Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master in Advanced Mathematics and Mathematical Engineering

Master's thesis

Finding Partite Hypergraphs Efficiently

Ferran Espuña Bertomeu

Supervised by Richard Lang

June 2025

Thanks to...

Abstract

Keywords

hypergraph, algorithm, graph, partite, extremal

1. Introduction

TODO: Write introduction

2. Preliminaries

In this section we introduce some basic definitions and results that are used throughout this thesis. We start by defining k-graphs, which generalize the usual notion of a graph.

Definition 2.1. For an integer $k \geq 2$ a finite *k-graph* is a tuple $G = (V, E)$ where V is a finite set and $E \subset \binom{V}{k}$. We call the elements of $V(G) = V$ its *vertices* and those of $E(G) = E$ its *edges*.

Remark 2.2. If we let $k = 2$ we recover the usual definition of an undirected graph with no loops.

Definition 2.3. Let $G = (V, E)$ be a k-graph and $v \in V$. The *degree $d_G(v)$* of v in G is the number of edges containing v, that is

$$d_G(v) = |\{e \in E \mid v \in e\}|.$$

Next, we introduce k-graph homomorphisms, embeddings and isomorphisms, which allow us to relate k-graphs (with the same value of k) to each other:

Definition 2.4. Let $G = (V, E)$ and $H = (W, F)$ be k-graphs. A *homomorphism* from G to H is a map $f : V \to W$ such that for every edge $e \in E$ the set $f(e) = \{f(v) \mid v \in e\}$ is an edge in H (that is, $f(e) \in F$). If such a homomorphism exists and is injective, we say that f is an embedding of G on H and that H contains G as a subgraph. If, furthermore, $f^{-1} : \mathrm{Im}(f) \to V$ is a homomorphism, we say that f is an *induced* embedding and that H contains G as an *induced* subgraph. We write $G \subset H$. If, in addition, f is a bijection, we say that f is an *isomorphism* and that G is *isomorphic* to H. We write $G \cong H$.

It is elementary to check that (induced) inclusion is an order relation and that isomorphism is an equivalence relation. Furthermore, isomorphism preserves (induced) inclusion. Therefore, we can talk about the (induced) subgraph condition up to isomorphism, both in the *host* k-graph (H) and in the *guest* k-graph (G).

Now we can state the *forbidden subgraph problem* for k-graphs. Informally, given a k-graph G, and an integer $n \geq |V(G)|$, we want to find the smallest M_0 such that all k-graphs with n vertices and $m > M_0$ edges contain G as a subgraph.

Proposition 2.5. *Let $G = (V, E)$ be a k-graph with nonempty edge set and $n \geq |V|$ be an integer. Then there exists an integer $M_0 = ex(n, G) \in \left[0, \binom{n}{k}\right)$ such that the condition*

"All k-graphs with n vertices and m edges contain G as a subgraph"

is true for all $\binom{n}{k} \geq m > M_0$ and false for all $0 \leq m \leq M_0$.

Proof. Note that, if M_0 exists, clearly it is unique. Also, the condition is clearly false for $m = 0$ and true for $m = \binom{n}{k}$ (the only graph H with vertex set W, $|W| = n$ and $\binom{|V|}{k}$ edges is the one having all k-sets of vertices so any injective map $f : V \to W$ is an embedding of G in H). We only need to show that if the condition is true for m then it is true for all $m' \geq m$. Suppose it is true for m and let $m' \geq m$. Let $H = (W, F)$ be a k-graph with n vertices and m' edges. We can just take $F' \subset F$ with $|F'| = m$. By hypothesis, the graph $H' = (W, F')$ contains G as a subgraph, and the identity map in W is an embedding of H' in H:

$$G \subset H' \subset H \implies G \subset H. \qquad \square$$

Definition 2.6. The integer $ex(n, G)$ defined in Proposition 2.5 is called the *extremal number* of G.

Remark 2.7. The extremal number is clearly invariant under isomorphism of G.

Definition 2.8. for an integer $p \geq k$, a k-graph $G = (V, E)$ is *p-partite* if there exists a partition $V = V_1 \cup \cdots \cup V_p$ such that every edge $e \in E$ intersects every part V_i in at most one vertex. We may write $G = (V_1, \ldots, V_p; E)$ and say that G is a partite k-graph on $V_1, \ldots, V_p$.

Remark 2.9. If $p = k$, every edge intersects every part in exactly one vertex, so we can identify the edges with a subset of $V_1 \times \cdots \times V_k$. If it is clear from context, we may slightly abuse notation when talking about ordered and unordered sets of vertices, as in the definition below.

Definition 2.10. A k-partite k-graph $G = (V_1, \ldots, V_k; E)$ is *complete* if $E = V_1 \times \cdots \times V_k$. That is, if all $(v_1, \ldots, v_k) \in V_1 \times \cdots \times V_k$ satisfy $\{v_1, \ldots, v_k\} \in E$.

Remark 2.11. If $V_1, \ldots, V_k, W_1, \ldots, W_k$ are disjoint sets and $|V_i| = |W_i| = a_i$ for all i,

$$K(V_1, \ldots, V_k) \cong K(W_1, \ldots, W_k),$$

where the isomorphism is given by any bijection

$$f : V_1 \times \cdots \times V_k \to W_1 \times \cdots \times W_k$$

such that $f(V_i) = W_i$ for all i. This allows us to talk, up to isomorphism, about *the* complete k-partite k-graph with part sizes $a_1, \ldots, a_k$, which we denote by

$$K(a_1, \ldots, a_k).$$

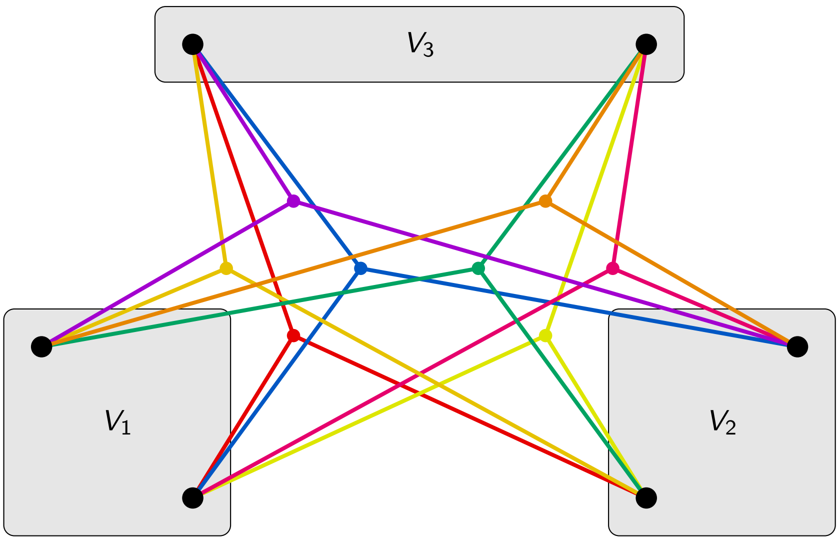

Figure 1: The complete 3-partite 3-graph $K(2, 2, 2)$, with parts V_1, V_2, V_3. Each vertex is represented as a black dot while each edge is represented as one of the colored dots, and connected by a line to the vertices it contains.

A different but equivalent way to view completeness is to talk about *common neighbors*:

Definition 2.12. Let $G = (V, E)$ be a 2-graph. The *common neighborhood* of a set $S \subset V$ is

$$N_G(S) = \{y \in V \mid \forall x \in S : \{x, y\} \in E\}.$$

That is, the common neighborhood of a set $S \subset U$ is the maximal set $T \subset W$ such that $S \times T \subset E$. Then, a 2-partite 2-graph $G = (U, W; E)$ is complete if and only if $N_G(U) = W$. This might seem trivial when dealing with 2-graphs, but it can be generalized to k-graphs in the following way.

Definition 2.13. Let $G = (V, E)$ be a k-graph. Let $S \subset V$ be a set of vertices. The *joint link* of S is the $(k-1)$-graph $G_S = (V \setminus S, E')$, where

$$E' = \left\{ Y \in \binom{V}{k-1} \middle| \forall x \in S : \{x\} \cup Y \in E \right\}.$$

Then, we can define completeness of k-partite k-graphs recursively: If $G = (V_1, \ldots, V_k; E)$ is a k-partite k-graph with $k \geq 3$, G is complete if and only if the graph G_{V_k} (which is a $(k-1)$-graph, partite on the sets $V_1, \ldots, V_{k-1}$) is complete.

Remark 2.14. All k-partite k-graphs with part sizes $b_1 \leq a_1, \ldots, b_k \leq a_k$ are contained in $K(a_1, \ldots, a_k)$ as subgraphs. This lets us follow the same argument as in Proposition 2.5 to define the following.

Definition 2.15. Let $0 < t_1 \leq v_1, \ldots, 0 < t_k \leq v_k$ be integers. Then the *generalized Zarankiewicz number* $z(v_1, \ldots, v_k; t_1, \ldots, t_k)$ is the largest integer $0 \leq z < \prod_i v_i$ for which there exists a k-partite k-graph H with part sizes $|V_1| = v_1, \ldots, |V_k| = v_k$ and z edges such that no embedding f of $K(T_1, \ldots, T_k)$ with $|T_i| = t_i$ in it exists satisfying $f(T_i) \subset V_i$ for all i.

From now on, every time we talk about embeddings from one k-partite k-graph onto another we assume the condition $f(T_i) \subset V_i$.

Remark 2.16. Finding this number can help us upper bound the extremal number of $K(t_1, \ldots, t_k)$ asymptotically: Assume that G is a $K(t_1, \ldots, t_k)$-free n-vertex k-graph with m edges. pick $v_1, \ldots, v_k$ such that $\sum_i v_i = n$ and $v_i \sim n/k$ (For example $\lfloor n/k \rfloor \leq v_i \leq \lceil n/k \rceil$) Let $V_1, \ldots, V_k$ be a random partition of $V(G)$ with $|V_i| = v_i$. for an edge $e \in E(G)$, the probability that e is an edge in $K(V_1, \ldots, V_k)$ is greater than

$$k! \prod_i v_i/n \sim k!(1/k)^k$$

which is independent of n. Therefore, the expected number of edges satisfying this condition is a positive fraction of m. Applying the first moment method, we can conclude that

$$ex(n, K(t_1, \overset{k}{\ldots}, t_k)) = O_n(z(\lceil n/k \rceil, \overset{k}{\ldots}, \lceil n/k \rceil; t_1, \ldots, t_k)).$$

The problem on finding the Zarankiewicz number was first posed by K. Zarankiewicz in 1951 for the case of bipartite 2-graphs (that is, finding $z(u, w; s, t)$), in terms of finding all-1 minors in a $0-1$ matrix. An upper bound for it in the case $m = n, s = t$ was found by Kővari, Sós and Turán in [4] in 1954. This was generalized to arbitrary complete bipartite 2-graphs by C. Hyltén-Cavallius in [3] in 1958. The result is stated and proved here for completeness:

Theorem 2.17. *Let $0 < s \leq u$ and $0 < t \leq w$ be integers. Then*

$$z(u, w; s, t) \leq (s-1)^{1/t}(w-t+1)u^{1-1/t} + (t-1)u$$

Proof. Suppose that we have a bipartite graph $G = (U, W; E)$ with $|U| = u$, $|W| = w$ and $|E| = z$ exceeding the bound stated above. Let us consider the set

$$P = \left\{ (x, Y) \in U \times \binom{W}{t} \middle| x \in N_G(Y) \right\}.$$

Counting on the first coordinate, we get

$$|P| = \sum_{x \in U} \binom{d_G(x)}{t} = \sum_{x \in U} \varphi(d_G(x)) \geq u \sum_{x \in U} \varphi(z/u) = u\binom{z/u}{t}, \tag{1}$$

where we define

$$\varphi(x) = \begin{cases} \binom{x}{t}, & \text{if } x \geq t-1. \\ 0, & \text{otherwise.} \end{cases}$$

The function φ is convex, so we get the inequality in (1) as a consequence of Jensen's inequality. The other equalities come from the fact that $\varphi(d)$ agrees with $\binom{d}{t}$ for all integers $d \geq 0$; and that by our bound on z, $z \geq (t-1)u \implies z/u \geq t-1$.

If we had s different elements of P with the same second coordinate T, they would all necessarily have different first coordinates (say $S = \{x_1, \ldots, x_s\}$). But now, by definition of P, for all $a \in S, b \in T$, we have $\{a, b\} \in E$. This would mean that the inclusion map from $S \cup T$ to $U \cup W$ is an embedding of $K(s, t)$ in G, as described in Definition 2.15. Supposing that this is not the case, by the averaging we get

$$|P| \leq (s-1)\binom{w}{t}. \tag{2}$$

Putting inequalities (1) and (2) together, we get

$$u\binom{z/u}{t} \leq (s-1)\binom{w}{t}. \tag{3}$$

Now, because we can see E as a subset of $U \times W$, we get $z \leq uw \implies z/u \leq w$. In particular, we have

$$\frac{(z/u-(t-1))^t}{\binom{z/u}{t}} \leq \frac{(w-(t-1))^t}{\binom{w}{t}}, \tag{4}$$

which is true for each factor when expanding the denominators. Multiplying inequalities (3) and (4) yields

$$u\,(z/u-(t-1))^t \leq (s-1)(w-(t-1))^t.$$

Algebraic manipulation then gives

$$z \leq (s-1)^{1/t}(w-t+1)u^{1-1/t} + (t-1)u,$$

In contradiction with our assumption. □

Remark 2.18. Following Remark 2.16, we can use this bound to get an upper bound on the extremal number of $K(s, t)$:

$$ex(n, K(s,t)) = O_n\left((s-1)^{1/t}\left(\left\lceil\frac{n}{2}\right\rceil - t + 1\right)n^{1-1/t} + (t-1)\left\lceil\frac{n}{2}\right\rceil\right) = O_n\left(n^{2-1/t}\right).$$

Note that if $s < t$, we get the better bound $O_n\left(n^{2-1/s}\right)$ by interchanging the roles of s and t.

In 1964, Erdős [2] generalized this result to arbitrary complete partite k-graphs in the following theorem.

Theorem 2.19. *For $k \geq 2$ and $2 \leq t \leq \frac{n}{k}$, $ex(n, K(t, \overset{k}{\ldots}, t)) = O_n\left(n^{k-\frac{1}{t^{k-1}}}\right)$.*

Proof. By Remark 2.16, it suffices to show that

$$z = z(w, \overset{k}{\ldots}, w; t, \overset{k}{\ldots}, t) = O_w\left(w^{k - \frac{1}{t^{k-1}}}\right).$$

We prove this by induction on k. For $k = 2$, this is obtained by setting $u = w$ and $s = t$ in Theorem 2.17. For $k > 2$, suppose by way of contradiction that the theorem is false. For all $w_0 \in \mathbb{N}$, $K \in \mathbb{R}^+$, there exists a k-partite k-graph $G = (W_1, \ldots, W_k; E)$ with part sizes $|W_i| = w \geq w_0$ and $|E| \geq Kw^{k - \frac{1}{t^{k-1}}}$ such that no embedding of $K(t, \overset{k}{\ldots}, t)$ in it exists. Consider, for each set $T \in \binom{W_k}{t}$, the associated joint link G_T. By our assumption, G_T does not contain $K(t, \overset{k-1}{\ldots}, t)$ as a subgraph. If we let

$$z' = z(w, \overset{k-1}{\ldots}, w; t, \overset{k-1}{\ldots}, t),$$

this means that G_T has at most z' edges.

Now, consider the bipartite graph $G' = (U, W; E')$, where

$$\begin{aligned} U &= W_1 \times \cdots \times W_{k-1}, \\ W &= W_k, \\ E' &= \{(X, y) \in V \times W \mid X \cup \{y\} \in E\}. \end{aligned}$$

Clearly, G' has the same number of edges as G. Furthermore, G' does not contain $K(z' + 1, t)$ as a subgraph. Finally, we invoke Theorem 2.17 with $u = |U| = w^{k-1}$ and $s = z' + 1$ to get

$$Kw^{k - \frac{1}{t^{k-1}}} \leq |E| = |E'| \leq (z')^{1/t}(w - t + 1)w^{(k-1)(1-1/t)} + (t-1)w^{k-1}. \tag{5}$$

By the inductive hypothesis, for w_0 and K' large enough, we can bound

$$z' \leq K'w^{(k-1) - \frac{1}{t^{k-2}}}.$$

Substituting this into inequality (5) and approximating yields

$$Kw^{k - \frac{1}{t^{k-1}}} \leq (K')^{1/t}w^{k - \frac{1}{t^{k-1}}} + (t-1)w^{k-1}.$$

Combining like terms and picking $K > 2(K')^{1/t}$ gives

$$\frac{1}{2}Kw^{k - \frac{1}{t^{k-1}}} < (t-1)w^{k-1},$$

which we can rewrite as

$$\frac{1}{2}Kw^{1 - \frac{1}{t^{k-1}}} < (t-1). \tag{6}$$

This is a contradiction, because the right side of inequality (6) is constant in w, while the left side grows to infinity as w increases. $\square$

This approach can be generalized to give a lower bound on the number of copies of $K(t_1, \ldots, t_k)$ in a k-partite k-graph G with different part sizes [1], therefore upper bounding all generalized Zarankiewicz numbers.

3. Our Algorithm

Let $G = (V, E)$ be a k-graph with n vertices and m edges. We define $d = m/n^k$ as the *density* of G. We describe a polynomial-time algorithm that finds a complete balanced k-partite k-graph in G with part sizes

$$t = t(n, d, k) = \left\lfloor \left(\frac{\log(n/2^{k-1})}{\log(3/d)} \right)^{\frac{1}{k-1}} \right\rfloor, \tag{7}$$

with the only assumption that $t \geq 2$ (otherwise, we may just select a set of k vertices forming an edge in G). More precisely, we show the following:

Theorem 3.1. *There is an algorithm that, given a k-graph G satisfying the conditions above, finds a complete balanced k-partite k-graph in G with part sizes $t = t(n, d, k)$. That is, the algorithm returns a tuple of sets $(V_1, \dots, V_k) \subset \binom{V}{t}^k$ such that $V_1 \times \cdots \times V_k \subset E$. Furthermore, the algorithm's runtime is polynomial in n.*

Remark 3.2. The stated condition implies that the sets $V_1, \dots, V_k$ are disjoint: If, for example, $v \in V_1 \cap V_2$ and for $3 \leq i \leq k$ $v_i \in V_i$ then $(v, v, v_3, \dots, v_k) \in V_1 \times \cdots \times V_k$ has size $k - 1$ as an unordered set so it cannot be an edge in G. This means that the inclusion map from $K(V_1, \dots, V_k)$ to V defines an embedding, as desired.

This gives a constructive proof of Theorem 2.19. Indeed, suppose we have a fixed value for t. For n large enough, we may choose d such that $t(n, d, k) \geq t$. By our definition of t, we only need that

$$d \geq 3 \left(\frac{2^{k-1}}{n} \right)^{\frac{1}{t^{k-1}}},$$

which is satisfied for

$$m = \left\lceil 3n^k \left(\frac{2^{k-1}}{n} \right)^{\frac{1}{t^{k-1}}} \right\rceil = O\left(n^{k - \frac{1}{t^{k-1}}} \right).$$

The construction in Theorem 3.1 then proves

$$ex(n, K(t, \overset{k}{\dots}, t)) < m = O\left(n^{k - \frac{1}{t^{k-1}}} \right).$$

For $k = 2$, this problem was already solved by an algorithm of Mubayi and Turán [5], which we present here (Algorithm 1) for context and clarity. A slightly different value for t is used because of different estimates in their proof of correctness. Specifically, t is set to

$$t_2(n, d) = \left\lfloor \frac{\log(n/2)}{\log(2e/d)} \right\rfloor,$$

whereas we get

$$t(n, d, 2) = \left\lfloor \left(\frac{\log(n/2)}{\log(3/d)} \right) \right\rfloor.$$

The vertex set $V(G)$ is partitioned into two sets U and W such that there are many edges between them and the size of W is logarithmic in n. This is achieved by selecting W to be a set of vertices of highest degree (that is, no vertex in U has a higher degree than any vertex in W). Then, by iterating over all t-subsets of W, such a set T is found satisfying that the set S of common neighbors of T in U has size at least t. In other words, $S \times T \subset E$ for $S, T \subset V$ of size at least t.

Algorithm 1 Finding a balanced bipartite graph in a 2-graph

Require: A graph $G = (V, E)$ with $|V| = n$, $E = m$
1: $d \leftarrow m/n^2$
2: **assert** $d \geq 3n^{-1/2}$
3: $t \leftarrow \left\lfloor \frac{\log(n/2)}{\log(2e/d)} \right\rfloor$, $w \leftarrow \lfloor t/d \rfloor$
4: $W \leftarrow$ a set of w vertices with highest degree in G
5: $U \leftarrow V \setminus W$
6: **for all** $T \in \binom{W}{t}$ **do**
7:     $S \leftarrow \{x \in U : \{x, y\} \in E \text{ for all } y \in T\}$
8:     **if** $|S| \geq t$ **then**
9:         **return** (S, T)
10:     **end if**
11: **end for**

The minimum density $d \geq 3n^{-1/2}$ in line 2 is required because if $d = o\left(n^{-1/2}\right)$ then there may not even be a $K(2,2)$ in G. If the set S is too large, a subset of it of size t can be returned instead. To see that the algorithm returns a pair of sets (S, T), one uses the fact that there is large number of edges between U and W (proportional to the size of W). Then, a direct application of Theorem 2.17 with $u = |U| = n - w$ and $s = t$ shows that there is a $K(t, t)$ in the bipartite graph $(U, W; E \cap (U \times W))$. This in turn means that for some T, the size of S is at least t and the algorithm returns (S, T). Finally, the algorithm runs in polynomial time because the number of iterations of the loop is

$$\binom{w}{t} \leq \left(\frac{ew}{t}\right)^t \leq \left(\frac{1}{d}\right)^t e^t < e^{t \log(1/d) + \log n} < e^{2 \log n} = n^2.$$

We now present Algorithm 2, which is a generalization of Algorithm 1 to k-graphs. It follows the same structure as Algorithm 1, but it is defined recursively. This is the algorithm mentioned in Theorem 3.1, and the main contribution of this work.

The main idea is to select a set $W \subset V$ of vertices of highest degree with

$$|W| = w = w(n, d, k) = \left\lceil \frac{2t(n, d, k)}{d} \right\rceil. \tag{8}$$

Then, for every t-subset T of W, we compute the set S of $(k-1)$-subsets of $V \setminus W$ that form an edge with every vertex in T. For a specific T, the set S satisfies

$$|S| \geq s = s(n, d, k) = \left\lceil d^{t(n,d,k)} n^{k-1} \right\rceil. \tag{9}$$

We define a new $(k-1)$-graph G' with vertex set $V \setminus W$ and edge set S. As it turns out, S is large enough (9) that applying the algorithm recursively to G' yields a $K(t', \overset{k-1}{\ldots}, t')$ in G' with $t' \geq t$. That is, a tuple $P' = (V_1, V_2, \ldots, V_{k-1}) \in \mathcal{P}(V \setminus W)^{k-1}$ such that $|V_i| = t'$ and $V_1 \times \cdots \times V_{k-1} \subset S$.

If we now concatenate P' with T (choosing a subset of $X_i \subset V_i$ of size t for each i if necessary), we get a k-tuple $(X_1, \dots, X_{k-1}, T)$, of t*−sets* of V which by the definition of S satisfies $X_1 \times \cdots \times X_{k-1} \times T \subset E = E(G)$ so it forms a $K(t, \overset{k}{\ldots}, t)$ in G.

Algorithm 2 Finding a balanced partite k-graph in a k-graph

1: **function** $\mathrm{FIND\_PARTITE}(G, k)$
2:     **assert** G is a k-graph
3:     **if** $k = 1$ **then**
4:         **return** $(\{x : \{x\} \in E(G)\})$
5:     **end if**
6:     $V \leftarrow V(G), E \leftarrow E(G), n \leftarrow |V|,\ m \leftarrow |E|,\ d \leftarrow m/n^k$
7:     $t \leftarrow t(n, d, k),\ w \leftarrow w(n, d, k),\ s \leftarrow s(n, d, k)$
8:     **assert** $t \geq 2$
9:     $W \leftarrow$ a set of w vertices with highest degree in G
10:     $U \leftarrow \binom{V \setminus W}{k-1}$
11:     **for all** $T \in \binom{W}{t}$ **do**
12:         $S \leftarrow \{\, y \in U \colon \{x\} \cup y \in E$ for all $x \in T\,\}$
13:         **if** $|S| \geq s$ **then**
14:             $G' \leftarrow (V \setminus W, S)$
15:             $(V_1, \dots, V_{k-1}) \leftarrow \mathrm{FIND\_PARTITE}(G', k-1)$
16:             **return** $(V_1, \dots, V_{k-1}, T)$
17:         **end if**
18:     **end for**
19: **end function**

The implementation of the algorithm and its proof of correctness are less cumbersome if we assume a 1-graph to be just a subset of a set and use it as the base case. We also make the simplification of not including in Algorithm 2 the size reduction of the sets obtained from the recursive call. The algorithm as stated in fact returns a complete k-partite subgraph with part sizes *at least* t, which can easily be post-processed if desired to get a complete *balanced* subgraph with part sizes t.

The aim of the rest of this section is to prove that this algorithm is correct (as long as the condition $t \geq 2$ in line 8 is met on the first call) and runs in polynomial time. That is, to prove it meets the requirements of Theorem 3.1. From now on, we assume $k \geq 2$ and $t \geq 2$, unless stated otherwise. The following observation is useful for some of the bounds we have to prove.

Remark 3.3. The requirement $t \geq 2$ is met whenever

$$d \geq 3 \cdot 2^{\frac{k-1}{2^{k-1}}} n^{-\frac{1}{2^{k-1}}},$$

However, d satisfies

$$d = \frac{m}{n^k} \leq \frac{\binom{n}{k}}{n^k} < \frac{1}{k!},$$

so we get a minimum value of n:

$$n > \left(k! \cdot 3 \cdot 2^{\frac{k-1}{2^{k-1}}}\right)^{2^{k-1}} \geq 72.$$

This also lets us prove the bound

$$d \geq 3\sqrt{\frac{2}{n}}$$

for all $k \geq 2$. We have already seen that this is the case for $k = 2$. For $k > 2$, suppose that the bound is not met. Then,

$$3n^{-\frac{1}{4}} \leq 3n^{-\frac{1}{2^{k-1}}} < d < 3\sqrt{\frac{2}{n}},$$

which by algebraic manipulation implies $n < 4$.

We start by proving that the selection of t, w, s in line 7 of Algorithm 2 is sound, in the sense that we only consider subsets of sizes smaller than the corresponding supersets.

Lemma 3.4. *For t, w, s as selected in line 7 of Algorithm 2, we have that $t \leq w \leq n$, $k - 1 \leq n - w$ and $s \leq \binom{n-w}{k-1}$.*

Proof. It is clear from the definitions that $w \geq t$. To see that $w \leq n$, we in fact show that $w < \frac{n}{2}$. If not, then

$$\frac{n}{2} \leq w = \left\lceil \frac{2t}{d} \right\rceil \leq 1 + \frac{2t}{d} < 1 + \frac{2\log(n/2)\sqrt{n}}{3} = 1 + \frac{n}{4}.$$

This implies that $n < 4$, in contradiction to Remark 3.3. It is also clear from Remark 3.3 that $n > 2k$ so we also have $k < n/2$. Therefore, $k + w < n/2 + n/2 = n$, which implies $k - 1 < n - w$, as we wanted to show.

Finally, for sake of contradiction, suppose $s > \binom{n-w}{k-1}$. By the definition of s (9) and the fact that $\binom{n-w}{k-1}$ is an integer, we have that $d^t n^{k-1} > \binom{n-w}{k-1}$. Then, using the fact that $w < \frac{n}{2}$,

$$\left(\frac{n}{2k}\right)^{k-1} \leq \left(\frac{n-w}{k-1}\right)^{k-1} \leq \binom{n-w}{k-1} < d^t n^{k-1},$$

which implies

$$\left(\frac{1}{2k}\right)^{k-1} < d^t \leq \left(\frac{1}{k!}\right)^2.$$

In the last inequality, we have used that $t \geq 2$ and that $d \leq \frac{1}{k!}$. Since $k!^2 \geq (2k)^{k-1}$ for all k, we have reached a contradiction. □

The next step is to show that there are many edges with exactly one vertex in W. More precisely, we have the following.

Lemma 3.5. *Given $W \subset V$ as defined in line 9 of Algorithm 2, There are at least $\frac{3}{2} dwn^{k-1}$ edges of G with exactly one vertex in W.*

Proof. The degree sum over V is kdn^k. By averaging, the degree sum over W is at least $\frac{w}{n} kdn^k = wkdn^{k-1}$. For $2 \leq j \leq n$, consider the contribution to this sum by edges with exactly j vertices in W. Each such edge contributes j to the sum, and there are at most $\binom{w}{j}\binom{n-w}{k-j} \leq \frac{w^j n^{k-j}}{j!} \leq \frac{w^j n^{k-j}}{j}$ of them. Thus, the total contribution of these edges is at most $w^j n^{k-j} \leq w^2 n^{k-2}$. The number of edges with only one vertex in W is then at least

$$wkdn^{k-1} - (k-1)w^2 n^{k-2} = dwn^{k-1}\left(k - \frac{(k-1)w}{nd}\right).$$

Suppose, by way of contradiction, that $k - \frac{(k-1)w}{nd} < \frac{3}{2}$. Using that $\frac{k-1}{k-3/2} \leq 2$ for $k \geq 2$, we arrive at

$$2 \geq \frac{nd}{w},$$

which implies

$$d \leq \frac{2w}{n} = \frac{2\left\lceil \frac{2t}{d} \right\rceil}{n} < \frac{6t}{dn},$$

where the last inequality follows from the fact that $t > 1$ and $d \leq 1$. Algebraic manipulation then yields

$$nd^2 < 6t.$$

We now closely follow the steps of Mubayi and Turán [5].

If $3\sqrt{\frac{2}{n}} \leq d \leq 3\sqrt{\frac{\log n}{n}}$, we get

$$18 \leq nd^2 < 6t \leq 6\frac{\log(n/2)}{\log(3/d)} < 6\frac{\log n}{\log\left(\sqrt{\frac{n}{\log n}}\right)} = 12\frac{\log n}{\log\left(\frac{n}{\log n}\right)} < 12\frac{\log n}{\log\left(\frac{n}{\log n}\right)} < 12\frac{\log n}{\log\left(n^{2/3}\right)} = 18,$$

which is a contradiction.

Otherwise, we have $d > 3\sqrt{\frac{\log n}{n}}$. This yields $9 \log n \leq nd^2 < 6t < 6 \log n$, again, a contradiction. $\square$

We use this fact to show that for some $T \subset W$, there is a large number of $(k-1)$-subsets of $V \setminus W$ that form an edge with every vertex in T.

Lemma 3.6. *For some $T \in \binom{W}{t}$, the corresponding set S defined in line 12 of Algorithm 2 has size at least s.*

Proof. We apply Theorem 2.17 to the 2-partite 2-graph

$$\mathcal{P} = (U, W; F),$$

where F is defined as

$$F = \{(x, y) \in U \times W | \{x\} \cup y \in E\}.$$

By Lemma 3.5, $\mathcal{P}$ has at least $\frac{3}{2}dwn^{k-1}$ edges. By way of contradiction, suppose that the lemma is false. There are no sets $S \in \binom{U}{s}$, $T \in \binom{W}{t}$ such that $(x, y) \in E(\mathcal{P})$ for all $x \in S, y \in T$. In other words, there is no embedding of $K(s, t)$ in $\mathcal{P}$. By Theorem 2.17 applied with $u = \binom{n-w}{k-1}$, this implies that

$$\frac{3}{2}dwn^{k-1} \leq z\left(\binom{n-w}{k-1}; w, s, t\right) \leq (s-1)^{1/t}(w-t+1)\binom{n-w}{k-1}^{1-1/t} + (t-1)\binom{n-w}{k-1}.$$

We now substitute into the above expression $(s-1) \leq d^t n^{k-1}$ (which follows from $s = \lceil d^t n^{k-1} \rceil$) and $w > 0$:

$$\frac{3}{2}dwn^{k-1} < dn^{\frac{k-1}{t}} w\binom{n}{k-1}^{1-1/t} + t\binom{n}{k-1} \leq dn^{\frac{k-1}{t}} wn^{(k-1)(1-1/t)} + tn^{k-1}.$$

Finally, we substitute $t \leq \frac{1}{2}dw$, which follows from $w = \left\lceil \frac{2t}{d} \right\rceil$:

$$\frac{3}{2}dwn^{k-1} < dn^{\frac{k-1}{t}} wn^{(k-1)(1-1/t)} + \frac{1}{2}dwn^{k-1} = \frac{3}{2}dwn^{k-1},$$

which is a contradiction. $\square$

This shows that we reach the recursive call in line 14 of Algorithm 2 at some iteration of the loop in line 11. The next step will be to show that this recursive call finds a $k-1$-partite $k-1$-graph in G' of part sizes at least t. For this, we bound the density d' of G':

$$d' \geq \frac{s}{(n-w)^{k-1}} \geq \frac{d^t n^{k-1}}{n^{k-1}} = d^t,$$

and ensure that the associated part size

$$t' = t(n-w, d', k-1)$$

satisfies $t' \geq t$.

Lemma 3.7. *For all $k \geq 3$, $t' \geq t$.*

Proof. Substituting the new parameters into the definition, we get

$$t' = \left\lfloor \left(\frac{\log((n-w)/2^{k-2})}{\log(3/d')} \right)^{\frac{1}{k-2}} \right\rfloor .$$

We start by using that $d' \geq d^t$ and that $w \leq n/2$:

$$t' \geq \left\lfloor \left(\frac{\log((n-w)/2^{k-2})}{\log(3/d^t)} \right)^{\frac{1}{k-2}} \right\rfloor \geq \left\lfloor \left(\frac{\log(n/2^{k-1})}{\log(3/d^t)} \right)^{\frac{1}{k-2}} \right\rfloor = \left\lfloor \left(\frac{\log(n/2^{k-1})}{\log 3 - t\log d} \right)^{\frac{1}{k-2}} \right\rfloor .$$

Then, we substitute the definition of t, where removing the floor function maintains the inequality because the right hand side is decreasing in t (recall $d \leq 1$):

$$t' \geq \left\lfloor \left(\frac{\log(n/2^{k-1})}{\log 3 - \left(\frac{\log(n/2^{k-1})}{\log(3/d)}\right)^{\frac{1}{k-1}} \log d} \right)^{\frac{1}{k-2}} \right\rfloor = \left\lfloor \left(\frac{\log(n/2^{k-1})^{1-\frac{1}{k-1}}}{\frac{\log 3}{\log(n/2^{k-1})^{\frac{1}{k-1}}} - \frac{\log d}{\log(3/d)^{\frac{1}{k-1}}}} \right)^{\frac{1}{k-2}} \right\rfloor . \tag{10}$$

Now we argue that $n/2^{k-1} \geq 3/d$. Otherwise, by Remark 3.3, we would have

$$\frac{3}{n^{\frac{1}{2^{k-1}}}} \leq d < \frac{3 \cdot 2^{k-1}}{n},$$

which implies

$$\sqrt{n} < n^{1-\frac{1}{2^{k-1}}} \leq 2^{k-1} < k!,$$

so that

$$n < k!^2,$$

against the minimum value of n in Remark 3.3.

This allows us to find a common denominator on the right side of (10):

$$t' \geq \left\lfloor \left(\frac{\log(n/2^{k-1})^{1-\frac{1}{k-1}}}{\frac{\log 3 - \log d}{\log(3/d)^{\frac{1}{k-1}}}} \right)^{\frac{1}{k-2}} \right\rfloor = \left\lfloor \left(\frac{\log(n/2^{k-1})^{1-\frac{1}{k-1}}}{\frac{\log(3/d)}{\log(3/d)^{\frac{1}{k-1}}}} \right)^{\frac{1}{k-2}} \right\rfloor = \left\lfloor \left(\frac{\log(n/2^{k-1})}{\log(3/d)} \right)^{\frac{1}{k-1}} \right\rfloor = t. \quad \square$$

This means that, assuming that the algorithm finds a $K(t', \overset{k-1}{\ldots}, t')$ in G' in the recursive call, it finds a $K(t, \overset{k}{\ldots}, t)$ in G. This argument only works if $k \geq 3$. For $k = 2$, the recursive call is handled by the base case in line 3 of Algorithm 2. Therefore, the part size of the (singleton) tuple returned by the recursive call is the number of (single-vertex) edges in G', which is at least s. To ensure that the algorithm returns a $K(t, t)$ in this case, it suffices to show the following.

Lemma 3.8. *For $k = 2$, Algorithm 2 finds $s \geq t$.*

Proof. By way of contradiction, suppose that $t > s$. Substituting $k = 2$ into $s = \lceil d^t n^{k-1} \rceil$, we get $t > \lceil d^t n \rceil$ which implies

$$t > d^t n \geq d^{\frac{\log n}{\log(3/d)}} n = 3^{\frac{\log n}{\log(3/d)}} (d/3)^{\frac{\log n}{\log(3/d)}} n \geq \frac{3^t}{n} n = 3^t,$$

which is false for all $t \geq 0$. □

All in all, we can now state our main theorem.

Theorem 3.9. *Algorithm 2 finds a balanced partite k-graph in a k-graph G with n vertices and $m = dn^k$ with part size $t(n, d, k)$ in polynomial time, as long as $t(n, d, k) \geq 2$.*

Proof. To prove the correctness of the algorithm, we proceed by induction on k. If $k = 2$, it follows from Lemmas 3.6 and 3.8. Indeed, the algorithm finds (V_1, T) with $|T| = t$ and $|V_1| \geq s \geq t$. Furthermore, V_1 is the set of vertices $x \in V \setminus W$ such that $\{x, y\} = \{x\} \cup \{y\} \in E$ for all $y \in T$. This means that $V_1 \times T \subset E(G)$.

If $k \geq 3$, Lemma 3.6 tells us that the algorithm reaches line 14 at some iteration of the loop. Furthermore, Lemma 3.7 tells us that the recursive call in line 14 has a part size $t' \geq t$. In particular, this means that $t' \geq 2$. Using the induction hypothesis for $k - 1$, this recursive call is successful and returns a tuple of sets $(X_1, X_2, \ldots, X_{k-1}) \in \mathcal{P}(V)^{k-1}$ such that $|X_i| \geq t(n-w, d', k-1) \geq t$ for all i and $X_1 \times \cdots \times X_{k-1} \subset E(G')$. However, by construction of G',

$$E(G') = S = \left\{ x \in \binom{V \setminus W}{k-1} : \{x\} \cup y \in E \text{ for all } y \in T \right\}.$$

That is, the tuple $(X_1, \ldots, X_{k-1}, T)$ returned in line 16 satisfies $X_1 \times \cdots \times X_{k-1} \times T \subset E = E(G)$, making the algorithm correct.

For the time complexity, note that all operations in the algorithm are in polynomial time, except for perhaps the for loop in line 11 and the recursive call in line 14.

We first argue that the for loop in line 11 runs in polynomial time. This is argued in [5], but we reproduce the argument here for completeness: As seen in [6], the t-sets of W can be enumerated in $O\left(\binom{w}{t}\right)$ steps. However, we can bound

$$\binom{w}{t} \leq \binom{2t/d + 1}{t} < \left(\frac{3et/d}{t}\right)^t = \left(\frac{3e}{d}\right)^t < e^{3t + t\log(1/d)} < e^{4\log n} = n^4.$$

Because there is only one recursive call, we can prove that it runs in polynomial time by induction on k. Clearly, if the algorithm runs in polynomial time for $k - 1$, it also runs in polynomial time for k. We can take as a base case $k = 1$, which has no recursive calls so it runs in polynomial time.

□

4. Bibliography

References

[1] Matías Azócar Carvajal, Giovanne Santos, and Mathias Schacht. Canonical ramsey numbers for partite hypergraphs. *arXiv preprint arXiv:2411.16218*, 2024.

[2] P. Erdös. On extremal problems of graphs and generalized graphs. *Israel Journal of Mathematics*, 2(3):183–190, September 1964.

[3] C. Hyltén-Cavallius. On a combinatorical problem. *Colloquium Mathematicae*, 6(1):61–65, 1958.

[4] T. Kővari, V. Sós, and Turán P. On a problem of k. zarankiewicz. *Colloquium Mathematicae*, 3(1):50–57, 1954.

[5] Dhruv Mubayi and György Turán. Finding bipartite subgraphs efficiently. *Information Processing Letters*, 110(5):174–177, 2010.

[6] Edward M Reingold, Jurg Nievergelt, and Narsingh Deo. *Combinatorial algorithms: theory and practice*. Prentice Hall College Div, 1977.