Finding Partite Hypergraphs Efficiently

Ferran Espuña[a]

Submitted: Jan 1, 2024; Accepted: Jan 2, 2024; Published: TBD

Abstract

TODO

Mathematics Subject Classifications: 05D99

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 [2] 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 [3] (for $k = 2$) and Erdős [1] (for any $k \geqslant 2$) established that

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

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 [4] 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 $k \geqslant 3$. More concretely, we prove the following.

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

$$t = t(n, d, k) = \left\lfloor \left(\frac{\log n}{\log(16/d)}\right)^{\frac{1}{k-1}} \right\rfloor$$

[a] ferranespuna@gmail.com.

 https://doi.org/10.37236/00000

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:

$$
\begin{aligned}
t = t(n, d, k) &= \left\lfloor \left(\frac{\log n}{\log(16/d)} \right)^{\frac{1}{k-1}} \right\rfloor, \\
w = w(n, d, k) &= \left\lceil \frac{4t}{d} \right\rceil, \text{ and} \\
s = s(n, d, k) &= \left\lceil \left(\frac{d}{4} \right)^t \binom{n}{k-1} \right\rceil,
\end{aligned}
$$

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

1. **Base Case ($\boldsymbol{k = 1}$):** 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 ($|S| \geqslant s$). 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

1: **function** FINDPARTITE(H, k)
2:     **if** $k = 1$ **then**
3:         **return** $(\{x \colon \{x\} \in E(H)\})$
4:     **end if**
5:     $n \leftarrow |V(H)|$, $m \leftarrow |E(H)|$, $d \leftarrow \frac{m}{\binom{n}{k}}$
6:     $t \leftarrow t(n, d, k)$, $w \leftarrow w(n, d, k)$, $s \leftarrow s(n, d, k)$
7:     **assert** $t \geqslant 2$
8:     $W \leftarrow$ a set of w vertices with highest degree in H
9:     **for all** $T \in \binom{W}{t}$ **do**
10:         $S \leftarrow \{\, y \in \binom{V}{k-1} \colon \forall x \in T, \{x\} \cup y \in E(H) \,\}$
11:         **if** $|S| \geqslant s$ **then**
12:             $H' \leftarrow (V, S)$ $\triangleright$ H' is a $(k-1)$-graph
13:             $(V_1, \ldots, V_{k-1}) \leftarrow$ FINDPARTITE$(H', k-1)$
14:             **return** $(V_1, \ldots, V_{k-1}, T)$
15:         **end if**
16:     **end for**
17: **end function**

3 Analysis

We now present the proof of correctness and polynomial runtime for our algorithm. We assume $t \geqslant 2$ 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 edges containing each vertex in W, so there are

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

of them, where the inequality follows from the fact that we have picked a set of w vertices with highest degree in H. The existence 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 [3], which we state and prove here for completeness.

Lemma 2. *Let u, w, s, t be positive integers with $u \geqslant s$, $w \geqslant t$, 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 checked that our parameter satisfy the requirements $u \geqslant s$ and $w \geqslant t$. Furthermore, $w \leqslant n$ so the set W is well-defined. Suppose, by way of contradiction, that

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

Algebraic manipulation then shows that

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

where the first inequality follows from $n \geqslant 2k$, which follows from $t \geqslant 2$. Finally, since $t \leqslant \frac{w \cdot d}{4}$ 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 $k \geqslant 2$, if $t \geqslant 2$, Algorithm 1 returns a tuple $(V_1, \ldots, V_k)$ of disjoint sets $V_i \subset V(H)$ such that $|V_i| \geqslant t$ 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 $|V_1| \geqslant t$. Now, since by construction $|V_1| = |S| \geqslant s$, it is enough that

$$s = \left\lceil \left(\frac{d}{4}\right)^t n \right\rceil \geqslant \left(\frac{d}{4}\right)^{\frac{\log n}{\log(16/d)}} n = \frac{1}{n} \cdot 4^{\frac{\log n}{\log(16/d)}} \cdot n \geqslant 4^t > t.$$

For $k \geqslant 3$, we assume the inductive hypothesis holds for $k-1$. If d' is the edge density of the $(k-1)$-graph H' and $t' = t(n, d', k-1)$, as long as $t' \geqslant 2$, the recursive call returns a tuple $(V_1, \ldots, V_{k-1})$ of disjoint sets $V_i \subset V(H)$ such that $|V_i| \geqslant t'$ and $H'[V_1, \ldots, V_{k-1}]$ is complete.

We claim that $t' \geqslant t$. This implies that $t' \geqslant 2$ so we get to apply the inductive hypothesis to H'. Furthermore, the sets V_i are at least as large as desired. By construction of H', for all $(x_1, \ldots x_{k-1}, y) \in V_1 \times \cdots \times V_{k-1} \times T$, we have that $\{x_1, \ldots, x_{k-1}, y\} \in E(H)$.

In particular, because all V_i are nonempty, this implies that T is disjoint from each of them. Furthermore, $H[V_1, \ldots, V_{k-1}, T]$ is complete, finishing the proof.

Let us now prove the claim that $t' \geqslant t$. By the definition of s, we have $d' \geqslant \left(\frac{d}{4}\right)^t$. Therefore,

$$t' \geqslant \left\lfloor \left(\frac{\log n}{\log\left(\frac{16}{(d/4)^t}\right)} \right)^{\frac{1}{k-2}} \right\rfloor \geqslant \left\lfloor \left(\frac{\log n}{\log 16 - t\log(d/4)} \right)^{\frac{1}{k-2}} \right\rfloor.$$

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 $d \leqslant 1$):

$$t' \geqslant \left\lfloor \left(\frac{\log n}{\log 16 - \left(\frac{\log n}{\log(16/d)}\right)^{\frac{1}{k-1}} \log(d/4)} \right)^{\frac{1}{k-2}} \right\rfloor = \left\lfloor \left(\frac{(\log n)^{\left(1-\frac{1}{k-1}\right)}}{\frac{\log 16}{(\log n)^{\frac{1}{k-1}}} - \frac{\log(d/4)}{\log(16/d)^{\frac{1}{k-1}}}} \right)^{\frac{1}{k-2}} \right\rfloor.$$

If we bound the denominator by showing that

$$\frac{\log 16}{(\log n)^{\frac{1}{k-1}}} - \frac{\log(d/4)}{\log(16/d)^{\frac{1}{k-1}}} \leqslant (\log(16/d))^{\left(1-\frac{1}{k-1}\right)}, \tag{1}$$

then the expression simplifies to

$$t' \geqslant \left\lfloor \left(\frac{(\log n)^{\left(1-\frac{1}{k-1}\right)}}{(\log(16/d))^{\left(1-\frac{1}{k-1}\right)}} \right)^{\frac{1}{k-2}} \right\rfloor = \left\lfloor \left(\frac{\log n}{\log(16/d)} \right)^{\frac{1}{k-2}\left(1-\frac{1}{k-1}\right)} \right\rfloor = \left\lfloor \left(\frac{\log n}{\log(16/d)} \right)^{\frac{1}{k-1}} \right\rfloor = t,$$

as desired. Suppose, by way of contradiction, that Inequality (1) does not hold. We can rewrite

$$(\log(16/d))^{\left(1-\frac{1}{k-1}\right)} = \frac{\log(16/d)}{\log(16/d)^{\frac{1}{k-1}}}$$

and rearrange the inequality to obtain

$$\frac{\log 16}{(\log n)^{\frac{1}{k-1}}} > \frac{\log(16/d) + \log(d/4)}{\log(16/d)^{\frac{1}{k-1}}} = \frac{\log 4}{(\log(16/d))^{\frac{1}{k-1}}}.$$

This implies that

$$t \leqslant \left(\frac{\log n}{\log(16/d)}\right)^{\frac{1}{k-1}} < \frac{\log 16}{\log 4} = 2,$$

which contradicts the assumption that $t \geqslant 2$. □

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 $k \geqslant 2$.

Acknowledgements

The ideas in this paper stem from the author's master's thesis at Universitat Politècnica de Catalunya, under the supervision of Dr. Richard Lang. The author thanks Dr. Lang for suggesting the problem and for his guidance and support throughout the project.

References

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

[2] 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.

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

[4] D. Mubayi and G. Turán. Finding bipartite subgraphs efficiently, *Information Processing Letters 2010*, 110(5), 174–177. doi:10.1016/j.ipl.2009.11.015.