Dependent k-Set Packing on Polynomoids

Meng-Tsung Tsai
Institute of Information Science, Academia Sinica, Taipei, Taiwan

Shi-Chun Tsai
Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

Tsung-Ta Wu
Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

Abstract

Specialized hereditary systems, e.g., matroids, are known to have many applications in algorithm design. We define a new notion called ***d*-polynomoid** as a hereditary system $(E, \mathcal{F} \subseteq 2^E)$ so that every two maximal sets in $\mathcal{F}$ have less than d elements in common. We study the problem that, given a d-polynomoid $(E, \mathcal{F})$, asks if the ground set E contains ℓ disjoint k-subsets that are not in $\mathcal{F}$, and obtain a complexity trichotomy result for all pairs of $k \geq 1$ and $d \geq 0$. Our algorithmic result yields a sufficient and necessary condition that decides whether each hypergraph in some classes of r-uniform hypergraphs has a perfect matching, which has a number of algorithmic applications.

2012 ACM Subject Classification Mathematics of computing → Matroids and greedoids

Keywords and phrases Hereditary Systems, Hypergraph Matchings, Compleixty Trichotomy

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.84

Funding This research was supported in part by the National Science and Technology Council under contract NSTC grants 108-2221-E-009-051-MY3, 109-2221-E-001-025-MY3, and 112-2221-E-001-007.

Acknowledgements We sincerely thank anonymous reviewers for their helpful comments.

1 Introduction

Finding the conditions that decide whether an r-uniform hypergraph H contains a perfect matching has received much attention. Some conditions are based on the minimum degree of a vertex in H [14, 34, 35], and some are based on the minimum degree of a set of $r-1$ vertices in H [38]. More conditions are known for bipartite hypergraphs, such as Hall's [24], Aharoni's [1, 3], and Haxell's [27], and multipartite hypergraphs [11, 2]. Because finding a maximum matching for r-uniform hypergraphs with $r \geq 3$ is APX-complete [32, 29], any computationally efficient conditions to decide whether an r-uniform hypergraph with $r \geq 3$ contains a perfect matching cannot be both sufficient and necessary unless P = NP. Indeed, all the conditions above except Hall's are sufficient but not necessary. In the literature, a number of polynomial-time algorithms to compute perfect matchings for dense r-uniform hypergraphs are known [33, 26, 25].

In this paper, we give a sufficient and necessary condition that, for any pair of integers $k > d \geq 0$, the k-uniform hypergraph

$$H = \left(V, E = \left\{e \in \binom{V}{k} : e \not\subseteq S_i \text{ for all } i \in [m]\right\}\right)$$

has a perfect matching, where $\binom{V}{k}$ denotes the collection of all k-subsets of V and $S_1, S_2, \ldots, S_m$ are subsets of V with $|S_i \cap S_j| < d$ for all $i \neq j \in [m] := \{1, 2, \ldots, m\}$. We prove also the hardness of finding a maximum matching for such hypergraphs when $k \leq d$. Combining the above, we obtain a complexity trichotomy for our problem for all pairs of k and d, detailed in Theorem 2.

48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023).
Editors: Jérôme Leroux, Sylvain Lombardy, and David Peleg; Article No. 84; pp. 84:1–84:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

To better understand the structure of the hypergraphs defined above and compare our results with related work, we restate finding a maximum matching for such hypergraphs as the dependent k-set packing on a kind of hereditary systems that we call ***polynomoids***, defined in Definition 1. Our notation for hereditary systems follows West [42].

▶ **Definition 1** (d-Polynomoid). *Let $d \geq 0$ be an integer. Let $(E, \mathcal{F})$ be a tuple where E is a finite set and $\mathcal{F} \subseteq 2^E$ is a non-empty collection of some subsets of E. The sets in $\mathcal{F}$ are called* ***independent sets****, and the other subsets of E are* ***dependent sets****. We say that $P = (E, \mathcal{F})$ is a* ***d-polynomoid*** *if P satisfies the following two properties:*

- *Hereditary Property: For every $B \in \mathcal{F}$, if $A \subseteq B$ then $A \in \mathcal{F}$.*
- *Join Property: For every $A, B \in \mathcal{F}$, if $|A \cap B| \geq d$ then $A \cup B \in \mathcal{F}$.*

If the join property is removed, then $P = (E, \mathcal{F})$ is a hereditary system [42]; if the join property is replaced with the exchange property, then P is a matroid [37].

Given a d-polynomoid $P = (E, \mathcal{F})$ and integers $k \geq 1, \ell \geq 0$, the ***dependent k-set packing*** for P asks if there exist ℓ pairwise disjoint ***k-sets*** not in $\mathcal{F}$, where a set is called k-set if it consists of k elements. If the ℓ disjoint k-sets exist, then output them. Our main result is a complexity trichotomy for the dependent k-set packing problem on polynomoids, stated formally in Theorem 2. An illustration is depicted in Figure 1.

▶ **Theorem 2.** *The time complexities of the dependent k-set packing on d-polynomoids for all pairs of integers $k \geq 1, d \geq 0$ can be classified into the following three categories:*

1. *If $k \leq d$ and $k \geq 3$, there exists a d-polynomoid P so that the dependent k-set packing for P is* APX*-complete.*
2. *If $k \leq d$ and $k \leq 2$, then:*
   - *for $k = 2$, there exists a d-polynomoid P so that the dependent k-set packing for P is as hard as the matching problem on ordinary graphs (i.e. 2-uniform hypergraphs);*
   - *for $k = 1$, this is a degenerate case solvable in $O(|E|q(1))$ time, where q is a function defined below.*
3. *Otherwise $k > d$, for any d-polynomoid $P = (E, \mathcal{F})$,*

$$E \text{ contains } \lfloor |E|/k \rfloor \text{ disjoint dependent } k\text{-sets if and only if } r(E) \leq (1 - 1/k)|E|,$$

where $r(E)$ is the size of a maximum independent subset of E. The dependent k-set packing can be found in $O(k|E|q(2k))$ time, where $q(t)$[1] denotes a monotone function that upper-bounds the running time to test whether a t-subset of E is independent.

It may be worth noting that for some polynomoids computing $r(E)$ requires quadratic time unless the 3SUM conjecture fails, as shown in Section 2.2. To obtain Theorem 2, it suffices to test $r(E) \leq (1 - 1/k)|E|$ without computing the exact value of $r(E)$, which can be tested in deterministic linear time (Section 4.1). In addition, greedily grouping the elements in a largest independent set with those from other independent sets in general cannot work correctly because the condition $r(E) \leq (1 - 1/k)|E|$ may be violated in the residual polynomoid obtained from the initial polynomoid with a greedy removal of the elements in a largest independent set and the corresponding grouped elements from other independent sets.

In addition to the complexity trichotomy result, Theorem 2 can also be used to yield a sufficient and necessary condition for each polynomoid when a perfect packing exists, such as Corollary 3 (also follows from [14]), Corollary 4, Corollary 22, and Corollary 24 (also proven in [7]).

[1] Indeed, it has to be written as $q_P(t)$ because it varies among different polynomoids. We suppress the subscript P when the context is clear.

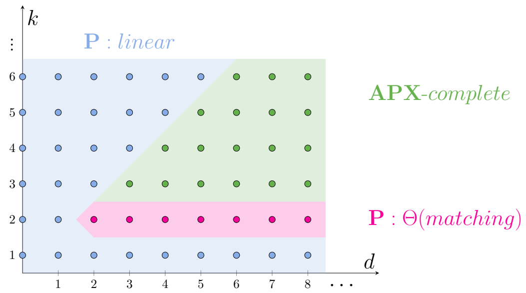

Figure 1 A complexity trichotomy for the dependent k-set packing.

▶ **Corollary 3.** *Let G be a 3-uniform hypergraph with at least four vertices. If the number of vertices in G is a multiple of 3 and every two hyperedges in G have at most one vertex in common, then the complement of G has a perfect matching.*

Proof. Let $P = (E, \mathcal{F})$ where E is the set of vertices in G and $\mathcal{F}$ is the set of hyperedges in G. By definition, P is a 2-polynomoid. By setting $(k, d) = (3, 2)$ and applying Theorem 2 on P, we are done. ◀

We remark that a number of related works discuss the independent set partition for hereditary systems [20, 43], and the minimal dependent set packing [39] and partition [17, 41, 31, 13, 21] for matroids. Note that partition problems can be reduced to packing problems.

1.1 Example Polynomoids

There are a number of structures that satisfy the requirements of polynomoids. We list three of them below, and more can be found in Appendix B.

1. Let E be a finite set of points in $\mathbb{R}^2$ and

$$\mathcal{F} = \{E' \subseteq E : \text{ all points in } E' \text{ are colinear}\}.$$

$P = (E, \mathcal{F})$ is a 2-polynomoid because:

- Hereditary Property: Let S be a point set in $\mathbb{R}^2$. If all points in S are colinear, then all points in any subset of S also are colinear.
- Join Property: Let S_1, S_2 be two point sets in $\mathbb{R}^2$. If all points in each of S_1 and S_2 are colinear and $|S_1 \cap S_2| \geq 2$, then all points in $S_1 \cup S_2$ are colinear.

The dependent k-set packing problem for P asks if E contains ℓ disjoint k-sets so that the points in each k-set are not colinear. In particular, for $k = 3$, the points in each k-set are on a circle with a finite radius, as depicted in Figure 2a. By Theorem 2, this problem can be solved in $O(|E|)$ time as $k, q(2k) = O(1)$.

▶ Remark. More generally, the above example can be generalized to any set of degree-d univariate polynomials for any $d \geq 1$. Let E denote a finite set of distinct points in $\mathbb{R}^2$. We may need a rotation of axes to ensure that no two points in E have the same x-coordinate. Let $\mathcal{L}$ denote a collection of polynomials with degree d. Let $\mathcal{F}$ denote the collection of all the subsets E' of E that some polynomial in $\mathcal{L}$ passes through all points in E'. It is not hard to check that such an $(E, \mathcal{F})$ is a d-polynomoid. This motivates us to call the hereditary systems defined in Definition 1 ***polynomoids***.

2. Let $G = (V, E)$ be an undirected simple graph and

$$\mathcal{F} = \{E' \subseteq E : \text{all edges in } E' \text{ have a vertex in common}\}.$$

Hence, for any $A \in \mathcal{F}$, the subgraph of G induced by the edges in A is a star graph. $P = (E, \mathcal{F})$ is a 2-polynomoid because:

- Hereditary Property: Any subgraph of a star graph also is a star.
- Join Property: If two star subgraphs have at least two edges in common, then their union also is a star.

The dependent k-set packing problem for P asks if E contains ℓ disjoint k-sets so that the edges in each k-set do not form a star graph. In particular, for $k = 3$, the edges in each k-set edge-induce a triangle or a ***linear forest***, i.e. each component in the forest is a path, as depicted in Figure 2b. Because the union of the edges in a triangle and those in any 3-edge non-star graph can be partitioned into two 3-edge linear forests (Lemma 25), the problem of partitioning the edges in a graph into 3-edge linear forests is linear-time reducible to the dependent k-set packing for P. By Theorem 2, both problems can be solved in $O(|E|)$ time as $k, q(2k) = O(1)$.

▶ Remark. It is shown in [9, 6, 4] that an m-edge undirected simple graphs with maximum degree Δ, except for a finite number of exceptions, can be edge-partitioned into $3P_2$s, i.e., three vertex-disjoint edges if and only if m is a multiple of 3 and $\Delta \leq m/3$. By Theorem 2 and the above discussion, we obtain an analogous result that:

▶ **Corollary 4.** *An m-edge undirected simple graph with maximum degree Δ can be edge-partitioned into 3-edge linear forests if and only if m is a multiple of 3 and $\Delta \leq 2m/3$.*

Let $\{H_1, H_2, \ldots, H_t\}$-decomposition be the problem that, given an undireted simple graph $G = (V, E)$, decide whether E can partitioned into subsets so that each subset edge-induce a subgraph isomorphic to H_i for some $i \in [t]$. It is conjectured in [40] that $\{H_1, H_2, \ldots, H_t\}$-decomposition is NP-complete if and only if $\{H_i\}$-decomposition is NP-complete for all $i \in [t]$. Let P_ℓ be a path of ℓ nodes. Let $P_i \cup P_j$ be the union of vertex-disjoint P_i and P_j and let kP_i be the union of k vertex-disjoint P_is. By the above conjecture and the fact that P_4-decomposition is NP-complete [28, 15] but $P_3 \cup P_2$-decomposition [18, 10] and $3P_2$-decomposition [9, 6, 4] are polynomial-time solvable, partitioning the edge set of an undirected simple graph into 3-edge linear forests shall (assuming the conjecture holds) be solvable in polynomial time. Our above linear-time algorithm gives an example that supports the conjecture.

3. Let $G = (V, A)$ be an edge-weighted directed graph and

$$\mathcal{F}_\Delta = \{V' \subseteq V : \text{ a minimum } st\text{-cut in } G \text{ has weight at least } \Delta \text{ for every } s \neq t \in V'\},$$

where an st-cut is a partition of V into two disjoint sets S and T with $s \in S, t \in T$ and the weight of an st-cut is defined to be the sum of weights on the directed edges from S to T. Note that by definition $\mathcal{F}_\Delta$ contains $\emptyset$ and all singleton sets. $P = (V, \mathcal{F}_\Delta)$ is a 1-polynomoid because:

- Hereditary Property: If a minimum st-cut in G has weight at least Δ for all pairs of $s \neq t \in V'$, then it also applies for all pairs of vertices in a subset of V'.
- Join Property: Let $A, B \in \mathcal{F}_\Delta$ and z be a vertex that A, B have in common. For any pair of $s \in A, t \in B$, a minimum st-cut (S, T) that separates s, t has either $z \in S$ or $z \in T$. We assume w.l.o.g. that $z \in S$. Since a minimum tz-cut and a minimum zt-cut both have weights at least Δ, the (S, T)-cut also has weight at least Δ.

The dependent k-set packing problem for P asks if V contains ℓ disjoint k-sets so that for each k-set there is a cut in G of weight less than Δ that separates the vertices in it, as depicted in Figure 2c. By Theorem 2, this problem can be solved in $O(k|V|k^2 f(G))$ time as $q(2k) = O(k^2 f(G))$ where $f(G)$ denotes the running time of exact maxflow computation between two distinct nodes s and t on G.

▶ Remark. A naive approach for this problem needs to compute the minimum st-cuts for all pairs of $s, t \in V$, but ours needs only $O(k^3 V) = O(V)$ pairs. For undirected graphs, by Gomory-Hu trees [23] the number of the minimum st-cuts that are needed to compute is also $O(V)$. For unweighted directed graphs, the running time also can be reduced by the approach in [12]. For weighted directed graphs, to the best of our knowledge, our algorithm is the first one with running time matching the current best algorithm for weighted undirected graphs.

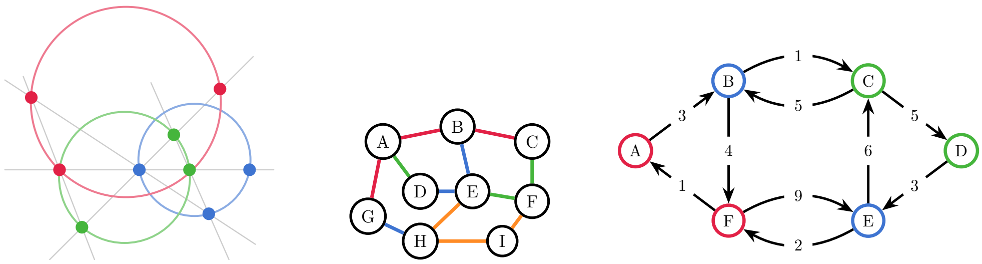

Figure 2 (a) The figure to the left: a partition of the given points into 3-sets so that the points in each 3-set are on a circle with a finite radius. (b) The figure in the middle: a partition of the edges into 3-sets so that the edges in each 3-set induces a linear forest. (c) The figure to the right: a partition of the vertices into 2-sets so that for each 2-set there is a cut in G has weight less than 7 that separates vertices in it.

1.2 Sharpness of Our Result

The two properties of polynomoids are essential to make the dependent k-set packing for polynomoids solvable in linear time for $k > d$.

- Case I: if the hereditary property is removed from the Definition 1, then we have an example problem for $k > d$ that cannot be solved in polynomial time unless P = NP. Let G be an undirected graph. Let $(E, \mathcal{F})$ be a set system where E denotes the vertex set of G and $\mathcal{F}$ consists of all the subsets E' of E so that the subgraph of G induced by the vertices in E' are connected. Note that, for any $E_1, E_2 \in \mathcal{F}$, if $|E_1 \cap E_2| \geq 1$, then $E_1 \cup E_2 \in \mathcal{F}$. Thus, this set system corresponds to the case $d = 1$.

  Set $k = 3$, so $k > d$. To find a dependent k-set packing for $(E, \mathcal{F})$, it is equivalent to asking whether E contains ℓ disjoint k-sets so that the subgraph in G induced by the vertices in each k-subset is disconnected. For $k = 3$, this problem is equivalent to asking whether E contains ℓ disjoint k-sets so that the subgraph in $\bar{G}$ (the complement graph of G) induced by the vertices in each k-set is connected, which is known to be NP-complete [16].
- Case II: if the join property is removed from the Definition 1, then the parameter d is removed. So the complexity trichotomy in Figure 1 collapses. For $k \geq 3$, then it is APX-complete; for $k = 2$, then it is as hard as matching [36]. It is not known how to solve either case in linear time.

1.3 Paper Organization

In Section 2, we devise a polynomial-time algorithm to find a maximum independent set for any polynomoid P, assuming that the independence oracle can be queried in polynomial time, and prove that no algorithms can solve this problem in truly subquadratic time unless the 3SUM conjecture fails. Then, in Section 3, we relate the problem of finding a maximum independent set and that of finding a largest dependent k-set packing for any d-polynomoids with $k > d$. Because finding a maximum independent set for polynomoids is 3SUM-hard in general, we devise a deterministic linear-time algorithm without involving the exact computation of the maximum independent sets in Section 4. We prove the hardness for the case of $k \leq d$ in Appendix A. Finally, we present more applications of our results in Appendix B and place omitted proofs in Appendix C.

2 Maximum Independent Sets

For any matroid, finding a maximum independent set can be done greedily because all maximal independent sets have equal size. Since maximal independent sets of a polynomoid may have different sizes, the greedy approach for matroids cannot be applied to polynomoids. In what follows, we devise a polynomial-time algorithm to find a maximum independent set for any polynomoid, assuming that testing whether a set is independent can be done in polynomial time. In addition, we prove that this problem cannot be solved by any truly subquadratic-time algorithm even if the independence oracle can be decided in time linear in the input size unless the 3SUM conjecture fails.

2.1 A Polynomial-Time Algorithm

Our polynomial-time algorithm for finding a maximum independent set is mainly based on the following key lemma.

▶ **Lemma 5.** *Let $P = (E, \mathcal{F})$ be a d-polynomoid for some $d \geq 0$. For any d-subset C of E, precisely one of the following two statements holds:*

- *No maximal independent sets in P contain C.*
- *Exactly one maximal independent set in P contains C.*

Proof. If C is dependent, then no independent set contains C due to the hereditary property of P. If C is independent and there exist two distinct maximal independent sets M_1, M_2 of P that contain C as a subset, then

$$|M_1 \cap M_2| \geq |C| = d.$$

By the join property of P, $M_1 \cup M_2$ is independent. Since $M_1 \neq M_2$ and they are maximal, we have

$$|M_1 \cup M_2| \geq \max\{|M_1|, |M_2|\} + 1,$$

contradicting the maximality of M_1 and M_2. Therefore, precisely one maximal independent set contains C. Each of the above two cases corresponds to one of the claimed statements. ◀

Lemma 5 yields an efficient algorithm that, for any independent set C of a d-polynomoid with size at least d, finds “the” maximal independent set containing C as a subset. Formally, we state it in Corollary 6.

▶ **Corollary 6.** *Let $P = (E, \mathcal{F})$ be a d-polynomoid for some $d \geq 0$. For any $C \in \mathcal{F}$ with size at least d, finding the maximal independent set M_C that contains C as a subset can be done in $O(|E|q(d+1))$ time.*

Proof. By Lemma 5, there is a unique independent set M_C that contains any d-subset C_d of C as a subset. Hence, for any $x \in E$, if $C_d \cup \{x\} \in \mathcal{F}$, then $x \in M_C$. Testing whether $C_d \cup \{x\} \in \mathcal{F}$ for all $x \in E$ can be realized by invoking the independence oracle of P on $(d+1)$-subsets of E $O(|E|)$ times. Thus, the total running time is $O(|E|q(d+1))$. ◀

Lemma 5 and Corollary 6 imply that for any polynomoid $P = (E, \mathcal{F})$ finding a maximum independent set can be done in $O(|E|^{d+1}q(d+1))$ time. This can be seen from the following two cases.

1. If the size s of maximum independent sets in P is at most d, then they can be found by invoking the independence oracle once for each subset of E that has size $\leq d$. Hence, the running time is

$$\sum_{i=0}^{d} \binom{|E|}{i} q(i) = O(|E|^d q(d)) \text{ for } |E| \geq 2 \text{ or } O(q(1)) \text{ for } |E| \leq 1.$$

2. Otherwise, there is a maximum independent set M that has size $> d$. Because $|M| > d$ and the hereditary property of P, M contains a subset M_d of size d in $\mathcal{F}$. By Lemma 5, exactly one maximal independent set W contains M_d, so $W = M$. By Corollary 6, $W = M_d \cup \{e \in E : M_d \cup \{e\} \in \mathcal{F}\}$, which can be found in $O(|E|q(d+1))$ time. Hence, the total running time is at most

$$|E|q(d+1) \cdot \binom{|E|}{d} = O(|E|^{d+1}q(d+1)).$$

The implementations of the above two cases can be unified as in the following pseudocode.

Algorithm 1 Finding a maximum independent set for polynomoids.

```
input  : a d-polynomoid P = (E, F)
output : a maximum independent set of P
A ← ∅;
foreach S ∈ {E' ⊆ E : |E'| ≤ d} do
    if S ∈ F then
        M ← a maximal independent set in F that contains S;
        if |M| > |A| then
            A ← M;
return A;
```

As a consequence, we have:

▶ **Theorem 7.** *For any d-polynomoid $P = (E, \mathcal{F})$, given an independence oracle $\mathcal{O}_{ind} : 2^E \to \{0, 1\}$ that tests whether a t-subset of E is contained in $\mathcal{F}$ in $q(t)$ time where q is a monotone function, then finding an independent set in $\mathcal{F}$ that has the largest cardinality can be done in $O(|E|^{d+1}q(d+1))$ time.*

2.2 3SUM-Hardness

We show in Theorem 8 that, unless the 3SUM conjecture fails, there exists some polynomoid $P = (E, \mathcal{F})$ so that any algorithm that finds a maximum independent set for P requires $\Omega(|E|^{2-\varepsilon})$ time for any constant $\varepsilon > 0$.

▶ **Theorem 8.** *There exists some polynomoid P whose independence oracle can be decided in time linear in the input size so that finding a maximum independent set of P is* 3SUM*-hard.*

Proof. Let $P = (E, \mathcal{F})$ be a polynomoid where E is a finite set of distinct points in $\mathbb{R}^2$ and $\mathcal{F} = \{E_/ \subseteq E \mid \text{all points in } E_/ \text{ are colinear}\}$. It is clear that P is a 2-polynomoid. A maximum independent set of P corresponds to a line in $\mathbb{R}^2$ that passes through the most number of points in E. Hence, it suffices to answer whether there exist three points in E that are colinear, which is known to be 3SUM-hard [19]. ◀

3 Dependent k-Set Packing

In this section, we will present a reduction from the dependent k-set packing to the maximum independent set.

For each polynomoid $P = (E, \mathcal{F})$, we define a *rank* function $r : 2^E \to \{0, 1, \ldots, |E|\}$ so that $r(S)$ denotes the cardinality of a largest subset of S that is contained in $\mathcal{F}$. In particular, $r(E)$ equals the size of a maximum independent set of P, which can be computed in polynomial time by Theorem 7. More generally, for every $S \subseteq E$, $r(S)$ equals the size of a maximum independent set of the polynomoid $Q = (S, 2^S \cap \mathcal{F})$. Hence, the rank function r for any subset of E is computable in polynomial time, assuming that the independence oracle of P can be decided in polynomial time. Let $\rho_k(E)$ denote the maximum number of disjoint dependent k-subsets that E contains. We claim that $r(E)$ and $\rho_k(E)$ can be related as follows, so $\rho_k(E)$ can be computed no slower than finding $r(E)$.

▶ **Theorem 9.** *Let k, d be two integers with $k > d \geq 0$. For any d-polynomoid $P = (E, \mathcal{F})$,*

$$\rho_k(E) = \begin{cases} |E| - r(E) & \text{if } r(E) > (1 - 1/k)|E| \\ \lfloor |E|/k \rfloor & \text{otherwise} \end{cases}$$

It suffices to prove Theorem 9 for $k = d+1$ because of the observation that a d-polynomoid is also a t-polynomoid for every $t > d$.

3.1 Case I: $\mathbf{r(E) > (1 - 1/k)|E|}$

We prove the first case of Theorem 9 by the following lemma.

▶ **Lemma 10.** *Let $d \geq 0$ and $k = d + 1$ be two integers. For any d-polynomoid $P = (E, \mathcal{F})$, if $r(E) > (1 - 1/k)|E|$, then $\rho_k(E) = |E| - r(E)$.*

Proof. The proof for $d = 0$ is clear because any 0-polynomoid has $\mathcal{F} = 2^S$ for some $S \subseteq E$. Hence we assume that $d \geq 1$, so $k \geq 2$. Let M be a maximum independent set of P. By definition, $|M| = r(E)$. For each element x in $E \setminus M$, we remove $k - 1$ distinct elements from M and let A_x be the set containing x and the $k - 1$ removed elements.

▷ Claim 11. A_x is dependent.

Proof. Suppose for contradiction that $A_x \in \mathcal{F}$, the intersection of A_x and M is $k - 1 = d$, so $A_x \cup M \in \mathcal{F}$ by the join property of P. This violates the maximality of M. Hence, A_x is dependent. ◁

Since $r(E) > (1 - 1/k)|E|$, we have $r(E) > (k - 1)(|E| - r(E))$. So the above grouping procedure can iterate until $E \setminus M$ is exhausted. Hence, we obtain a collection of $|E| - r(E)$ dependent k-sets (not necessarily the largest one), so $\rho_k(E) \geq |E| - r(E)$. Note that, if $\rho_k(E) > |E| - r(E)$, by the pigeonhole principle, at least one of the $\rho_k(E)$ k-sets contains elements only from M. By the hereditary property of P, such a k-set must be independent because it is a subset of M, a contradiction. As a result, $\rho_k(E) = |E| - r(E)$. ◀

3.2 Case II: $\mathbf{r(E) \le (1-1/k)|E|}$

We prove the second case of Theorem 9 by the following lemmas. We will show that, if $r(E) \le (1-1/k)|E|$, let $Z_1, Z_2, \dots, Z_\ell$ be k-sets of E and $W = E \setminus \bigcup_{i\in[\ell]} Z_i$ with $|W| < k$, then either Z_is are all dependent, or there exist two sets in $\{Z_i : i \in [\ell]\} \cup W$ whose elements can be exchanged so as to increase the number of dependent sets in $\{Z_i : i \in [\ell]\}$.

We begin with a helper lemma.

▶ **Lemma 12.** *Let $d \ge 0$ and $k = d+1$ be two integers. Let $P = (E, \mathcal{F})$ be a d-polynomoid. For any two disjoint k-subsets $X, Y \subseteq E$, if $r(X \cup Y) \le 2d$, then $X \cup Y$ can be partitioned into two disjoint dependent k-subsets of E. This partition can be done in $O(d^2 q(d+1))$ time.*

Proof. We begin with the proofs of the following claims.

▷ Claim 13. If X has a subset $Z \in \mathcal{F}$ with $|Z| \ge d$, then $Z \cup \{y\} \notin \mathcal{F}$ for some $y \in Y$.

Proof. If such a y does not exist, then $Z \cup \{y\} \in \mathcal{F}$ for every $y \in Y$. By Lemma 5, there is exactly one maximal independent set that contains Z. The above two facts imply that $Z \cup Y \in \mathcal{F}$. This yields $r(Z \cup Y) \ge 2d+1 > r(X \cup Y)$, a contradiction. ◁

▷ Claim 14. If $X \in \mathcal{F}$, there exists $y \in Y$, for every d-subset Z of X, $Z \cup \{y\} \notin \mathcal{F}$.

Proof. By Claim 13, $X \cup \{y\} \notin \mathcal{F}$ for some $y \in Y$. If X has a d-subset Z with $Z \cup \{y\} \in \mathcal{F}$, since $X \cap (Z \cup \{y\}) = Z$, by the join property of P we have $X \cup (Z \cup \{y\}) = X \cup \{y\} \in \mathcal{F}$, a contradiction. ◁

We are ready to give a proof. If both X and Y are dependent, then we are done. Otherwise, we assume w.l.o.g. that $X \in \mathcal{F}$. By Claim 14, there exists an $y^* \in Y$, for every $x \in X$, $X \setminus \{x\} \cup \{y^*\} \notin \mathcal{F}$. If $Y \setminus \{y^*\} \in \mathcal{F}$, then by Claim 13 there exists some $x^* \in X$ so that $Y \setminus \{y^*\} \cup \{x^*\} \notin \mathcal{F}$. Otherwise $Y \setminus \{y^*\} \notin \mathcal{F}$, then for any $x \in X$ we have $Y \setminus \{y^*\} \cup \{x\} \notin \mathcal{F}$. As a result, $X \setminus \{x^*\} \cup \{y^*\}$ and $Y \setminus \{y^*\} \cup \{x^*\}$ both are not in $\mathcal{F}$ and together partition $X \cup Y$. By enumerating all possible x^*, y^*, it yields the time bound. ◀

We are ready to prove that the swapping procedure can iterate until no sets in $\{Z_i : i \in [\ell]\}$ are independent.

▶ **Lemma 15.** *Let $d \ge 0$, $k = d+1$, and $\ell \ge 1$ be integers. Let $P = (E, \mathcal{F})$ be a d-polynomoid. For any subset S of E, if $|S| = \ell k + t$ and $r(S) \le \ell d + t$ for some $t \ge 0$, then S contains ℓ disjoint dependent k-subsets of E.*

Proof. Initially, we partition S arbitrarily into ℓ k-sets $Z_1, Z_2, \dots, Z_\ell$ and one t-set W. If none of Z_i for $i \in [\ell]$ is independent, then we are done. Otherwise, $Z_i \in \mathcal{F}$ for some $i \in [\ell]$. Since $Z_i \in \mathcal{F}$ and $|Z_i| = k \ge d$, by Lemma 5 there is exactly one maximal independent set $M(Z_i)$ that contains Z_i. Hence, there are at least $|S| - |M(Z_i)|$ elements $x \in S$ so that $Z_i \cup \{x\} \notin \mathcal{F}$. We color these elements blue. By the maximality of $r(S)$, $|M(Z_i)| \le r(S) \le \ell d + t$. Hence, $|S| - |M(Z_i)| \ge \ell k + t - \ell d - t = \ell$. There are two cases to discuss.

1. W contains a blue element b. By the definition of blue elements, $Z_i \cup \{b\} \notin \mathcal{F}$. Let y be an arbitrary element in Z_i. Then, $(Z_i \setminus \{y\}) \cup \{b\} \notin \mathcal{F}$; otherwise, by the join property of P

$$\{y\} \cup (Z_i \setminus \{y\}) \in \mathcal{F} \text{ and } (Z_i \setminus \{y\}) \cup \{b\} \in \mathcal{F} \text{ together imply } Z_i \cup \{b\} \in \mathcal{F},$$

a contradiction. If we set $(Z_i, W) \to (Z_i \setminus \{y\} \cup \{b\}, W \setminus \{b\} \cup \{y\})$, then the number of independent sets in $\{Z_1, Z_2, \dots, Z_\ell\}$ is reduced by one.

2. W contains no blue elements. Because Z_i also contains no blue elements, the $\geq \ell$ blue elements are distributed among $\{Z_1, Z_2, \ldots, Z_\ell\} \setminus \{Z_i\}$. By the pigeonhole principle, some Z_j contains at least 2 blue elements. We claim that $r(Z_i \cup Z_j) \leq 2d$. Thus, by Lemma 12 we can partition $Z_i \cup Z_j$ into two dependent k-sets. Hence, the number of independent sets in $\{Z_1, Z_2, \ldots, Z_\ell\}$ is reduced by at least one.
   We prove the claim as follows. Suppose for contradiction that $r(Z_i \cup Z_j) \geq 2d+1$. Because $|Z_i| + |Z_j| = 2(d+1)$, there exists one element x so that $Z_i \cup Z_j \setminus \{x\} \in \mathcal{F}$. Because there are two blue elements $b_1, b_2 \in Z_j$ and any independent superset of Z_i contains no blue elements, $x \notin Z_j$ and thus $x \in Z_i$. By the hereditary property of P, $Z_i \setminus \{x\} \cup \{b_1\} \in \mathcal{F}$. By the join property of P, $\{x\} \cup (Z_i \setminus \{x\}) \in \mathcal{F}$ and $(Z_i \setminus \{x\}) \cup \{b_1\} \in \mathcal{F}$ imply that $Z_i \cup \{b_1\} \in \mathcal{F}$, a contradiction.

For each case, we can reduce the number of independent sets in $\{Z_1, Z_2, \ldots, Z_\ell\}$ by at least one. Since ℓ is finite, one can always obtain a feasible packing. ◀

Let $|E| = \ell k + t$ for some $t \in [0, k)$. Thus $\lfloor |E|/k \rfloor = \ell$. Since $r(E) \leq (1 - 1/k)(\ell k + t) \leq \ell d + t$, by Lemma 15 we complete the proof of the second case.

4 Finding a Largest Dependent k-Set Packing in Deterministic Linear Time

In Theorem 8, we have shown that computing $r(E)$ in general requires $\Omega(|E|^{2-\varepsilon})$ time for any constant $\varepsilon > 0$ unless the 3SUM conjecture fails. Hence, to compute the k-set packing in $O(E)$ time, one cannot directly compute $r(E)$ to distinguish which case in Theorem 9 applies.

4.1 The Deterministic Linear-Time Algorithm

In this section, we devise a deterministic linear-time algorithm for finding a largest dependent set packing for any d-polynomoid $P = (E, \mathcal{F})$. Let d, k be two integers with $k = d+1$. Recall that we have to consider only the case of $k = d + 1$. Let $|E| = \ell k + t$ for some $t \in [0, k)$. Initially, we partition E arbitrarily into $Z_1, Z_2, \ldots, Z_\ell$ and W so that $|Z_i| = k$ for $i \in [\ell]$ and $|W| = t$. Then we apply the following five steps to find a largest dependent k-set packing $\mathcal{D}$.

1. Set $\mathcal{A} = \{Z_1, Z_2, \ldots, Z_\ell\}$. Set $\mathcal{D} = \emptyset$.
2. If some $Z_i \in \mathcal{A}$ is dependent, remove Z_i from $\mathcal{A}$ and set $\mathcal{D} = \mathcal{D} \cup \{Z_i\}$. Otherwise, proceed to the next step.
   ▶ Remark. This step takes $O(q(k)|E|/k)$ time.
3. If there exist $Z_i, Z_j \in \mathcal{A}$ that $Z_i \cup Z_j$ is dependent, remove Z_i and Z_j from $\mathcal{A}$ and set $\mathcal{D} = \mathcal{D} \cup \{Z_i', Z_j'\}$ where Z_i' and Z_j' are dependent k-sets and they partition $Z_i \cup Z_j$. Otherwise, proceed to the next step.
   ▶ Remark. The existence of Z_i' and Z_j' is shown in Section 4.2. Because $|Z_i| \geq d$ and $|Z_j| \geq d$, $Z_i \cup Z_j$ is independent iff Z_i and Z_j belong to the same maximal set. So if $Z_i \cup Z_j \in \mathcal{F}$ but $Z_i \cup Z_k \notin \mathcal{F}$, then $Z_j \cup Z_k \notin \mathcal{F}$. Hence, we can keep a list of Zs so that their pairwise unions are independent sets. For each unpaired Y outside the pool, pick any Z in the pool, if $Z \cup Y \in \mathcal{F}$, then $Z' \cup Y \in \mathcal{F}$ for any Z' in the pool, so expand the pool by adding Y; otherwise, pair Y, Z and throw out Y, Z. Indeed, this is a generalization of the majority voting [8]. This step takes $O((q(2k) + k^2 q(k))|E|/k)$ time.
4. If $\mathcal{A} = \emptyset$, return $\mathcal{D}$ and stop. Otherwise, find a maximal independent set $M_{\mathcal{A}}$ of P that contains all elements in $\mathcal{A}$ as subsets.

▶ Remark. If $\mathcal{A} = \emptyset$, then $\mathcal{D}$ is largest possible, so it is a largest dependent k-set packing. The existence of $M_{\mathcal{A}}$ is shown in Section 4.2. To find $M_{\mathcal{A}}$, let Z be some element in $\mathcal{A}$ and find the maximal independent set that contains Z as a subset in $O(|E|q(k))$ time by Corollary 6. Since $|Z| = k \geq d$ and $Z \in \mathcal{F}$, we known that $M_{\mathcal{A}} = M_Z$ by Lemma 5.

5. This final step is reached only if $|\mathcal{A}| \geq 1$.
   - Case 1: $r(E) > (1 - 1/k)|E|$. By Lemma 18, $M_{\mathcal{A}}$ is a maximum independent set of P. Since $M_{\mathcal{A}}$ is given, one can simulate Lemma 10 in $O(|E|)$ time.
   - Case 2: $r(E) \leq (1 - 1/k)|E|$. By Lemma 17, $M_{\mathcal{A}}$ contains all elements in $\mathcal{A}$. One can simulate Lemma 15 efficiently as follows.
     For each $Z \in \mathcal{A}$, to implement Lemma 15, we need to find a set $Z' \in \{Z_i : i \in [\ell]\} \cup \{W\}$ that contains a sufficient number of elements in $E \setminus M_{\mathcal{A}}$ (aka "blue elements" in the proof of Lemma 15). Given $M_{\mathcal{A}}$, we compute $E \setminus M_{\mathcal{A}}$ in $O(|E|)$ time and maintain the locations of these "blue elements" so that for each $Z \in \mathcal{A}$ we can find the Z' in $O(1)$ time.

     Hence, the number of swapping steps is $O(|E|/k)$ and each takes $O(k^2 q(k))$ time (Lemma 12). So this step needs $O(k|E|q(k))$ time.

As a consequence, the total running time of all steps is bounded by $O(k|E|q(2k))$. This completes the proof of Theorem 2 for $k > d$.

4.2 The Existence Proofs

We will prove the existence proofs required by the algorithm in Section 4.1. In Step 3, $Z_i, Z_j \in \mathcal{F}$ and $Z_i \cup Z_j \notin \mathcal{F}$, so $r(Z_i \cup Z_j) \leq 2d$ by Lemma 16. Hence, by Lemma 12, $Z_i \cup Z_j$ can be partitioned into two disjoint dependent sets of size k. The existence of $M_{\mathcal{A}}$ required by Step 4 is shown in Lemma 17. Finally, we prove in Lemma 18 that $M_{\mathcal{A}}$ is a maximum independent set of P if $r(E) \geq (1 - 1/k)|E|$ and $|\mathcal{A}| \geq 1$.

▶ **Lemma 16.** *For any k-subsets X, Y of E, if $X, Y \in \mathcal{F}$ but $X \cup Y \notin \mathcal{F}$, then $r(X \cup Y) \leq 2d$.*

Proof. Let S be any subset of $X \cup Y$ with $|S| = r(X \cup Y)$. If $|S| > 2d$, then there exists $z \in X \cup Y$ so that $(X \cup Y) \setminus \{z\} \in \mathcal{F}$. We assume w.l.o.g. that $z \in X$ and $Y \subseteq S$. Let M be a maximal independent set of P that contains S as a subset. Because $|M \cap X| \geq |S \cap X| \geq d$, by the join property of P we have $M \cup X \in \mathcal{F}$. By the maximality of M, $X \subseteq M$, so $X \cup Y \subseteq M$. By the hereditary property of P, $M \in \mathcal{F}$ implies $X \cup Y \in \mathcal{F}$. This violates the setting. Therefore, $r(X \cup Y) \leq 2d$. ◀

▶ **Lemma 17.** *In Step 4, such a maximal independent set $M_{\mathcal{A}}$ of P always exists.*

Proof. The construction of $\mathcal{A}$ yields that, for any $X, Y \in \mathcal{A}$, $X \cup Y \in \mathcal{F}$. By Lemma 5, there is a unique maximal independent set M_X (resp. $M_{X \cup Y}$) that contains X (resp. $X \cup Y$) as a subset. The uniqueness of M_X and $M_{X \cup Y}$ implies that $M_X = M_{X \cup Y}$. Similarly, $M_Y = M_{X \cup Y}$. Hence, $M_X = M_Y$. Because this fact applies to every pair of elements in $\mathcal{A}$, there is a maximal independent set $M_{\mathcal{A}}$ that contains every element in $\mathcal{A}$ as a subset. ◀

▶ **Lemma 18.** *In Step 4, if $r(E) > (1 - 1/k)|E|$ and $|\mathcal{A}| \geq 1$, then $M_{\mathcal{A}}$ is a maximum independent set of P.*

Proof. We begin with the proofs that, for any maximal independent set M of P, each removed Z_i in Steps 2 and 3 contains at least one element outside M on average.

▷ Claim 19. Let M be any maximal independent set of P. In Step 2, for each removed Z_i, Z_i contains at least one element in $E \setminus M$.

Proof. Because $Z_i \notin \mathcal{F}$ and $M \in F$, by the hereditary property of P, Z_i is not a subset of M. Hence, Z_i contains at least one element outside $E \setminus M$. ◁

▷ Claim 20. Let M be any maximal independent set of P. In Step 3, for each pair of removed Z_i and Z_j, $Z_i \cup Z_j$ contains at least two elements in $E \setminus M$.

Proof. By Lemma 16, we know that $r(Z_i \cup Z_j) \leq 2d$. If $M \cap (Z_i \cup Z_j) > 2d$, then $r(Z_i \cup Z_j) > 2d$, a contradiction. Hence, $M \cap (Z_i \cup Z_j) \leq 2d$, as desired. ◁

We are ready to give a proof. Let S be a subset of E with $|S| = r(E)$, and let

$$U_{\mathcal{A}} = W \cup \bigcup_{Z \in \mathcal{A}} Z \text{ and } U_B = E \setminus U_{\mathcal{A}}.$$

By Claims 19 and 20, we have

$$\frac{|S \cap U_B|}{|U_B|} \leq \frac{k-1}{k}. \tag{1}$$

By restating $r(E) > (1 - 1/k)|E|$, we get

$$\frac{|S \cap E|}{|E|} > \frac{k-1}{k}. \tag{2}$$

Combining (1), (2), and the average argument, it yields that

$$\frac{|S \cap U_{\mathcal{A}}|}{|U_{\mathcal{A}}|} > \frac{k-1}{k}. \tag{3}$$

To satisfy the inequality (3), if $|\mathcal{A}| \geq 1$, either S contains $U_{\mathcal{A}} \setminus W$ as a subset or S does not contain $U_{\mathcal{A}} \setminus W$ as a subset. The former implies that $S = M_{\mathcal{A}}$ due to the uniqueness of $M_{\mathcal{A}}$ (Lemma 5), as desired. Note that $|S \cap (U_{\mathcal{A}} \setminus W) \leq k-2|$; otherwise, $S \cup M_{\mathcal{A}} \in \mathcal{F}$ due to the join property of P and thus $S = M_{\mathcal{A}}$ by the maximality of S and $M_{\mathcal{A}}$. The latter cannot hold because $|S \cap (U_{\mathcal{A}} \setminus W)| \leq k-2$ and $0 \leq |W| \leq k-1$ implies that

$$\frac{k-1}{k} < \frac{|S \cap U_{\mathcal{A}}|}{|U_{\mathcal{A}}|} = \frac{|S \cap W| + |S \cap (U_{\mathcal{A}} \setminus W)|}{|W| + |U_{\mathcal{A}} \setminus W|} \leq \frac{|W| + k - 2}{|W| + k|\mathcal{A}|} \leq \frac{2k-3}{2k-1},$$

which cannot hold for positive k. ◀

5 Conclusion

We obtain a complexity trichotomy result for the dependent k-set packing problem on d-polynomoids. For each of the three categories, our algorithm is optimal. It may worth noting that the running time of the algorithm for the case of $k > d$ can be reduced by a factor of k by group testing [30, 22], which will be introduced in the full version of this manuscript. Though this yields a constant-factor improvement, it may affact the performance of real applications.

References

1 Ron Aharoni. Matchings in n-partite n-graphs. *Graphs Comb.*, 1(1):303–304, 1985.
2 Ron Aharoni, Eli Berger, Dani Kotlar, and Ran Ziv. Degree conditions for matchability in 3-partite hypergraphs. *J. Graph Theory*, 87(1):61–71, 2018.
3 Ron Aharoni and Penny Haxell. Hall's theorem for hypergraphs. *J. Graph Theory*, 35(2):83–88, 2000.
4 N. Alon. A note on the decomposition of graphs into isomorphic matchings. *Acta Mathematica Hungarica*, 42(3-4):221–223, September 1983.
5 János Barát and Dániel Gerbner. Edge-decomposition of graphs into copies of a tree with four edges. *Electron. J. Comb.*, 21(1):1, 2014.
6 A. Bialostocki and Y. Roditty. 3K2-decomposition of a graph. *Acta Mathematica Hungarica*, 40(3-4):201–208, September 1982.
7 Ahmad Biniaz, Anil Maheshwari, Subhas C. Nandy, and Michiel H. M. Smid. An optimal algorithm for plane matchings in multipartite geometric graphs. In *Algorithms and Data Structures – 14th International Symposium (WADS)*, pages 66–78. Springer, 2015.
8 Robert S. Boyer and J. Strother Moore. MJRTY: A fast majority vote algorithm. In *Automated Reasoning: Essays in Honor of Woody Bledsoe*, pages 105–118. Kluwer Acad. Publishers, 1991.
9 Andries Brouwer and R. Wilson. The decomposition of graphs into ladder graphs, January 1980.
10 Krzysztof Brys and Zbigniew Lonc. Polynomial cases of graph decomposition: A complete solution of Holyer's problem. *Discret. Math.*, 309(6):1294–1326, 2009.
11 Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment problems: revised reprint*. SIAM, 2012.
12 Ho Yee Cheung, Lap Chi Lau, and Kai Man Leung. Graph connectivities, network coding, and expander graphs. *SIAM J. Comput.*, 42(3):733–751, 2013.
13 Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14(1):210–223, 1985.
14 David E Daykin and Roland Häggkvist. Degrees giving independent edges in a hypergraph. *Bulletin of the Australian Mathematical Society*, 23(1):103–109, 1981.
15 Dorit Dor and Michael Tarsi. Graph decomposition is NP-complete: A complete proof of Holyer's conjecture. *SIAM J. Comput.*, 26(4):1166–1187, 1997.
16 M.E. Dyer and A.M. Frieze. On the complexity of partitioning graphs into connected subgraphs. *Discrete Applied Mathematics*, 10(2):139–153, 1985.
17 Jack Edmonds. Minimum partition of a matroid into independent subsets. *Journal of Research of the National Bureau of Standards*, 69B:67–72, 1965.
18 O. Favaron, Z. Lonc, and M. Truszczynski. Decomposition of graphs into graphs with three edges. *Ars Combin.*, 20:125–146, 1985.
19 Anka Gajentaan and Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995.
20 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
21 Eunice Gogo Mphako. Circuit and cocircuit partitions of binary matroids. *Czechoslovak Mathematical Journal*, 56(1):19–25, 2006.
22 Oded Goldreich. Combinatorial property testing (a survey). *Randomization Methods in Algorithm Design*, 43:45–59, 1999.
23 Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.
24 Philip Hall. On representatives of subsets. *Classic Papers in Comb.*, pages 58–62, 1987.
25 Jie Han and Peter Keevash. Finding perfect matchings in dense hypergraphs. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2366–2377. SIAM, 2020.
26 Jie Han and Andrew Treglown. The complexity of perfect matchings and packings in dense hypergraphs. *J. Comb. Theory, Ser. B*, 141:72–104, 2020.

27 Penny E. Haxell. A condition for matchability in hypergraphs. *GCOM.*, 11(3):245–248, 1995.
28 Ian James Holyer. The computational complexity of graph theory problems, 1981.
29 Cor A. J. Hurkens and Alexander Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discret. Math.*, 2(1):68–72, 1989.
30 Frank K Hwang. A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67(339):605–608, 1972.
31 Per M. Jensen and Bernhard Korte. Complexity of matroid property algorithms. *SIAM J. Comput.*, 11(1):184–190, 1982.
32 Viggo Kann. Maximum bounded 3-dimensional matching is MaxSNP-complete. *Inf. Process. Lett.*, 37(1):27–35, 1991.
33 Peter Keevash, Fiachra Knox, and Richard Mycroft. Polynomial-time perfect matchings in dense hypergraphs. In *Symp. on Theory of Computing (STOC)*, pages 311–320. ACM, 2013.
34 Imdadullah Khan. Perfect matchings in 3-uniform hypergraphs with large vertex degree. *SIAM J. Discret. Math.*, 27(2):1021–1039, 2013.
35 Imdadullah Khan. Perfect matchings in 4-uniform hypergraphs. *J. Comb. Theory, Ser. B*, 116:333–366, 2016.
36 Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{VE})$ algorithm for finding maximum matching in general graphs. In *Symp. on Fnd. of Computer Science (FOCS)*, pages 17–27. IEEE, 1980.
37 James G. Oxley. *Matroid theory.* Oxford University Press, 1992.
38 Vojtech Rödl, Andrzej Rucinski, and Endre Szemerédi. Perfect matchings in uniform hypergraphs with large minimum degree. *Eur. J. Comb.*, 27(8):1333–1349, 2006.
39 Paul D. Seymour. Packing and covering with matroid circuits. *J. Comb. Theory, Ser. B*, 28(2):237–242, 1980.
40 Nicolas Teypaz and Christophe Rapine. Graph decomposition into paths under length constraints. Technical report, UniK University Graduate Center, University of Oslo, 2004.
41 D.J.A. Welsh. Euler and bipartite matroids. *Journal of Comb. Theory*, 6(4):375–377, 1969.
42 Douglas B. West. *Combinatorial Mathematics.* CUP, 2020.
43 Sanming Zhou. Minimum partition of an independence system into independent sets. *Discret. Optim.*, 6(1):125–133, 2009.

A Hardness Reduction

In this section, we prove that the dependent k-set packing for d-polynomoids with $d \geq k$ is as hard as hypergraph matchings in general. We reduce the matching problem for k-uniform hypergraphs to the dependent k-set packing problem on d-polynomoids with $d \geq k$ in Theorem 21. The other direction is clear because the latter problem is a special case of the former one. This completes the proof of Theorem 2 for $k \leq d$.

▶ **Theorem 21.** *For any integers $d \geq k \geq 1$, there exists a d-polynomoid $P = (E, \mathcal{F})$ so that dependent k-set packing on P is as hard as matchings on k-uniform hypergraphs.*

Proof. We prove this lemma by showing a reduction from the perfect matching for k-uniform graphs to the dependent k-set packing for P. Let $G = (V, \mathcal{E})$ be a k-uniform hypergraph. Let $P = (V, \mathcal{F})$ be a d-polynomoid where $\mathcal{F} = \{\text{all subsets of } V \text{ with size at most } k\} \setminus \mathcal{E}$. We now show that P is a d-polynomoid. Note that each set in $\mathcal{F}$ has size at most k. By the definition of $\mathcal{F}$, all subsets of V with size at most $k-1$ are in $\mathcal{F}$ so P satisfies the hereditary property. When $k = d$, two sets A, B in $\mathcal{F}$ have $|A \cap B| \geq d$ if and only if $A = B$. When $k < d$, two sets A, B in $\mathcal{F}$ have $|A \cap B| \geq d$ cannot happen. Hence, P satisfies the join property. Observe that a hyperedge is in G if and only if it is a dependent set of P. Hence, finding a dependent k-set packing for P is equivalent to finding a matching for G. ◀

B More Applications

In this section, we present more applications of our results that are not covered in Section 1.1.

1. Let $G = (V, E)$ be an m-edge undirected simple graph. Let $P = (E, \mathcal{F})$ so that $\mathcal{F} = \{A \subseteq E : \text{ there exists a triangle in } G \text{ that contains all edges in } A\}$. One may verify that P is a 2-polynomoid. By Theorem 2, we obtain a deterministic linear-time algorithm for the dependent k-set packing for P, which is equivalent to partitioning the edge set of G into subsets, each of which edge-induces a 3-edge forest. In addition, we obtain the following sufficient and necessary condition:
   ▶ **Corollary 22.** *An m-edge undirected simple graph G can be edge-partitioned into 3-edge forests if and only if m is a multiple of 3 and G is not a triangle.*
   In [5], they give a sufficient and necessary condition to partition the edge set of the given graph into a designated four-edge tree for highly-edge-connected graphs.
2. Let $G = (V, E)$ be a complete multipartite graph. Let $P = (V, \mathcal{F})$ so that $\mathcal{F} = \{A \subseteq V : \text{ all vertices in } A \text{ are from the same partite set}\}$. One may verify that P is a 1-polynomoid. By Theorem 2, we obtain the following sufficient and necessary condition:
   ▶ **Corollary 23.** *An n-vertex undirected simple complete multipartite graph G has a perfect matching if and only if n is a multiple of 2 and the number of vertices in a largest partite set is at most $n/2$.*
   By an argument in [7], Corollary 23 suffices to prove:
   ▶ **Corollary 24.** *Given a set of points in $\mathcal{R}^2$ in general position where each point has a color in $[c]$, one can group the points into pairs so that the line segment joining the points in a pair does not cross that of another pair and the points in each pair have different colors if and only if n is a multiple of 2 and for each color $i \in [c]$ the number of points of color i is at most $n/2$.*

C Omitted Proofs

▶ **Lemma 25.** *Let $G = (V, E)$ be an undirected simple graph consisting of six edges. If E can be partitioned into E_1, E_2 so that E_1 edge-induces a triangle and E_2 edge-induces a non-star graph, then E can also be partitioned into two subsets so that each subset edge-induces a 3-edge linear forest.*

Proof. By definition, E_2 edge-induces either a triangle or a 3-edge linear forest. Suppose that E_2 edge-induces a triangle. For any $E' \subseteq E$, let $V(E')$ denote the set of the end-vertices of edges in E'. Since G is simple, $V(E_1)$ and $V(E_2)$ have at most one vertex in common. Hence, there exist $e_1 \in E_1, e_2 \in E_2$ so that $V(\{e_1\}) \cap V(E_2) = \emptyset$ and $V(\{e_2\}) \cap V(E_1) = \emptyset$. This yields that $E_1 \cup \{e_2\} \setminus \{e_1\}$ and $E_2 \cup \{e_1\} \setminus \{e_2\}$ both edge-induce $P_3 \cup P_2$, i.e., two vertex-disjoint paths of 3 verices and 2 vertices.

Suppose that E_2 edge-induces a 3-edge linear forest. Let e_2 be an edge in E_2 so that $E_2 \setminus \{e_2\} = 2P_2$, i.e., two edges with no end-vertices in common. Since G is simple and E_1 edge-induces a complete graph, $|V(\{e_2\}) \cap V(E_1)| \leq 1$. Hence, $E_1 \cup \{e_2\}$ has at most one vertex of degree ≥ 3 and e_2 is not an edge in any cycle of $E_1 \cup \{e_2\}$. Let e_1 be any edge in E_1 incident with the maximum degree vertex in $E_1 \cup \{e_2\}$. Thus, $E_1 \cup \{e_2\} \setminus \{e_1\}$ has max degree ≤ 2 and contains no cycle, i.e., a linear forest. On the other hand, because the union of $2P_2$ and any other edge also has max degree ≤ 2 and contains no cycle, $E_2 \cup \{e_1\} \setminus \{e_2\}$ also is a linear forest. ◀