Zarankiewicz problem / Finding colored $K_{2,\ell}$

congyu

July 29, 2025

Problem 1 (Zarankiewicz problem, algorithmic version...) *Given a groundset U and a collection $\mathcal{S} = \{S_1, \ldots, S_n\}$ of n subsets of U. Let $m = \sum_{i \in [n]} |S_i|$. Check if there exist k subsets which have exactly ℓ elements in common.*

$\alpha(G)$ is the arboricity of G. $\deg(v)$ is the degree of vertex v. $N = |U|$.

An equivalent formulation of Problem 1 on graph is the following,

Problem 2 (colored $K_{k,\ell}$) *Given a bipartite graph $G = (V_1 \sqcup V_2, E)$ where V_1, V_2 and E represent elements in U, sets in $\mathcal{S}$ and element membership respectively. Color vertices in V_1 red and vertices in V_2 blue. Check if there is an induced $K_{k,\ell}$ with k blue vertices and ℓ red vertices.*

1 Existing algorithms

1.1 subcubic combinatorial alg for detecting induced C_4

A recent paper https://arxiv.org/pdf/2507.18845v1.

2 Finding $K_{2,\ell}$

This has already been described in https://chaoxu.prof/posts/2019-01-21-high-degree-low-degree-technique.html.

Theorem 2.1 ([1] section 4) *There exists an algorithm which finds all $K_{2,\ell}$ in G with time complexity $O(m\alpha(G))$.*

Proof: The algorithm is described in Figure 2.1, This algorithm outputs all possible pairs (v, w) such that v and w have exactly ℓ common neighbors. The total number of vertices this algorithm visits is $\sum_v [\deg(v) + \sum_{u \in N(v)} (\deg(u) - 1)] = 2 \sum_{(u,v) \in E} \min\{\deg(u), \deg(v)\} \leq 4\alpha(G)m$. Thus this algorithm has running time $O(m\alpha(G))$. □

Theorem 2.2 *There exists an algorithm which solves Problem 1 and has complexity in $O(n^2\ell)$ when $k = 2$.*

Proof: There is an algorithm for finding axis-parallel rectangles in $O(m^{3/2})$ described in [3]. Now we show that this algorithm also finds colored $K_{2,\ell}$ in bipartite graphs and has complexity $O(m + n^2)$ with better analysis. We need to construct a data structure for the algorithm in Figure 2.2. We use a linked list to stored all elements in each S_i. This takes $O(M)$ times. For finding all S_j such that $j < i$ and $u \in S_j$, we store a linked list starting from each u and link u with the element representing u in S_k's linked list, where k is the largest value smaller than i that S_k contains u. These links can be built using bucket sort in $O(M)$ time.

For the complexity we notice that for each S_i we visit at most $O(n\ell)$ times since each counter $C[S_i]$ is at most ℓ. Thus the total running time is $O(M + n^2\ell) = O(n^2\ell)$. □

```
sort vertices in G in such that deg(v_1) ≥ ··· ≥ deg(v_n)
for i ∈ [n]:
   for each vertex u ∈ N(v_i):
      let U[v] = ∅ for all v.
      for each vertex w ∈ N(u) that is not v_i:
         add u to U[w].
   for all vertex w ∈ V that is not v_i:
      if |U[w]| = ℓ:
         output tuple (v_i, w, U[w])
   G = G − v_i
```

Figure 2.1. An $O(m\alpha(G))$ algorithm for finding all colored $K_{2,\ell}$

```
for i ∈ [n]:
   for S_k ∈ 𝒮:
      C[S_k] ← 0
   for u ∈ S_i:
      for all S_j s.t. j < i and u ∈ S_j:
         C[S_j] ← C[S_j] + 1
         if C[S_j] ≥ ℓ:
            output exist K_{2,ℓ}
output no K_{2,ℓ}
```

Figure 2.2. An $O(n^2\ell)$ algorithm for checking existence of a $K_{2,\ell}$

There is actually a much simpler algorithm for the more general problem...

Theorem 2.3 *There exist an algorithm which solves Problem 1 in $O(\ell n^k)$.*

Proof: Create a array A of length $\binom{n}{k}$, originally all zero. Note that the length of the array is $O(n^k)$. Each element in the array corresponds to a combination of k sets out of $\mathcal{S}$. We iterate over all elements in U and find for each element e the set of positions in A for which the corresponding set contains e. We add 1 to these positions. The process stops when there is some $A[i] \geq \ell$. We visit every position in A at most ℓ times. Thus the running time is $O(\ell n^k)$. □

Theorem 2.4 ([2]) *There exist a constant c such that each n vertex graph with more than $c\ell^{1/2}n^{3/2}$ must contain $K_{2,\ell}$.*

Now we compose these algorithms to get a better running time for testing if a given bipartite graph contains a colored $K_{2,\ell}$.

Lemma 2.5 *If $\alpha(G) > t$, then there is a subgraph $H \subseteq G$ such that* $\min\{\deg(v) : v \in V(H)\} \geq t$.

Proof: H can be found through repeatedly removing the vertex with minimum degree in G. Once the minimum degree is at least t, the subgraph induced by the remaining vertices has the minimum degree at least t. Now we need to show that H is nonempty given that $\alpha(G) > t$. First observe that for any vertex $v \in G$ if $\alpha(G-v) > \deg_G(v)$, then $\alpha(G) = \alpha(G-v)$. Thus if H is empty we would have $\alpha(G) \leq \min \deg(v) \leq t$, a contradiction. □

Theorem 2.6 *There is an algorithm which finds $K_{2,\ell}$ in time $O(l^{1/3}m^{4/3})$.*

Proof: We compose previous algorithm using low-degree high-degree technique.

```
CHECKEXISTANCEK_{2,ℓ}(G)
if t ≥ α(G)
   use Algorithm2.1
else
   find a subgraph H ⊆ G with min degree at least t
   if c|V(H)|^{3/2} ≤ |E[H]|
      there exists K_{2,ℓ} ∈ H, return True
   else
      run Algorithm2.2 on H
```

Figure 2.3. An $O(l^{1/3}m^{4/3})$ algorithm for checking existence of a $K_{2,\ell}$

Note that while finding H we repeatedly delete the vertex with min degree. Meanwhile we can check the existence of $K_{2,\ell}$ which is not contained in H. □

References

[1] Norishige Chiba and Takao Nishizeki. Arboricity and Subgraph Listing Algorithms. *SIAM Journal on Computing*, 14(1):210–223, February 1985. Publisher: Society for Industrial and Applied Mathematics.

[2] Zoltán Füredi. New Asymptotics for Bipartite Turán Numbers. *Journal of Combinatorial Theory, Series A*, 75(1):141–144, July 1996.

[3] Marc J. van Kreveld and Mark T. de Berg. Finding squares and rectangles in sets of points. In Manfred Nagl, editor, *Graph-Theoretic Concepts in Computer Science*, pages 341–355, Berlin, Heidelberg, 1990. Springer.