]

Clearly \((<\ast>\) is fulfilled and \(|B' \cup F| < |B \cup F|\), so \((e)\) is achieved.

Case 2.2. \(C_1 \not\subseteq M + f\). Let \(g \in C_1 - (M + f)\). If \(M + f\) is a flower, then let \(C_2\) be its circuit. By assumption \(C_1 \not\subseteq C_2\), so \(C_2 \not\subseteq B + f\) and so we may select a line \(k \in C_2 - (B + f)\). Setting

\[
B' = B + f - g, \quad M' = M + f - k,
\]
(6) is achieved. If \( M + f \) is a matching then
\[
B' = B + f - h, \quad M' = M + f - e
\]
achieves (γ).\]

Algorithm 3.2.

Input:

1° A set \( H \) of lines in a projective space.

2° An integer \( ν \geq 1 \).

3° A non-trivial \( ν \)-double-flower \( D \); we denote its kernel by \( K \).

4° A \( ν \)-element matching \( B \) such that \( K \not\subseteq \bar{B} \).

Output: A \((ν + 1)\)-element matching of \( H \).

Remark. If \((S, r)\) is graphic, this algorithm does the following. Given a 3-star with centre \( x \) and \((ν - 1)\) edges disjoint from each other and the 3-star; also given a \( ν \)-element matching not covering \( x \), find a \((ν + 1)\)-element matching. This task is rather easily solved by alternating chains. The algorithm described below specializes to this natural solution in the graphic case.

It is somewhat surprising that in the graphic case the use of this subroutine could be avoided (but not in the general case!).

Description. Select a point \( p \in K - \bar{B} \). Select a line \( e \in D \) such that \( e \not\subseteq \bar{B} + p \); since
\[
\bar{r}(\bar{B} + p) = 2ν + 1 < \bar{r}(D) = 2ν + 2,
\]
such a line exist. If \( \bar{r}(\bar{B} + e) = 2ν + 2 \) then \( B + e \) is a \((ν + 1)\)-element matching and we are finished. Suppose \( \bar{r}(\bar{B} + e) = 2ν + 1 \). Then \( B + e \) is a \( ν \)-flower. Furthermore,
\[
\bar{r}(\bar{B} + e + p) = 2ν + 2 > \bar{r}(\bar{B} + p)
\]
and hence \( p \not\subseteq \bar{B} + e \). Let \( C \) be the circuit in the \( ν \)-flower \( B + e \), then \( p \not\subseteq \bar{C} \) as \( \bar{C} \subseteq \bar{B} + e \). Hence by the choice of \( p \), \( C \not\subseteq \bar{D} \). Let \( f \in C - D \), and \( B' = B + e - f \). Then \( B' \) is a \( ν \)-element matching, \( p \in \bar{B}' \) since \( \bar{B}' \subseteq \bar{B} + e \), and
\[
(1) \quad B' \cup D \subseteq B \cup D
\]
since \( f \) occurs on the right hand side but not on the left. So put \( B = B' \) and return to the beginning. By (1) in at most \( ν \) iterations we find a \((ν + 1)\)-element matching in \( B \cup D \).

Now we come to the first important case, which again can be settled in any polymatroid. The algorithms that follow gradually relax the assumptions and solve the matching problem for more general polymatroids. They, however, make use of 3.2 and thus of the representation of the polymatroid.

Algorithm 3.3.

Input:

1° A 2-polymatroid \((S, r)\).

2° An integer \( ν \geq 1 \) such that \( r(S) \geq 2ν + 2 \).

3° A set \( ϕ \) of \( ν \)-flowers such that the circuits in \( ν \)-flowers in \( ϕ \) form the edges of a connected hypergraph \( χ \) on \( V(χ) = S \).

Output: One of the following:

(a) A \((ν + 1)\)-element matching.

(b) A non-trivial \( ν \)-double flower.

Remark. This algorithm specializes in the graphic case to an algorithmic proof of Gallai's lemma that if \( G \) is a connected graph such that no point of \( G \) is covered by all maximum matchings, then \( G \) has matching number \( \frac{1}{2} (|V(G)| - 1) \). For let \( ν \) be the matching number of \( G \) and assume indirectly that \( |V(G)| \geq 2ν + 2 \). Assume that for each \( x \in V(G) \) we are given a maximum matching \( M_x \) missing \( x \). Let \( ϕ \) consist of all \( ν \)-flowers \( M_x \cup \{e\} \), where \( e \) is an edge incident with \( x \). It is not too difficult to prove that the corresponding hypergraph (or, in
this case, graph) $\chi$ is connected. So Algorithm 3.3 results in a $(v - 1)$-element matching or a non-trivial $v$-double flower, consisting of a 3-star with centre $x$ and a $(v - 1)$-element matching. Using 3.2, we can find a $(v + 1)$-element matching in this second case too.

**Description.** Let $F_0 \in \varphi$. We show that we can achieve (a) or (b) or  

(γ) replace some $F \in \varphi$ by a $v$-flower $F'$ such that $F$ and $F'$ have the same circuit and $\text{Span } F \neq \text{Span } F_0$ but $\text{Span } F' = \text{Span } F_0$.

Clearly, in at most $|\varphi|$ iterations one of (α) or (β) is achieved.

By 2°, $S \neq \text{Span } F_0$ and so by 3°, we find a $v$-flower $F \in \varphi$ such that $\text{Span } F \neq \text{Span } F_0$. Again by 3° we can pick two $v$-flowers $F_1$ and $F_2$ such that $\text{Span } F_1 = \text{Span } F_0$ but $\text{Span } F_2 \neq \text{Span } F_0$, and the circuits $C_1$ and $C_2$ in $F_1$ and $F_2$, respectively, have an element in common.

By $\text{Span } F_1 \neq \text{Span } F_2$, there exists a line $e \in F_1 - \text{Span } F_2$. Consider $F_2 + e$. By Proposition 1.12, $F_2 + e$ contains a $(v + 1)$-element matching, or is a $v$-double flower. So we are finished, unless it is the direct sum of two circuits and a matching. Then one of these circuits is $C_2$ and the other one, say $C$, contains $e$. Since $C_1$ and $C_2$ meet but $C$ and $C_2$ do not, we have $C \notin C_1$ and hence $C \notin F_1$. Therefore we can find a line $f \in C - F_1$. Then $F'' = F_2 + e - f$ is a $v$-flower containing the same circuit as $F_2$, and, furthermore, satisfies $|F''| < |F_2|$. So replace $F_2$ by $F''$ and repeat. In at most $v$ iterations (γ) is achieved.

**Algorithm 3.4.**

**Input:**

1° A set $H$ of lines.

2° An integer $v \geq 1$.

3° A (non-empty) set $\beta$ of $v$-element matchings of $H$ such that $\bigcap \{B: B \in \beta\} = \emptyset$.

**Output:**

(α) A $(v + 1)$-element matching of $H$.

(β) A partition $H = H_1 \cup \ldots \cup H_k$ such that

$$\sum_{i=1}^{k} \left[ \frac{r(H_i)}{2} \right] \leq v$$

and thereby a proof that $H$ contains no $(v + 1)$-element matching.

**Description.** We shall build up a set $\varphi$ of $v$-flowers. The circuits of these $v$-flowers will constitute a hypergraph $\chi$. We consider every element of $H$ as a (possibly isolated) point of $\chi$. We denote the connected components of $\chi$ by $\chi_1, \ldots, \chi_k$ and set $H_i = V(\chi_i)$. Some of the $H_i$ may be singletons, and $H_1 \cup \ldots \cup H_k$ is a partition of $H$.

We begin with $\varphi = \emptyset$. Given any $\varphi$, we shall describe an algorithm which achieves one of (α), (β) or

(γ) finding a $v$-flower $F$ such that its circuit contains lines from at least two classes $H_i$.

It is clear that if (γ) is achieved and we replace $\varphi$ by $\varphi + F$ then $k$ decreases. So in not more than $|H|$ iterations we must end up with (α) or (β).

If $H$ has an isolated point $e$, then by 3° we find a $v$-element matching $B$ such that $e \notin B$. Then $B + e$ achieves either (α) or (γ). So suppose that $H$ has no isolated points.

Let us pick a $v$-element matching $B$ (say $B \in \beta$). Let $B_i = B \cap H_i \quad (i = 1, \ldots, k)$.

**Case 1.** $r(H_i) < r(B_i) + 1$ for every $i = 1, \ldots, k$.

Then

$$\sum_{i=1}^{k} \left[ \frac{r(H_i)}{2} \right] < \sum_{i=1}^{k} \left[ \frac{r(B_i) + 1}{2} \right] = \sum_{i=1}^{k} \left[ \frac{r(B_i)}{2} \right] = \sum_{i=1}^{k} |B_i| = |B| = v,$$
so we have conclusion ($\beta$).

Case 2. Let $r(H_i) \geq r(B_i) + 2$ for some $i$, say $r(H_1) \geq r(B_1) + 2$ and $|H_1| \neq 1$. For every $C \in E(H_1)$, consider the $v$-flower $F_C \in \varphi$ containing $C$, pick a line $f_C \in F$, and consider the $v$-element matching $F_C - f_C$. If

$$|(F_C - f_C) \cap H_1| < |B_1|$$

for some $C \in E(H_1)$, then apply Algorithm 3.1 with $M = B_1$, $B = F_C - f_C$, $V = H_1$, and reach conclusion ($\alpha$) or ($\gamma$) above (conclusion 3.1 ($\gamma$) is impossible since $|M'| = |M| > |B \cap V| = |B' \cap V|$). If

$$|(F_C - f_C) \cap H_1| > |B_1|,$$

then we conclude similarly. So suppose that $|(F_C - f_C) \cap H_1| = |B_1| = v_1$ for every $C \in E(H_1)$. Similarly we may suppose that $|B \cap H_1| = v_1$ for every $B \in \beta$. Set

$$\varphi_1 = \{F_C \cap H_1 : C \in E(H_1)\},$$

$$\beta_1 = \{B \cap H_1 : B \in \beta\}.$$

Then $\varphi_1$ consists of $v_1$-flowers of $H_1$ and the circuits of these $v_1$-flowers form a connected hypergraph on $H_1$. Apply Algorithm 3.3 to get a $(v_1 + 1)$-element matching of $H_1$ or a $v_1$-double-flower $D_1$ of $H_1$. In the second case, apply Algorithm 3.2 to produce a non-trivial $(v_1 + 1)$-element matching in $H_1$ from $D_1$ and an appropriate $B_1 \in \beta_1$. So we have a $(v_1 + 1)$-element matching in $H_1$. Apply again 3.1.

Case 3. $H_i = \{e\}$ for some $i$. Then let $B \in \beta$ such that $e \not\subset B$. $B + e$ achieves ($\gamma$).\]

Algorithm 3.5.

Input:

1° A set $H$ of lines.

2° An integer $v \geq 1$.

3° A set $\beta$ of $v$-element matchings and a $B_0 \in \beta$ such that

$$K = \cap \{\bar{B} : B \in \beta\}$$

contains no line of $B_0$.

Output: One of the following:

($\alpha$) A $(v + 1)$-element matching of $H$.

($\beta$) A partition $H = H_1 \cup \ldots \cup H_k$ and a flat $A \subseteq K$ such that

$$v \geq r(A) + \sum_{i=1}^k \left[ \frac{r(H_i + A) - r(A)}{2} \right],$$

(and thereby a proof of that $H$ contains no $(v + 1)$-element matching).

Remark. In the graphic case, the procedure followed here can be described as follows. We have a number $v$ and a non-empty family $\beta$ of $v$-element matchings of the graph. If $K = \cap (V(B) : B \in \beta) = \phi$ then we can conclude as in 3.4. So suppose that this intersection is non-empty, then we try to find, a $v$-element matching $B_1$ which misses at least one point of $K$. Delete the points of $K$ and look for a $(v - |K| + 1)$-element matching in the remaining graph. If it does not exist, we easily see that $G$ has matching number $v$. If we find a $(v - |K| + 1)$-element matching $M$ then $B_1$ is constructed by an alternating chain argument.

Description. We show that we can achieve ($\alpha$), ($\beta$) or ($\gamma$) find a $v$-element matching $B_1$ such that $K \not\subset \bar{B}_1$.

Adding $B_1$ to $\beta$ and repeating the procedure at most $v$ times we must end up with ($\alpha$) or ($\beta$).

First we achieve ($\alpha$) or ($\beta$) or find a matching $M$ such that

($*$) $r(M + K) \geq |M| + v + 1$.

If $r(K) = s > v$ then we can take $M = \phi$; so suppose $s \leq v$.

Set $H' = H / K$. For each $B \in \beta$, let $B^*$ be a $(v - s)$-element matching in $B / K$ (this exists, and can be constructed, by Proposition 1.6). Let
\[ \beta' = \{ B^*: B \in \beta \}. \]

Then clearly
\[ \bigcap \{ \tilde{U}; U \in \beta' \} = \phi. \]

Applying Algorithm 3.5 we either get a \((v - s + 1)\)-element matching in \(H'\) or a partition \(H' = H'_1 \cup \ldots \cup H'_k\) such that
\[ \sum_{i=1}^{k} \left[ \frac{r(H'_i)}{2} \right] \leq v - s. \]

In the latter case we simply denote by \(H_i\) the subset of \(H\) corresponding to \(H'_i\), we denote by \(H_{k+1}, \ldots, H_t\) the singleton sets consisting of elements of \(H\) which meet \(K\) (and, therefore, do not correspond to any member of \(H'\)), and set \(A = K\). Then
\[ v \geq s + \sum_{i=1}^{k} \left[ \frac{r(H'_i)}{2} \right] = r(A) + \sum_{i=1}^{k} \left[ \frac{r(H'_i + A) - r(A)}{2} \right] = \]
\[ = r(A) + \sum_{i=1}^{k} \left[ \frac{r(H'_i + A) - r(A)}{2} \right]. \]

Thus \((\beta)\) is achieved.

So suppose that when applying Algorithm 3.5 to \(H'\) we end up with a \((v - s + 1)\)-element matching \(M\) in \(H\) such that \(\tilde{M} \cap K = \phi\). Then \(M\) satisfies \((*)\).

We show now that if we have a matching \(M\) satisfying \((*)\) then we can achieve \((\alpha)\), \((\beta)\), \((\gamma)\) or \((\delta)\) find a matching \(M'\) satisfying \((*)\) such that
\[ |M' \cap B_0| > |M \cap B_0|. \]
After at most \(v\) iterations we clearly achieve \((\alpha)\), \((\beta)\) or \((\gamma)\).

Clearly if \(|M| \geq v\) then either \((\alpha)\) or \((\gamma)\) is achieved, so suppose that \(|M| < v\). If there is a line \(e \in B_0\) such that \(e \not\subset \tilde{M} + K\) then
\[ r(M + e + K) > |M + e| + v + 1. \]

If \(M + e\) is a matching then we may set \(M + e\). If \(M + e\) is a flower then let \(C\) be its circuit. Since \(C \not\subset B_0\) we may choose a line \(f \in C - B_0\). Now we can put \(M' = M - f + e\).

If
\[ r(M + K) > |M| + v + 1 \]
then choose any \(e \in B_0\) such that \(e \not\subset \tilde{M}\). Then
\[ r(M + e + K) > |M + e| + v + 1 \]
and we conclude as before.

So we may assume that every \(e \in B_0\) is contained in \(\tilde{M} + K\) and also that \(r(M + K) = |M| + v + 1\). Hence \(|M| = v - 1\) and \(\tilde{M} + K = \tilde{B}_0\).

Let \(e \in B_0\) be such that \(e \not\subset \tilde{M}\). Let \(B_1 \in \beta\) be a matching such that \(e \not\subset \tilde{B}_1\). We show that we can achieve \((\alpha)\), \((\beta)\), \((\gamma)\) or \((\delta)\) find a matching \(M'\) satisfying \((*)\) such that
\[ |M' \cap B_0| > |M \cap B_0|. \]
After at most \(v\) iterations we clearly achieve \((\alpha)\), \((\beta)\) or \((\gamma)\).

Since \(\tilde{B}_1 \not\subset \tilde{B}_0\), \(\tilde{M} + K\), we can find a line \(f \in B_1\) such that \(f \not\subset \tilde{M}\). Consider \(M + f\). If this is a matching then \((\gamma)\) is achieved. So suppose \(M + f\) is a \((v - 1)\)-flower. Let \(C\) be its circuit, and let \(g \in C - B_1\). If \(g \not\subset B_0\) then set
\[ M^* = M - g + f, \quad B^*_0 = B_0. \]
If \(g \in B_0\) then set
\[ M^* = M - g + f, \quad B^*_0 = B - g + f. \]
In both cases we are finished.
Algorithm 3.6.

Input:

1° A set $H$ of lines.
2° A matching $B_0 \subseteq H$.

Output: One of the following:

(α) A matching in $H$ larger than $B_0$.

(β) A flat $A$ and a partition $H_1 \cup \ldots \cup H_k = H$ such that

$$|B_0| = r(A) + \sum_{i=1}^{k} \left[ \frac{r(H_i + A) - r(H_i)}{2} \right]$$

(and thereby a proof of that $B_0$ is a maximum matching).

Description. We form a set $\beta$ of $\nu$-element matchings ($\nu = |B_0|$).

We set

$$K = \bigcap \{ \bar{B} : B \in \beta \}.$$  

We start with $\beta = \{ B_0 \}$ and show that we can achieve (α), (β) or (γ).

(γ) find a $\nu$-element matching $B_1$ such that $K \notin \bar{B}_1$.

Trivially, in at most $2\nu$ iterations, we must achieve (α) or (β).

Clearly $K \subseteq \bar{B}_0$. Let

$$Y = \{ e \in B_0 : e \subseteq K \}.$$

If $Y = \emptyset$ then we can apply Algorithm 3.5 and we are finished. So suppose $Y \neq \emptyset$. Clearly $\bar{Y} \subseteq K$. Choose a point $p \in \bar{Y}$ such that $p \notin \bar{Y} - \bar{y}$ for each $y \in Y$. Thus $B_0 / p$ is a $(\nu - 1)$-flower whose circuit is $Y$. For each $B \in \beta$ select a line $b$ such that $p \notin \bar{B} - b$ and let $B' = (B - b) / p$. Also let $B_p = (B_0 - y) / p$ ($y \in Y$). Clearly $B_p$ is a $(\nu - 1)$-matching as well as $B'$ for all $B \in \beta$. Set

$$\beta' = \{ B_p : y \in Y \} \cup \{ B' : B \in \beta \},$$

and

$$K' = \bigcap \{ \bar{B} : B \in \beta' \}.$$  

Note that $K' \subseteq K / p$. Select a $y_0 \in Y$ and set $B'_0 = B_{y_0}$. We claim that $K'$ contains no line of $B'_0$. For let $e' \in B'_0$ and let $e$ denote the corresponding line in $B_0$. If $e \in Y$ then, since $B_0 / p$ is a flower, $e' \notin (\bar{B}_0 - e) / p = \bar{B}_e$ and so $e' \notin K'$. If $e \in B_0 - Y$ then $e \notin K$ and so, since $p \in K$, we can again conclude that $e' \notin K'$.

So if we pick a $y \in Y$, then $(H', \nu, -1, \beta, B_y)$ satisfy the conditions under which Algorithm 3.5 can be applied. The algorithm yields one of two things. Either a $\nu$-element matching $M'$ of $H'$ or a partition $H' = H'_1 \cup \ldots \cup H'_k$ and a flat $A'$ such that

$$\nu - 1 > r(A') + \sum_{i=1}^{k} \left[ \frac{r(H'_i + A') - r(A')}{} \right].$$

In the first case, the corresponding subset $M \subseteq H$ is a $\nu$-element matching such that $p \notin M$ and so (γ) is achieved. In the second case we set

$$A = A' + p,$$

and let $H_1$ be the subset of $H$ corresponding to $H'_1$. Also let $H_{k+1}, \ldots, H_m$ be the lines of $H$ through $p$ (which, therefore, do not correspond to lines in $H'$), as singleton sets. Then

$$\nu = r(A) = \sum_{i=1}^{m} \left[ \frac{r(H'_i + A) - r(A)}{2} \right]$$

and so (β) is achieved.

4. CONCLUDING REMARKS: APPLICATIONS, EXTENSIONS

4.1. The matching problem for polymatroids is not only a common generalization of the matching problem for graphs and the matroid parity problem, but also of a number of other graph-theoretical and combinatorial problems, where the translation to polymatroids is less apparent. These problems include selecting a maximum circuit-free subhypergraph of a 3-uniform hypergraph; packing openly disjoint paths in a graph such that the number of paths ending at a certain point is bounded; or finding a minimum number of points in a flexible planar structure whose pinning
down fixes the structure. However, we shall not elaborate these applications in this paper.

4.2. It is certain that the matching problem can be solved for a class of polymatroids larger than the linear ones. For example, we know that it can be solved for the case when the rank function of the polymatroid is the sum of the rank functions of two matroids (matroid intersection problem).

It may serve as an apology for restricting ourselves to the linear case that in the above applications the polymatroids which arise are linear. The reason for that is that we very seldom encounter matroids other than those arising from free matroids by the operations of sum, dual, series and parallel extensions, principal extensions, upper and lower truncations, restriction and contraction, and these operations always yield matroids representable over, say, the real field.

A possible common generalization of the matroid intersection and linear 2-polymatroid matching algorithms would be an algorithm which, when applied to a general 2-polymatroid, either solves the problem or detects a minor isomorphic, say, to the Vámos — Higgs-2-polymatroid mentioned in Section 2. But I do not even know how to formulate this requirement exactly.

4.3. Another generalization of the matching algorithm is the recent algorithm by Minty [8], which finds a maximum stable set in a graph containing no induced 3-star (claw).

Although this generalization is quite different from the 2-polymatroid matching problem, some common features are apparent. Any matching can be transformed into an optimum solution by exchanges carried out one at a time, by Algorithm 3.1. Similar property holds for stable sets in claw-free graphs. Given a matching $M$, the rest of the 2-polymatroid decomposes into 3 sets $A_0, A_1, A_2$, such that the elements of $A_2$ can be added to $M$ to form a larger matching, the elements of $A_1$ can be added to $M$ at the expense of deleting an element of $M$, and we know that $A_0$ spans no matching larger than $M$. Again similar property holds true for stable sets in claw-free graphs. So it is natural to ask if there is any common generalization of Minty’s algorithm and the algorithm presented in this paper.

REFERENCES


L. Lovász
Bolyai Institute, University of Szeged, Hungary.