FINDING PARTITE HYPERGRAPHS EFFICIENTLY

FERRAN ESPUÑA

ABSTRACT. We provide a deterministic polynomial-time algorithm that, for a given k-uniform hypergraph H with n vertices and edge density d, finds a complete k-partite subgraph of H with parts of size at least $c(d,k)(\log n)^{1/(k-1)}$. This generalizes work by Mubayi and Turán on bipartite graphs. The value we obtain for the part size matches the order of magnitude guaranteed by the non-constructive proof due to Erdős and is tight up to a constant factor.

1. INTRODUCTION

Hypergraph Turán problems concern 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 [5] that $\mathrm{ex}(n, G)$ is sublinear in $\binom{n}{k}$ 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 [6] (for $k = 2$) and Erdős [3] (for any $k \geq 2$) established that

$$\mathrm{ex}(n, K(t, \overset{k}{\ldots}, t)) \leq \binom{n}{k} \cdot n^{-\frac{1}{t^{(k-1)}}} .$$

This result implies the following.

Remark 1.1. *If H is a k-graph with at least $d\binom{n}{k}$ edges for some constant $d > 0$, then it contains $K(t, \overset{k}{\ldots}, t)$ as a subgraph, with $t = c(d,k)(\log(n))^{1/(k-1)}$.*

Furthermore, the order of magnitude of t is tight up to a constant factor: For some constant $\hat{c}(d,k) > 0$, there are k-graphs with n vertices and edge density d that do not contain $K(\hat{t}, \overset{k}{\ldots}, \hat{t})$ as a subgraph, where $\hat{t} = \hat{c}(d,k)\log(n)^{1/(k-1)}$. This was already noted by Erdős [3] and can be proved via the random alteration method [1].

Due to the fundamental role of Erdős' result, it is natural to ask whether a fast search algorithm for a complete k-partite subgraph of the size stated in Remark 1.1 exists. A 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 $t = \Omega\left((\log n)^{1/(k-1)}\right)$. For $k = 2$, Mubayi and Turán [7] developed a deterministic polynomial-time algorithm which reaches the stated order of magnitude for the subgraph part size. This work 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 1.2. *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*

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

This value of t matches the order of magnitude from Remark 1.1.

E-mail address: `ferran.espuna@gmail.com`.

2. The algorithm

We present a recursive algorithm, `FindPartite`, that finds a $K(t, \overset{k}{\dots}, t)$ 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 ($k = 1$):** The edge set of a 1-graph is just a collection of singleton sets 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 is 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 $V_1, \dots, V_{k-1}$ 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, $(V_1, \dots, V_{k-1}, T)$ form the desired $K(t, \overset{k}{\dots}, t)$ 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**
            $H' \leftarrow (V, S)$ $\triangleright$ H' is a $(k-1)$-graph
            $(V_1, \dots, V_{k-1}) \leftarrow$ FindPartite($H', k-1$)
            return $(V_1, \dots, V_{k-1}, T)$
        end if
    end for
end function

---

3. Proof of correctness

We now present the proof that Algorithm 1 is correct in the sense that all the steps are well-defined and that it returns a tuple $(V_1, \dots, V_k)$ of disjoint sets $V_i \subset V(H)$ of size at least t spanning a complete

k-partite subgraph in H. 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.

It is not immediately clear that the set W defined in step 2 of the algorithm is well-defined, as for this it is necessary that $w \leq n$. To show this, we first observe that our assumption $t \geq 2$ implies that $1 \geq d \geq \frac{16}{\sqrt{n}}$. Suppose, by way of contradiction, that $w > n$. Then, we have

$$n \leq w - 1 \leq \frac{4t}{d} \leq \frac{4 \log n}{d \log(16/d)} \leq \frac{4\sqrt{n} \log n}{16 \log(16/d)}.$$

Taking the logarithms to be in base e, we note that $\log x \leq \sqrt{x}$ for all positive x, and that $\log(16/d) > 1$. Therefore, we get $n < \frac{n}{4}$, which is a contradiction.

Next, we 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. That is, Algorithm 1 reaches line 13 at some point in the for loop. 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

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

of them, where the inequality follows from the fact that we have picked a set of w vertices with highest degree in H. The 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 the following version [4] of the Kővári–Sós–Turán theorem [6].

Lemma 3.1. *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 $B[S, T]$ is complete.

We apply the lemma with $u = \binom{n}{k-1}$. It is clear from the definitions that our parameter satisfy the requirements $u \geq s$ and $w \geq t$. Suppose, by way of contradiction, that

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

Dividing by $\binom{n}{k-1}$ then shows that

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

where the first inequality follows from $n \geq 16^{2^{k-1}} > 2(k-1)$, which follows from $t \geq 2$ and $d \leq 1$. Finally, since $t \leq \frac{w \cdot d}{4}$ by the definition of w, we obtain

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

in contradiction to the definition of s. So far, we have shown that the sets defined in Algorithm 1 are well-defined and that the recursive call in line 13 is reached. We are now ready to prove that the algorithm returns a $K(t, \overset{k}{\ldots}, t)$ by examining what happens in the recursive call. More precisely, we show the following.

Theorem 3.2. *For $k \geq 2$, if $t \geq 2$, 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

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

For $k \geq 3$, 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 $t' \geq 2$, the recursive call returns a tuple $(V_1, \dots, V_{k-1})$ of disjoint sets $V_i \subset V(H)$ such that $|V_i| \geq t'$ and $H'[V_1, \dots, V_{k-1}]$ is complete.

We claim that $t' \geq t$. This implies that $t' \geq 2$ so we get to apply the inductive hypothesis to H'. Furthermore, the sets V_i that we obtain when applying the algorithm to H' are of size at least $t' \geq t$. By construction of S, all the edges in H' are disjoint from T, and therefore so are the sets V_i. This means that the sets $V_1, \dots, V_{k-1}, T$ are disjoint. In addition, for all $(x_1, \dots x_{k-1}, y) \in V_1 \times \cdots \times V_{k-1} \times T$, we have that $\{x_1, \dots, x_{k-1}\} \in S$ so $\{x_1, \dots, x_{k-1}, y\} \in E(H)$. Equivalently, $H[V_1, \dots, V_{k-1}, T]$ is complete, finishing the proof.

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

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

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

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

We claim that we bound the denominator by showing that

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

Then, the expression simplifies to

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

as desired. Let us prove Inequality (1). Suppose, by way of contradiction, that it 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

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

which contradicts the assumption that $t \geq 2$. $\square$

4. Proof of Polynomial Complexity

We now analyze the computational complexity of Algorithm 1 to show that it runs in time polynomial in n for any fixed uniformity k. We consider the input hypergraph H to be given as an adjacency k-dimensional tensor. We make the assumption that we have a machine with arbitrarily large random-access memory, and that the operations on integers and floating-point numbers take constant time. Not making these assumptions would only add logarithmic factors to the running time, which we ignore for simplicity.

Let $T_k(n)$ denote the worst-case running time of the function `FindPartite` when called on a k-uniform hypergraph with n vertices. The algorithm's structure gives a recurrence relation for $T_k(n)$. We first analyze the cost of the operations within a single call for a fixed k, excluding the recursive step.

Assuming that the input tensor consists of a flattened boolean array of size n^k, querying whether a set of k vertices forms an edge in H can be done in constant time. Therefore, calculating the number of edges m can be done in $\mathcal{O}\left(n^k\right)$ time by iterating through all $\binom{n}{k}$ possible edges. The parameters n, m, d, t, w, s are computed in constant time after this step.

The set W of w vertices with highest degrees can be constructed in time $\mathcal{O}\left(n^k\right)$, by, for example, creating an array with the degree of each vertex (in time $\mathcal{O}\left(n \cdot n^{k-1}\right) = \mathcal{O}\left(n^k\right)$), sorting it (in time $\mathcal{O}\left(n \log n\right)$), and taking the first w elements.

The subsets of t elements of W can be iterated through in time $\mathcal{O}\left(\binom{w}{t}\right)$ [8]. Similarly to the argument by Mubayi and Turán [7], we can bound this quantity by a polynomial in n. Indeed,

$$\binom{w}{t} \leq \left(\frac{ew}{t}\right)^t \leq \left(\frac{e(4t/d+1)}{t}\right)^t \leq \left(\frac{4e}{d} + \frac{e}{2}\right)^t \leq \left(\frac{4.5 \cdot e}{d}\right)^t < e^{t(2.6+\log(1/d))} = e^{2.6 \cdot t} e^{\log(1/d) \cdot t}.$$

For the first term, we use the fact that $t < (\log n)^{1/(k-1)} \leq \log n$ and so $e^{2.6 \cdot t} < e^{2.6 \cdot \log n} = n^{2.6}$. For the second term, we use the fact that $t < (\log(n)/\log(1/d))^{1/(k-1)} \leq \log(n)/\log(1/d)$, so $e^{\log(1/d) \cdot t} < n$. All in all, we have

$$\binom{w}{t} < n^{2.6} n = n^{3.6}.$$

In each iteration of the loop, the set S can be constructed in the following way. We initialize a flattened boolean array A of size n^{k-1}, with all entries set to `true`. We iterate through all $x \in \binom{[n]}{k-1}$ and $y \in T$ and set the entry corresponding to x to `false` if $x \cup \{y\}$ is not an edge in H. All in all, this takes $\mathcal{O}\left(tn^{k-1}\right) = \mathcal{O}\left(n^k\right)$ steps, and then counting the number of `true` entries in the array takes $\mathcal{O}\left(n^{k-1}\right)$ time. Therefore, the for loop (with the recursive call) takes $\mathcal{O}\left(\binom{w}{t} n^k\right) = \mathcal{O}\left(n^{k+3.6}\right)$ time.

Finally, when the condition $|S| \geq s$ is satisfied, the recursive call to `FindPartite` is made. We can pass the array A to the recursive call directly, and the recursive call takes time $T_{k-1}(n)$. Putting everything together, we have the recurrence relation $T_k(n) = T_{k-1}(n) + \mathcal{O}\left(n^{k+3.6}\right)$. This, together with the base case $T_1(n) = \mathcal{O}\left(n\right)$, gives us $T_k(n) = \mathcal{O}\left(n^{k+3.6}\right)$.

5. 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-blowup of an arbitrary fixed k-graph G when its Turán density is exceeded by a positive constant? For example, for $k = 2$, there are existence results for $t = \Omega\left(\log n\right)$ [2], but directly adapting Algorithm 1 would only yield t $= \mathcal{O}\left((\log n)^{1/(|V(G)|-1)}\right)$.
- **Unbalanced k-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.

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

References

[1] Noga Alon and Joel H Spencer. *The probabilistic method*. John Wiley & Sons, 2016.
[2] Béla Bollobás and Pál Erdős. On the structure of edge graphs. *The Bulletin of the London Mathematical Society*, 5:317–321, 1973.
[3] Pál Erdős. On extremal problems of graphs and generalized graphs. *Israel Journal of Mathematics*, 2:183–190, 1964.
[4] C. Hyltén-Cavallius. On a combinatorical problem. *Colloquium Mathematicum*, 6:59–65, 1958.
[5] Peter Keevash. Hypergraph Turán problems. In *Surveys in combinatorics 2011*, volume 392 of *London Math. Soc. Lecture Note Ser.*, pages 83–139. Cambridge Univ. Press, Cambridge, 2011.
[6] Thomas Kövari, Vera T. Sós, and Pál Turán. On a problem of K. Zarankiewicz. *Colloquium Mathematicum*, 3:50–57, 1954.
[7] Dhruv Mubayi and György Turán. Finding bipartite subgraphs efficiently. *Information Processing Letters*, 110(5):174–177, 2010.
[8] Edward M. Reingold, Jurg Nievergelt, and Narsingh Deo. *Combinatorial algorithms: theory and practice*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977.