Extending Mubayi and Turán's Algorithm to k-graphs

Ferran Espuña

Let G be an r-graph with n vertices and $m = dn^k$ edges. A polynomial time algorithm is given to find a $K_{q,\dots,q} \subset G$ for

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

As long as there are more than $n^{k-\frac{1}{2^{k-1}}}$ edges in G.

Note that this result is tight up to the constant $c(k,d)$, as proved in [1]. This result is a generalization of the result in 2-graphs by [3], and algorithm will be analogous to the one given there. This algorithm, referred to as `FIND_PARTITE`$(k,\cdot)$, involves the following steps:

1. Choose parameters q, r, s depending on n, k and d.
2. Find the set R of r vertices with the highest degree in G.
3. find a subset $Q \subset R$ with q vertices and a $S \subset T := \binom{[n]\setminus Q}{k-1}$ with s edges satisfying

$$\{x_1, x_2, \dots, x_k\} \in E(G)\ \forall \{x_2, \dots, x_k\} \in S,\ x_1 \in Q$$

4. The set S induces a $(k-1)$-graph G' on T. Evaluate `FIND_PARTITE`$(k-1, G')$ to find a $K_{q',\dots,q'}$ in G' (say, $H' = \{U_1, \dots, U_{k-1}\}$). It will turn out that $q' \geq q$, and because of the condition for S the k-partite subgraph $H = \{Q, V_1, \dots V_{k-1}\}$ is complete in G, where $V_i \subset U_i$ is a subset of size q. Return H.

For step 1, we will use the following formulas:

$$q(k,d) = \left\lfloor \left(\frac{\log n}{\log(2^k/d)} \right)^{\frac{1}{k-1}} \right\rfloor,\ r(k,d) = \left\lceil \frac{2q(k,d)}{d} \right\rceil,\ s(k,d) = \left\lfloor d^{q(k,d)} n^{k-1} \right\rfloor$$

The goal is to prove that the algorithm is successful and runs in polynomial time.

Lemma 1. *This selection of parameters is sound in the sense that $q \leq r \leq n$, $k-1 \leq n-r$ and $s \leq \binom{n-r}{k-1}$.*

Proof. $q \leq r$ is clear from the definition of r. Suppose by way of contradiction that $r > n$. ...

$\square$

Lemma 2. *With the above restrictions and choice of parameters, for $k \geq 3$, there are at least $\frac{3}{2}drn^{k-1}$ edges with exactly one vertex in R.*

Proof. The degree sum over $V(G)$ is kdn^k. Thus, by the pigeonhole principle, the degree sum over R is at least $\frac{r}{n}kdn^k = rkdn^{k-1}$. For $2 \leq j \leq n$, consider the contribution to this sum by edges with exactly j vertices in R. Each such edge contributes j to the sum, and there are at most $\binom{r}{j}\binom{n-r}{k-j} \leq \frac{r^j n^{k-j}}{j!} \leq \frac{r^j n^{k-j}}{j}$ of them. Thus, the total contrition of these edges is at most $r^j n^{k-j} \leq r^2 n^{k-2}$. The number of edges we want is then at least

$$rkdn^{k-1} - (k-1)r^2n^{k-2} = drn^{k-1}\left(k - \frac{(k-1)r}{nd}\right)$$

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

$$d < 2rn^{-1} \leq \frac{3q}{nd} \implies nd^2 < 3\log n$$

Applying our minimum density, this means

$$\sqrt{n} \leq n^{1-\frac{1}{2^{k-2}}} < 3\log n$$

which is false for all n. $\square$

Lemma 3. *For this selection of parameters, there exist sets Q, S as described in step 3 of the algorithm.*

Proof. Consider the biparite graph with vertex set $\left(R, \binom{T}{k-1}\right)$ and edges corresponding to edges in G with exactly one vertex in R (and thus all others in T). The sets Q and S we want to find correspond to a complete bipartite subgraph of this graph with parts of size q and s respectively. Suppose that such a subgraph does not exist. [2] tells us then that

$$\begin{aligned}\frac{3}{2}drn^{k-1} < z\left(\binom{n-r}{k-1}, r; s, q\right) &< (s-1)^{1/q}(r-q+1)\binom{n-r}{k-1}^{1-1/q} + (q-1)\binom{n-r}{k-1} \\ &\leq s^{1/q}r\binom{n}{k-1}^{1-1/q} + q\binom{n}{k-1} \leq s^{1/q}r\binom{n}{k-1}^{1-1/q} + \frac{1}{2}drn^{k-1}\end{aligned}$$

Where the last inequality follows from our choice of r.

Rearranging and approximating the binomial coefficient, we get

$$drn^{k-1} < s^{1/q} r n^{(k-1)(1-1/q)} \iff d < \left(\frac{s}{n^{k-1}}\right)^{1/q}$$

Which is false for the given choice of s.

$\square$

Lemma 4. *For this choice of parameters, the number of edges s in G' is such that we can apply the algorithm to G'. Furthermore, the resulting q' satisfies $q' \geq q$.*

Proof. First we calculate a lower bond for the corresponding edge density d' in G':

$$d' = \frac{s}{(n-r)^{k-1}} \geq \frac{\left(n^{k-1} d^q - 1\right)}{n^{k-1}} \geq \frac{\left(n^{k-1-\frac{q}{2^{k-1}}} - 1\right)}{n^{k-1}}$$

Therefore, we can bound q' as follows:

$$q' \geq \left(\frac{\log n}{(k-1)\log 2 - \left(\frac{\log n}{\log(2^k/d)}\right)^{\frac{1}{k-1}} \log d} \right)^{\frac{1}{k-2}} - 1$$

$\square$

References

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

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

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