Finding Partite Hypergraphs Efficiently

Ferran Espuña Bertomeu

June 2025

Abstract

TODO

1 Introduction

Hypergraph Turán problems study how many edges a k-uniform hypergraph $H = (V, E)$ with n vertices can have without containing a specific subgraph G. The maximal such number is known as the *Turán number* $\mathrm{ex}(n, G)$. It is known [3] that $\mathrm{ex}(n, G) = o\left(\binom{n}{k}\right)$ if and only if G is k-partite, i.e., if its vertex set can be partitioned into k disjoint sets such that each edge contains exactly one vertex from each part. Kővári, Sós, and Turán [4] (for $k = 2$) and Erdős [2] (for any $k \geq 2$) established that

$$\mathrm{ex}(n, K(t, \overset{k}{\ldots}, t)) = \mathcal{O}\left(n^{k - \frac{1}{t^{(k-1)}}}\right), \tag{1}$$

where $K(t, \overset{k}{\ldots}, t)$ is the complete balanced k-partite k-graph with k parts of size t. Furthermore, if H is a k-graph with at least $d\binom{n}{k}$ edges for some constant $d > 0$, then it contains a $K(t, \overset{k}{\ldots}, t)$ with $t = c_d \log(n)^{1/(k-1)}$.

This result is non-constructive, meaning it guarantees the existence of such a subgraph but does not provide an efficient way to find it. Note that a simple brute-force search for a $K(t, \overset{k}{\ldots}, t)$ would involve checking all $\binom{n}{kt}$ vertex subsets, which is superpolynomial in n for $t = \Theta((\log n)^{1/(k-1)})$. Mubayi and Turán [5] developed a polynomial-time algorithm for the case $k = 2$, which reaches the stated order of magnitude for the subgraph part size. This paper extends their approach to the general case of k-uniform hypergraphs, reaching analogous results for $k \geq 3$. More concretely, we prove the following.

Theorem 1. *There is a deterministic algorithm that, given a k-graph H with n vertices and $m = dn^k$ edges, finds a complete balanced k-partite subgraph $K(t, \overset{k}{\ldots}, t)$ in polynomial time, where*

$$t = t(n, d, k) = \ldots$$

This value of t matches the order of magnitude from existence proofs. In fact, a probabilistic argument shows that it is the best possible up to a constant factor.

2 The algorithm

We present a recursive algorithm, `FindPartite`, that finds a $K(t, \overset{k}{\ldots}, t)$ in a given k-graph H. The core idea is to reduce the uniformity of the problem from k to $k - 1$ in each recursive step. The algorithm takes a k-graph H with n vertices and m edges as input. It first defines the target part size t, a small set size w, and a threshold edge count s for the recursive call, based on the input graph's parameters:

$$t = t(n,d,k) = \left\lfloor \left(\frac{\log n}{\log(6/d)} \right)^{\frac{1}{k-1}} \right\rfloor,$$
$$w = w(n,d,k) = \left\lceil \frac{4t}{d} \right\rceil, \text{ and}$$
$$s = s(n,d,k) = \left\lceil \left(\frac{d}{4}\right)^t \binom{n}{k-1} \right\rceil,$$

where $d = \frac{m}{\binom{n}{k}}$ is the edge density of H. The main steps are:

1. **Base Case ($k = 1$):** The edge set of a 1-graph is just a collection of vertices. Return the set of all vertices that are "edges".
2. **Select High-Degree Vertices:** Choose a set $W \subset V$ of w vertices with the highest degrees in H.
3. **Find a Dense Link Graph:** Iterate through all t-subsets $T \subset W$. For each T, consider the set S of all $(k-1)$-subsets of V that form a hyperedge with *every* vertex in T.
4. **Recurse:** As we prove further along using the Kővári–Sós–Turán theorem, for at least one choice of T, the resulting set S will be large ($|S| \geq s$). We form a new $(k-1)$-graph $H' = (V, S)$ and make a recursive call: `FindPartite(`H'`, ` $k-1$`)`.
5. **Construct Solution:** The recursive call returns $k-1$ parts $V_1, \ldots, V_{k-1}$ of size at least t. By construction, every choice of vertices from these parts forms an edge in H' with every vertex of T. Thus, $(T_1, \ldots, T_{k-1}, T)$ form the desired $K(t, \overset{k}{\ldots}, t)$ in the original graph H.

The pseudocode is given in Algorithm 1.

Algorithm 1 Finding a balanced partite k-graph

function FindPartite(H, k)
  if $k = 1$ **then**
    return $(\{x \colon \{x\} \in E(H)\})$
  end if
  $n \leftarrow |V(H)|$, $m \leftarrow |E(H)|$, $d \leftarrow \frac{m}{\binom{n}{k}}$
  $t \leftarrow t(n,d,k)$, $w \leftarrow w(n,d,k)$, $s \leftarrow s(n,d,k)$
  assert $t \geq 2$
  $W \leftarrow$ a set of w vertices with highest degree in H
  for all $T \in \binom{W}{t}$ **do**
    $S \leftarrow \{\, y \in \binom{V}{k-1} \colon \forall x \in T, \{x\} \cup y \in E(H) \,\}$
    if $|S| \geq s$ **then**
      $H' \leftarrow (V, S)$ ▷ H' is a $(k-1)$-graph
      $(V_1, \ldots, V_{k-1}) \leftarrow$ FindPartite($H', k-1$)
      return $(V_1, \ldots, V_{k-1}, T)$
    end if
  end for
end function

3 Analysis

We now presnt the proof of correctness and polynomial runtime for our algorithm. We assume $t \geq 2$ for our estimates to be easier. If $t < 2$, we may just return the vertices of any single edge in H.

3.1 Correctness

First, we will prove that in step 3 of the algorithm we indeed find a set $T \in \binom{W}{t}$ such that the associated set $S \subset \binom{V}{k-1}$ has size at least s. For this, consider the bipartite graph B with parts $\binom{V}{k-1}$ and W with edge set

$$\left\{ (x, y) \in \binom{V}{k-1} \times W \middle| x \cup \{y\} \in E \right\}.$$

The edges of B correspond to the the edges containing each vertex in W, so there are

$$z = \sum_{y \in W} d_H(y) \geq k \cdot m \cdot \frac{w}{n} = \frac{k \cdot w \cdot d \cdot \binom{n}{k}}{n} = w \cdot d \cdot \binom{n-1}{k-1}$$

of them, where the inequality follows from the fact that we have picked a set of w vertices with highest degree in H. The existance of a set $T \subset W$ as desired is equivalent to there being $T \subset W$ of size t and a set $S \subset \binom{V}{k-1}$ of size s such that the induced bipartite subgraph $B[S, T]$ is complete. To prove that this is the case, we use a version the Kővári–Sós–Turán theorem [4], which we state and prove here for completeness.

Lemma 2. *Let u, w, s, t be positive integers with $u \geq s$, $w \geq t$, and let B be a bipartite graph with parts W and U such that $|U| = u, |W| = w$. If B has more than*

$$(s-1)^{1/t}(w-t+1)u^{1-1/t} + (t-1)u$$

edges, then there are $T \subset W$ of size t and $S \subset U$ of size s such that the induced bipartite subgraph $B[T, S]$ is complete.

We apply this lemma with $u = \binom{n}{k-1}$. It can be chekced that our parameter satisfy the requirements $u \geq s$ and $w \geq t$. Furthermore, $w \leq n$ so the set W is well-defined. Suppose, by way of contradiction, that

$$w \cdot d \cdot \binom{n-1}{k-1} \leq (s-1)^{1/t}(w-t+1)\binom{n}{k-1}^{1-1/t} + (t-1)\binom{n}{k-1}.$$

Algebraic manipulation then shows that

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

where the first inequality follows from $n \geq 2k$, which follows from $t \geq 2$. Finally, since $t \leq \frac{w \cdot d}{4}$ by the definition of w, we obtain

$$\left(\frac{d}{4}\right)^t \binom{n}{k-1} < s - 1,$$

against the definition of s. We are now ready to prove that the algorithm returns a $K(t, \overset{k}{\ldots}, t)$. More precisely, we show the following.

Theorem 3. *For $k \geq 2$, Algorithm 1 returns a tuple $(V_1, \ldots, V_k)$ of disjoint sets $V_i \subset V(H)$ such that $|V_i| \geq t$ and $H[V_1, \ldots, V_k]$ is complete.*

Proof. We proceed by induction on k. For $k = 2$, the recursive call returns the common neighborhood V_1 of the vertices in T, which is obviously disjoint from T, so it only remains to check that $|V_1| \geq t$. Now, since by construction $|V_1| = |S| \geq s$, it is enough that

$$s = \left\lceil \left(\frac{d}{4}\right)^t n \right\rceil \geq \left(\frac{d}{4}\right)^{\frac{\log n}{\log(6/d)}} n = \frac{1}{n} \cdot \left(\frac{3}{2}\right)^{\frac{\log n}{\log(6/d)}} \cdot n \geq \left(\frac{3}{2}\right)^t \geq t.$$

For $k \geq 3$, we assume the inductive hypothesis holds for $k-1$. That is, if d' is the edge density of the $(k-1)$-graph H', then the recursive call returns a tuple $(V_1, \ldots, V_{k-1})$ of disjoint sets $V_i \subset V(H)$ such that $|V_i| \geq t' = t(n, d', k-1)$ and $H'[V_1, \ldots, V_{k-1}]$ is complete. By construction of H', this means that $H[V_1, \ldots, V_{k-1}, T]$ is complete, and that T is disjoint from $V_1, \ldots, V_{k-1}$ (as long as they are all nonempty). The final step in the proof is to show that $t' \geq t$ (and in particular, that $t' \geq 2$). By the definition of s, we have $d' \geq \left(\frac{d}{4}\right)^t$. Therefore,

$$t' \geq \left\lfloor \left(\frac{\log n}{\log\left(\frac{6}{(d/4)^t}\right)} \right)^{\frac{1}{k-2}} \right\rfloor \geq \left\lfloor \left(\frac{\log n}{\log 6 - t\log(d/4)} \right)^{\frac{1}{k-2}} \right\rfloor .$$

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 $d \leq 1$):

$$t' \geq \left\lfloor \left(\frac{\log n}{\log 6 - \left(\frac{\log n}{\log(6/d)}\right)^{\frac{1}{k-1}} \log(d/4)} \right)^{\frac{1}{k-2}} \right\rfloor = \left\lfloor \left(\frac{(\log n)^{\left(1-\frac{1}{k-1}\right)}}{\frac{\log 6}{(\log n)^{\frac{1}{k-1}}} - \frac{\log(d/4)}{\log(6/d)^{\frac{1}{k-1}}}} \right)^{\frac{1}{k-2}} \right\rfloor .$$

Note that $n \geq 6/d$, as otherwise we would have $t < 1$. This allows us to find a common denominator in the expression of the lower bound for t':

$$t' \geq \left\lfloor \left(\frac{(\log n)^{\left(1-\frac{1}{k-1}\right)}}{\frac{\log 6 - \log(d/4)}{\log(6/d)^{\frac{1}{k-1}}}} \right)^{\frac{1}{k-2}} \right\rfloor = \text{NO SE CANCELA FUUUUUUUUCK}$$

□

3.2 Complexity

TODO re-evaluate the complexity analysis.

4 Conclusion and Future Work

We have presented a deterministic, polynomial-time algorithm to find a large complete balanced k-partite subgraph in any sufficiently dense k-uniform hypergraph. This provides a constructive counterpart to a classical existence result by Erdős in extremal hypergraph theory.

Several avenues for future research remain open.

- **General Blow-ups:** Our algorithm finds a blow-up of a single edge, $K(t, \overset{k}{\ldots}, t)$. Can this framework be adapted to find a t_n-blowup of an arbitrary fixed k-graph G? Existence theorems guarantee such structures, but efficient algorithms are lacking.

- **Unbalanced Partite Graphs:** The algorithm could be modified to search for unbalanced complete partite graphs $K(t_1, \ldots, t_k)$, where the part sizes may grow at different rates.

- **Optimality:** The bounds on t are asymptotically tight, but the constants can likely be improved with a more refined analysis. For $k = 2$, it is known that in dense graphs one can find a $t = \Theta(\log n)$ blow-up of any bipartite graph. It is an open question if a constructive proof for this stronger result exists for $k \geq 2$.

References

[1] B. Bollobás and P. Erdős. On the structure of edge graphs. *Bulletin of the London Mathematical Society*, 5(3):317–321, 1973.

[2] P. Erdős. On extremal problems of graphs and generalized graphs. *Israel Journal of Mathematics*, 2:183–190, 1964.

[3] P. Keevash. Hypergraph Turán problems. In *Surveys in combinatorics 2011*, volume 392 of *London Mathematical Society Lecture Note Series*, pages 83–139. Cambridge University Press, 2011.

[4] T. Kővári, V. T. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloquium Mathematicum*, 3:50–57, 1954.

[5] D. Mubayi and G. Turán. Finding bipartite subgraphs efficiently. *arXiv preprint arXiv:0905.2527*, 2009.