Tetrahedron-intersecting families of 3-uniform hypergraphs

Nitya Mani Owen Jianwen Zhang

May 7, 2025

Abstract

An H*-intersecting family* is a collection of (hyper)graphs $\mathcal{F}$ on a fixed underlying set of labeled vertices, such that for each pair $G_1, G_2 \in \mathcal{F}$, the intersection $G_1 \cap G_2$ contains a subgraph isomorphic to H. Understanding how large $\mathcal{F}$ can be for a given H is of great importance in extremal combinatorics and theoretical computer science. Ellis, Filmus, and Friedgut conjectured a tight upper bound on the size of a K_t-intersecting family, but only the cases of $t = 3$ and $t = 4$ have been resolved (by Ellis, Filmus, and Friedgut, and Berger and Zhao respectively). We resolve the case $t = 5$. We also give the first resolution of an analogous conjecture in the hypergraph setting, giving a tight bound on the size of a tetrahedron-intersecting family of 3-uniform hypergraphs.

1 Introduction

Intersection problems in theoretical computer science and extremal combinatorics have a rich history, with their study dating back to the early 1900s [Ell22]. Such questions broadly have the following flavor:

How large can a family $\mathcal{F}$ of subsets of some ground set Ω be, if for every pair of elements $G_1, G_2 \in \mathcal{F}$, the intersection $G_1 \cap G_2$ must satisfy some property?

One of the most basic intersecting families problems has a very simple solution: how large can a family $\mathcal{F}$ of subsets of $[n] := \{1, \ldots, n\}$ be so that every pair of sets in $\mathcal{F}$ has nonempty intersection? Taking all subsets of $[n]$ that contain $\{1\}$ gives an *intersecting family* $\mathcal{F}$ of size 2^{n-1}; this bound is tight, as can be seen by noticing that for any $S \subset [n]$, we cannot have both S and $\overline{S} = [n]\backslash S$ in $\mathcal{F}$. One of the oldest and one of the most famous results in the area is the Erdős-Ko-Rado theorem [EKR61], which gives a tight upper bound on the maximum size of an intersecting family of size k subsets of $\{1, \ldots, n\}$. Since the 1930s, a wide variety of intersection problems over the integers, permutations, graphs, groups, and other combinatorial objects have been heavily studied, but many basic questions remain open [EFF12, EKR61, CGFS86, Wil84, Kat64, FW81, FR87, DEF78, FW86, JT08, Fri08]. This work focuses on H*-intersecting families of (hyper)graphs*, one example of a heavily studied problem where basic, natural-sounding questions have remained unanswered for decades.

Definition 1.1 (H-intersecting family)**.** Let H be a fixed, unlabeled graph. A family $\mathcal{F}$ of graphs on n labeled vertices is H*-intersecting* if the intersection of any two graphs in $\mathcal{F}$ contains some subgraph isomorphic to H.

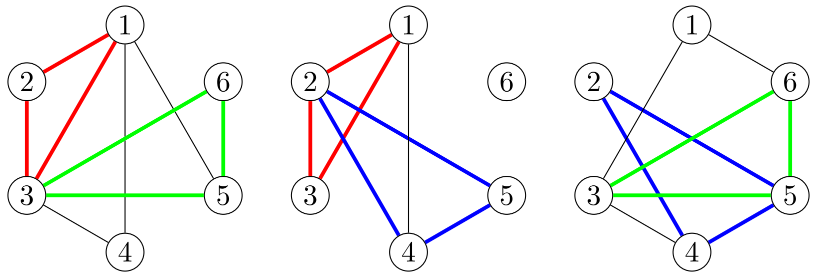

Figure 1: A triangle-intersecting family of graphs on 6 labeled vertices.

In 1976, Simonovits and Sós conjectured that the largest triangle-intersecting family of graphs on n labeled vertices has size at most $2^{\binom{n}{2}-3}$, tight only when every graph in the family contains some fixed triangle. We call such a family of graphs a $\triangle$-umvirate.

Definition 1.2 (H-umvirate)**.** A family $\mathcal{F}$ of graphs on vertex set $[n]$ is an *H-umvirate*, if $\mathcal{F}$ consists of all graphs on $[n]$ that contain a single fixed copy of H as a subgraph.

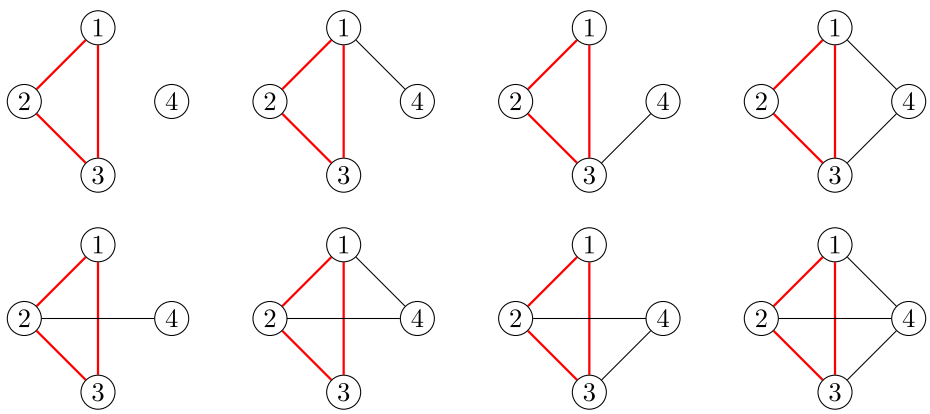

Figure 2: A triangle-intersecting family of graphs on 4 labeled vertices with size $2^{\binom{4}{2}-3}$, consisting of all graphs containing the triangle with vertices $1, 2$ and 3.

Little progress was made until 1984, when Chung, Graham, Frankl and Shearer [CGFS86] proved an upper bound of $2^{\binom{n}{2}-2}$ (falling short of the Simonovits-Sós conjecture by a factor of 2) by bounding the entropy of a random graph within a triangle-intersecting family. In the course of establishing this upper bound, they developed the ubiquitous *entropy method*, introducing *Shearer's inequality*, just one example of a technical contribution with outsized impact made while studying an intersection problem. Finally, in 2010, Ellis, Filmus, and Friedgut [EFF12] used Fourier analytic methods to prove the Simonovits-Sós conjecture, hence resolving the case where H is a triangle.

Theorem 1.3 (Theorem 1.4 in [EFF12], conjectured by Simonovits-Sós)**.** *Let $\mathcal{F}$ be a triangle-intersecting family of graphs on $[n] := \{1, 2, \ldots, n\}$. Then, $|\mathcal{F}| \leq 2^{\binom{n}{2}-3}$, and the above upper bound is an equality if and only if $\mathcal{F}$ is a $\triangle$-umvirate.*

For the case when H is a complete graph, Ellis, Filmus, and Friedgut [EFF12] formally posed the following generalization of the Simonovits-Sós conjecture:

Conjecture 1.4 (Subsection 6.3 in [EFF12])**.** *Let $\mathcal{F}$ be a K_t-intersecting family of graphs on $[n]$.*

- ***(Upper bound)*** $|\mathcal{F}| \leq 2^{\binom{n}{2}-\binom{t}{2}}$.
- ***(Uniqueness)*** *The above upper bound is an equality if and only if $\mathcal{F}$ is a K_t-umvirate.*

In 2021, Berger and Zhao [BZ23] extended the methods of Ellis, Filmus, and Friedgut [EFF12] to prove Conjecture 1.4 for $t = 4$ via a linear-programming approach. Both Ellis, Filmus, and Friedgut [EFF12] and Berger and Zhao [BZ23] gave additional stability results, showing that for $t = 3, 4$, nearly optimal K_t-intersecting families must be very *close* to a K_t-umvirate. It is natural to wonder if Conjecture 1.4 generalizes to *all* subgraphs H. Unfortunately, this is not the case, as in general an H-intersecting family can be a constant factor larger than an H-umvirate. As discussed in [BZ23], Noga Alon showed that for every fixed star forest H, the largest H-intersecting family on n vertices has size $(1 - o(1))2^{\binom{n}{2}-1}$ (a bound that is tight, as an H-intersecting family must be intersecting and thus cannot contain both a graph and its complement). He further posed the following conjecture.

Conjecture 1.5 (Noga Alon)**.** *There is a universal constant $c > 0$ such that for H not a star forest, the largest H-intersecting family on n vertices has size at most $(1 - c)2^{\binom{n}{2}-1}$.*

Given the resolution of triangle-intersecting families, it would suffice to verify this conjecture for $H = P_4$ a path with 3 edges, but it is known that the largest P_4-intersecting family is *not* always a P_4-umvirate (see Subsection 6.3 of [EFF12] for further discussion). Nonetheless, Conjecture 1.4 is widely believed to be true for all K_t, despite only being known to hold in the $t = 3$ and $t = 4$ settings. Verifying Conjecture 1.4 for K_4-intersecting families in [BZ23] involved computationally verifying all "small" graphs. This work additionally gave a framework that could *theoretically* verify Conjecture 1.4 for any $t \geq 3$ given enough computing power. However, going beyond $t = 4$ using their argument is quite difficult due to the rate at which the number of possible graphs grows (super-exponentially), causing their methods to require an infeasible amount of computing power for even $t = 5$.

Another natural generalization of the Simonovits-Sós conjecture is to consider higher-dimensional combinatorial structures. A 3-*uniform hypergraph* $G = (V, E)$ is given by a collection of vertices V and an edge set E that comprises a collection of unordered *triples* of vertices. As posed in [BZ23] and informally much earlier, one might wonder if a hypergraph analogue of Conjecture 1.4 holds.

Question 1.6. *For complete hypergraphs H, is the largest H-intersecting family of 3-uniform hypergraphs on n labelled vertices an H-umvirate? If H is complete, is $|\mathcal{F}| \leq 2^{\binom{n}{3}-e(H)}$ for any H-intersecting family of hypergraphs $\mathcal{F}$?*

Our main contribution in this work is to provide an affirmative answer to Question 1.6 for the smallest non-trivial example of such a hypergraph, the *tetrahedron* $K_4^{(3)}$, the complete 3-uniform hypergraph on 4 vertices (see Fig. 3).

Theorem 1.7. *Let $\mathcal{F}$ be a tetrahedron-intersecting family of 3-uniform hypergraphs on $[n]$. Then $|\mathcal{F}| \leq 2^{\binom{n}{3}-4}$, and the upper bound is an equality if and only if $\mathcal{F}$ is a tetrahedron-umvirate.*

Our approach to prove Theorem 1.7 also involves computationally verifying all "small" 3-uniform hypergraphs. However, there are heuristically many more small 3-uniform hypergraphs than graphs, due to each edge having $\binom{n}{3}$ possibilities as compared to $\binom{n}{2}$. Hence, it is especially important to

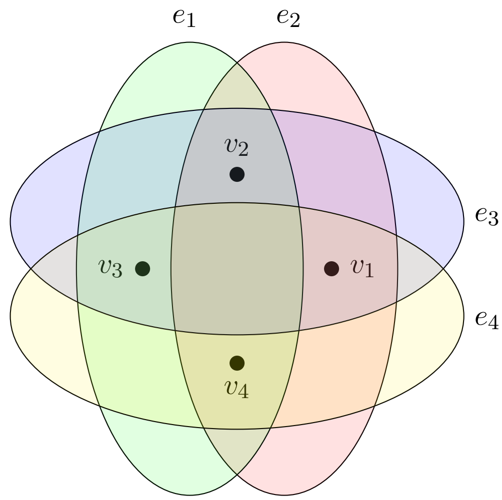

Figure 3: $K_4^{(3)}$, a tetrahedron.

show tight bounds which minimize the size of the final computation. As a result, the bulk of our paper consists of a careful graph-theoretic analysis of the intersections of tetrahedron-free 3-uniform hypergraphs, from which we derive bounds that ensure the resulting computation is feasible.

Along the way, we also resolve Conjecture 1.4 for $t = 5$, adding to the small collection of graphs H, for which Conjecture 1.4 is known to hold.

Theorem 1.8. *Let $\mathcal{F}$ be a K_5-intersecting family of graphs on $[n]$. Then, $|\mathcal{F}| \leq 2^{\binom{n}{2}-\binom{5}{2}}$, and the above upper bound is an equality if and only if $\mathcal{F}$ consists of all graphs containing some fixed K_5.*

Organization. In Section 2, we summarize the framework in [EFF12, BZ23] that reduces Theorem 1.7 to a linear program. In Section 3, we present our solution to the linear program in Section 2 and verify it for 3-uniform hypergraphs on up to 7 vertices. In Section 4, we verify our solution for 3-uniform hypergraphs on 8 to 13 vertices. In Section 5, we verify our solution for 3-uniform hypergraphs on 14 or more vertices, completing the proof of Theorem 1.7. We give a proof of Theorem 1.8 in Appendix A.

Notation. Throughout, we let $[n] := \{1, 2, \ldots n\}$. $G = (V, E)$ will always denote a graph or 3-uniform hypergraph, typically with $V = [n]$, and H will be a small *fixed* subgraph. All hypergraphs considered in this paper are 3-uniform, simple hypergraphs (with all distinct edges). We denote by K_t the complete graph on t vertices and by $K_t^{(3)}$ the complete hypergraph on t vertices. For any (hyper)graph G, let $V(G)$ be the vertex set of G with size $v(G) = |V(G)|$ and let $E(G)$ be the edge set of G with size $e(G) = |E(G)|$. Given (hyper)graphs $G_1 = ([n], E_1), G_2 = ([n], E_2)$ on the same vertex set, their *intersection* is the (hyper)graph $G_1 \cap G_2 = ([n], E_1 \cap E_2)$ comprising the edges present in both G_1 and G_2. Given two distinct vertices $v_1, v_2 \in G$, let their *codegree* $\operatorname{codeg}(v_1, v_2)$ be the number of edges $e \in G$ with $v_1, v_2 \in e$. Let $\Delta(G)$ be the maximum degree of any vertex in G, and let $\Delta_2(G)$ be the maximum codegree of any pair of vertices in G.

2 Reduction to a linear program

We begin by recalling the framework of [EFF12, BZ23] used to bound the size of K_3 and K_4-intersecting families. In [BZ23], much of this reduction is given in generality for K_t-intersecting families.

Definition 2.1 (Definition 2.4 [BZ23])**.** For a graph G on $[n]$, let $[q]^{[n]}$ be the set of maps $\varphi\colon V(G) \to [q]$, viewed as q-colorings of $[n]$ (not necessarily proper). For each coloring $\varphi\colon V(G) \to [q]$, define $\varphi(G)$ to be the subgraph of G formed by deleting all monochromatic edges of G, and then deleting all isolated vertices that result. Let G_q be the random graph $\varphi(G)$ given by choosing $\varphi \sim \mathrm{Unif}([q]^{[n]})$.

In particular, Definition 2.1 ensures that the random graph G_q never contains the complete graph $K_{(q+1)}$ as a subgraph, which is necessary in the proof of the following proposition—see [BZ23] for further details.

Proposition 2.2 (Proposition 2.5 in [BZ23])**.** *Suppose there exists $t \geq 3$ an integer, $\{H\}$ a set of unlabeled graphs, $\{c_H\}$ an associated set of coefficients with $c_\varnothing = 2^{\binom{t}{2}} - 1$, and $\delta > 0$ a constant, such that the function*

$$\mu(G) := (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_{(t-1)} \cong H]$$

satisfies the following conditions:

1. *$|\mu(G)| \leq 1$ whenever $1 \leq e(G) \leq \binom{t}{2}$.*
2. *$|\mu(G)| \leq 1 - \delta$ whenever $\binom{t}{2} < e(G)$.*

Let $\mathcal{F}$ be a K_t-intersecting family of graphs on $[n]$. Then,

- ***(Upper bound)** $|\mathcal{F}| \leq 2^{\binom{n}{2} - \binom{t}{2}}$.*
- ***(Maximal families)** The upper bound is an equality if and only if $\mathcal{F}$ is a K_t-umvirate.*

In their paper, Berger and Zhao found and verified a satisfactory function μ for Proposition 2.2 with $t = 4$ and $c_\varnothing = 63 = 2^{\binom{4}{2}} - 1$, hence proving that the maximal size of a K_4-intersecting family of graphs on $[n]$ is $2^{\binom{n}{2} - \binom{4}{2}}$.

We adapt Berger and Zhao's framework to bound the sizes of $K_4^{(3)}$-intersecting families of hypergraphs. To do so, we will need to devise an analogue of Definition 2.1 that is appropriate for the tetrahedron-intersecting setting.

Unfortunately, the most natural analogue—coloring vertices one of three colors and deleting all monochromatic edges does not ensure that the resulting hypergraph never contains $K_4^{(3)}$. This can be seen as $K_4^{(3)}$ itself can be colored with $1, 1, 2, 3$ in some order, resulting in no edges being deleted. So, we must study the following more complicated random subgraph process.

Definition 2.3. For a 3-uniform hypergraph G on $[n]$, let $[3]^{[n]}$ be the set of maps $\varphi\colon V(G) \to [3]$. For each $\varphi \in [3]^{[n]}$, define $\varphi(G)$ to be the subgraph of G formed by deleting all monochromatic edges of G and all edges whose vertex colors sum to 2 mod 3, and then deleting all isolated vertices. Define G_* to be the random hypergraph $\varphi(G)$ given by choosing φ uniformly at random from $[3]^{V(G)}$.

We will leverage the following straightforward properties of G_*.

Observation 2.4. Consider G_* as in Definition 2.3. Then, we have the following properties.

1. G_* never contains $K_4^{(3)}$.

2. For any edge $e \in G$, $\mathbb{P}(e \in G_*) = 5/9$.

3. Let $\varphi \in [3]^{V(G)}$ be uniformly randomly chosen. For any edge $e \in G$ and vertex $v \in e$, $\mathbb{P}(e \in \varphi(G) \mid \varphi(v)) = 5/9$.

4. Let $\varphi \in [3]^{V(G)}$ be uniformly randomly chosen. For any edge $e \in G$ and distinct vertices $v_1, v_2 \in e$, we have that

$$\mathbb{P}(e \in \varphi(G) \mid \varphi(v_1) = \varphi(v_2)) = \frac{1}{3}, \quad \mathbb{P}(e \in \varphi(G) \mid \varphi(v_1) \neq \varphi(v_2)) = \frac{2}{3}.$$

Since G_* never contains $K_4^{(3)}$, we may use it in the hypergraph analogue of Proposition 2.2. Proven identically, it is as follows:

Proposition 2.5. *Suppose there exists a finite set $\{H\}$ of unlabeled 3-uniform hypergraphs, $\{c_H\}$ an associated set of coefficients with $c_\varnothing = 2^{\binom{4}{3}} - 1 = 15$, and $\delta > 0$ a constant, such that the function*

$$\mu(G) := (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_* \cong H] \tag{1}$$

satisfies the following conditions:

1. *$|\mu(G)| \leq 1$ whenever $1 \leq e(G) \leq 4$.*

2. *$|\mu(G)| \leq 1 - \delta$ whenever $4 < e(G)$.*

Let $\mathcal{F}$ be a $K_4^{(3)}$-intersecting family of hypergraphs on $[n]$. Then,

- ***(Upper bound)*** *$|\mathcal{F}| \leq 2^{\binom{n}{3}-4}$.*

- ***(Maximal families)*** *The upper bound is an equality if and only if $\mathcal{F}$ is a $K_4^{(3)}$-umvirate.*

In Berger and Zhao's paper [BZ23] on K_4-intersecting families, they constructed a candidate μ by providing a finite collection of $\{c_H, H\}$. They then showed that it sufficed to verify that Constraint 2 of Proposition 2.2 held for all graphs on at most 9 vertices, and then manually checking the constraints for all such small graphs. It is natural to expect small graphs G to be the only ones that contain "binding" constraints; for fixed H, the $\mathbb{P}[G_q \cong H]$ decays to 0 as G grows large. Hence $\mu(G)$ also decays, so for large G it is easier to abstractly verify $|\mu(G)| < 1$ without actually computing $\mu(G)$. There are a limited number of graphs with small $v(G)$, so one might expect it to be easier to just calculate $\mu(G)$ for each individual small G. One can then push these two methods until they meet.

Unfortunately, such a two-pronged approach cannot be directly applied to $K_4^{(3)}$-intersecting families. For $K_4^{(3)}$-intersecting families, $\mathbb{P}[G_* \cong H]$ and hence $\mu(G)$ decay significantly slower. Broadly, this is because for $K_4^{(3)}$-intersecting families, G_* is expected to have 5/9 the edges of G by Observation 2.4, whereas the ratio is 3/4 for K_4-intersecting families, and so the size of G_* grows slower. This implies that G_* is "sparser" relative to G, and so G would need to be larger for it to be unlikely that G_* is the same size as H. Hence, a uniform bound cannot prove $\mu(G) < 1$ unless it only applies to G with a much larger number of vertices. We must then individually compute

$\mu(G)$ for all other G, which is computationally infeasible given the super-exponential growth of the number of hypergraphs on n vertices. This issue is especially bad for hypergraphs (there are over 10^{19} nonisomorphic 3-uniform hypergraphs on just 9 vertices alone)!

However, the general strategy of "individually check smaller G, get a bound for larger G" still holds promise. To facilitate this approach, in the rest of the paper, we will carefully analyze the structure of G_* to achieve tighter nonuniform bounds on $\mu(G)$.

3 A construction for μ

For the remainder of the main body of this article, we study tetrahedron-intersecting families of hypergraphs. In this section, we give a construction of μ for Proposition 2.5 and verify that the conditions of Proposition 2.5 hold for all 3-uniform hypergraphs G on up to 7 vertices. In Section 4, we verify Proposition 2.5 for all G on between 8 and 13 vertices, and in Section 5 we verify Proposition 2.5 for all G on 14 or more vertices.

We generate our choice of c_H in Python using CVXPY [DB16, AVDB18], a convex programming solver, to choose a set of coefficients where $\mu(\varnothing) = c_\varnothing = 15$ and satisfying $|\mu(G)| \leq 1$ for all $G \neq \varnothing$ on up to 7 vertices. Our choices of c_H for Proposition 2.5 are listed in Appendix D as Table 2. We list the subset of c_H with notable magnitude in Table 1.

Set of edges of H	$\boldsymbol{c_H}$
$\varnothing$	15.0
$\{(1,2,3)\}$	-10.2
$\{(1,2,3),(1,2,4)\}$	0.6
$\{(1,2,3),(1,2,4),(1,3,4)\}$	-6.6
$\{(1,2,5),(1,3,4)\}$	3.48
$\{(1,2,4),(1,2,5),(1,3,4),(1,3,5)\}$	-4.249
$\{(1,5,6),(2,3,4)\}$	3.48
$\{(1,2,5),(1,2,6),(1,3,4)\}$	1.051
$\{(1,2,6),(1,3,4),(1,3,5),(1,4,5)\}$	1.28
$\{(1,2,3),(1,2,4),(1,5,6),(2,3,4)\}$	1.395
$\{(1,4,6),(1,5,6),(2,3,4),(2,3,5)\}$	-1.4
$\{(1,2,3),(2,3,4),(5,6,7)\}$	4

Table 1: Values of notable c_H

It turns out that the coefficients c_H for which $H \subseteq K_4^{(3)}$ are uniquely determined when $c_\varnothing = 2^{\binom{4}{3}} - 1$; see Appendix B for a proof. The proof in Appendix B also easily generalizes to K_t-

intersecting families and other $K_t^{(3)}$-intersecting families for any integer $t \geq 3$. Heuristically, we have less freedom over the choice of c_H for small H and more freedom for larger H.

In our choice of coefficients c_H, the hypergraphs H with non-zero c_H are relatively small in size. They have at most 6 edges, and furthermore, no hypergraph with 5 or 6 edges has a coefficient of notable magnitude. Additionally, all but one hypergraph have at most 6 vertices. The one other hypergraph has 7 vertices, and is the last entry in Table 1. Keeping the H small is beneficial as it allows $\mathbb{P}[G_* \cong H]$ and hence $\mu(G)$ to decay as quickly as possible with respect to the size of G.

We will prove Proposition 2.5 for this choice of c_H via a case analysis on $v(G)$. For each value of $v(G)$, the list of all 3-uniform hypergraphs on $v(G)$ vertices is generated using SageMath [The25], which in turn invokes Brendan McKay's Nauty [MP14].

For a given choice of $\{c_H, H\}$, it may not be clear at first glance how one would verify $|\mu(G)| < 1$ for all hypergraphs without an infinitely large computation. The key intuition that underpins our argumentation is that one should only have to individually compute $\mu(G)$ for small G, as we should be able to bound $|\mu(G)|$ away from 1 for large G. To demonstrate this, we give a simple bound that establishes $|\mu(G)| < 1$ for very very large G:

Lemma 3.1 (Terrible bound)**.** *With c_H chosen as in Table 2, and $\mu(G)$ as defined in Eq. (1), we have that $|\mu(G)| < 1$ for all hypergraphs G with at least 250 vertices.*

Proof. We straightforwardly upper bound $|\mu(G)|$, noticing that it's very unlikely that $v(G_*) \leq 7$ if $v(G)$ is large, which is a necessary condition for $G_* \cong H$ for any H with nonzero coefficient c_H. More precisely, we have the following

$$\begin{aligned}
|\mu(G)| &= \left| \sum_H c_H \cdot \mathbb{P}[G_* \cong H] \right| && \text{(Eq. (1))} \\
&\leq \left(\max_H |c_H| \right) \cdot \sum_{c_H \neq 0} \mathbb{P}[G_* \cong H] \\
&\leq 15 \cdot \mathbb{P}[v(G_*) \leq 7] && (c_H \neq 0 \implies v(H) \leq 7) \\
&\leq 15 \cdot \binom{v(G)}{7} \cdot \max_{\substack{S \subset V(G) \\ |S| = v(G)-7}} \mathbb{P}[v \notin G_* \ \forall\ v \in S]
\end{aligned}$$

Let $S_E := \{e \in E(G) \mid \exists v \in S : v \in e\}$ be the set of edges in G containing some vertex in S. Arbitrarily label the edges in S_E as $e_1, \ldots, e_{|S_E|}$. Then,

$$\begin{aligned}
|\mu(G)| &\leq 15 \cdot \binom{v(G)}{7} \cdot \max_{\substack{S \subset V(G) \\ |S| = v(G)-7}} \mathbb{P}[e \notin G_* \ \forall\ e \in S_E] \\
&= 15 \cdot \binom{v(G)}{7} \cdot \max_{\substack{S \subset V(G) \\ |S| = v(G)-7}} \left(\prod_{i=1}^{|S_E|} \mathbb{P}[e_i \notin G_* \mid e_1, \ldots, e_{i-1} \notin G_*] \right)
\end{aligned}$$

If there exists a vertex $v \in e_i$ such that $v \notin e_1, \ldots, e_{i-1}$, then the probability $\mathbb{P}[e_i \notin G_* \mid e_1, \ldots, e_{i-1} \notin G_*]$ is at most $2/3$ by (4) in Observation 2.4.

$$|\mu(G)| \leq 15 \cdot \binom{v(G)}{7} \cdot \max_{\substack{S \subset V(G) \\ |S| = v(G)-7}} \left(\prod_{\substack{1 \leq i \leq |S_E| \\ \exists v \in e_i : v \notin e_1, \ldots, e_{i-1}}} \mathbb{P}[e_i \notin G_* \mid e_1, \ldots, e_{i-1} \notin G_*] \right)$$

$$
\begin{aligned}
&\le 15 \cdot \binom{v(G)}{7} \cdot \max_{\substack{S \subset V(G) \\ |S| = v(G)-7}} \left(\prod_{\substack{1 \le i \le |S_E| \\ \exists v \in e_i : v \notin e_1, \ldots, e_{i-1}}} \frac{2}{3} \right) && \text{(Observation 2.4)} \\
&\le 15 \cdot \binom{v(G)}{7} \cdot \max_{\substack{S \subset V(G) \\ |S| = v(G)-7}} \left(\frac{2}{3}\right)^{|S|/3} && (|e_i| = 3,\ S \subseteq e_1 \cup \cdots \cup e_{|S_E|}) \\
&= 15 \cdot \binom{v(G)}{7} \cdot \left(\frac{2}{3}\right)^{(v(G)-7)/3} \\
&\le 15 \cdot \binom{250}{7} \cdot \left(\frac{2}{3}\right)^{(250-7)/3} && (v(G) \ge 250) \\
&\le 0.999.
\end{aligned}
$$

□

Of course, individually checking hypergraphs on up to 249 vertices is far beyond the limits of computational feasibility. In practice, we can only individually check hypergraphs on up to 7 vertices.

3.1 Verifying 3-uniform hypergraphs G with $v(G) \le 7$

All verification computations are scripted in C++ and are on GitHub at `https://github.com/yunowe/tetrahedronintersectingfamilies`. 3-uniform hypergraphs are encoded via their edge-set indicator vector, which is implemented using the std::bitset data structure. The intersection of 3-uniform hypergraphs then corresponds to the binary AND operation between bitsets.

There are only 7,013,320 3-uniform hypergraphs on up to 7 vertices, which is well within the range of computational feasibility. Note that

$$\mu(G) = (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_* \cong H] = (-1)^{e(G)} \mathop{\mathbb{E}}_{\varphi \in [3]^{V(G)}} [c_{\varphi(G)}],$$

where $c_{\varphi(G)}$ is the value of c_H for the H isomorphic to $\varphi(G)$ (defined in Definition 2.3), and 0 if no such H exists. Hence, for each G, we compute $\mu(G)$ by averaging $c_{\varphi(G)}$ over all $\varphi \in [3]^{V(G)}$.

While computing $\varphi(G)$ is easy, finding which c_H has H isomorphic to $\varphi(G)$ in order to compute $c_{\varphi(G)}$ is not as simple due to the inefficiencies of checking hypergraph isomorphism. However, since computing $c_{\varphi(G)}$ must be done many times for various φ and G, we may do precomputation to significantly speed up the computation of $c_{\varphi(G)}$. For every H with non-zero c_H, we find all 3-uniform hypergraphs on $v(G)$ vertices isomorphic to H by iterating over all ways to choose the vertices of H from $\{1, \ldots, v(G)\}$. This results in a hash table that allows a computer to look up $c_{\varphi(G)}$ in constant time after computing $\varphi(G)$.

Lemma 3.2. *With c_H chosen as in Table 2, and $\mu(G) = (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_* \cong H]$, we have that*

1. *$|\mu(G)| \le 1$ for G with at most 4 edges and at most 7 vertices, and*
2. *$|\mu(G)| < 1$ for G with more than 4 edges and at most 7 vertices.*

Proof. Properties (1) and (2) are verified by computing $\mu(G)$ for each G on up to 7 vertices using the method described above. This computation was relatively quick, taking 47 seconds to complete on a desktop computer. □

4 Verifying 3-uniform hypergraphs G with $8 \le v(G) \le 13$

There are over 10^{12} 3-uniform hypergraphs on 8 vertices alone, so it is infeasible to individually compute $\mu(G)$ for each G using the coefficients in Section 3 to verify Proposition 2.5. Instead, in this section we work to bound $|\mu(G)| < 1$ for enough 3-uniform hypergraphs G to the point where individually computing $\mu(G)$ for the remaining G is computationally feasible. Then, we compute $\mu(G)$ for the remaining G to verify Proposition 2.5 for all G on between 8 and 13 vertices.

Firstly, note that the sum $|\mu(G)| = |\sum c_H \cdot \mathbb{P}[G_* \cong H]|$ in Proposition 2.5 can be bounded by the maximum of the total magnitudes of the positive and negative terms:

$$|\mu(G)| = \left|\sum_H c_H \cdot \mathbb{P}[G_* \cong H]\right| \le \max\left\{\sum_{\substack{H \\ c_H>0}} c_H \cdot \mathbb{P}[G_* \cong H], \sum_{\substack{H \\ c_H<0}} -c_H \cdot \mathbb{P}[G_* \cong H]\right\}.$$

Next, we define monotonic bounds on the sums in the RHS.

Definition 4.1. Let

$$M^+(G) := \sum_{H'}\left(\left(\max_{\substack{H \supseteq H' \\ c_H>0}} c_H\right)\mathbb{P}[G_* \cong H']\right) \quad \text{and} \quad M^-(G) := \sum_{H'}\left(\left(\max_{\substack{H \supseteq H' \\ c_H<0}} -c_H\right)\mathbb{P}[G_* \cong H']\right),$$

where both sums iterate over all unlabeled 3-uniform hypergraphs H'.

Note that both M^+ and M^- are monotone with respect to taking subgraphs in the sense that if $K \subseteq G$ then $M^+(K) \ge M^+(G)$ and $M^-(K) \ge M^-(G)$.

Lemma 4.2. *Let $M(K) := \max\{M^+(K), M^-(K)\}$. For any $K \subseteq G$, we have $|\mu(G)| \le M(K)$.*

Proof. We have the following inequalities based on the definition of $\mu(G)$ in Eq. (1).

$$\begin{aligned}
|\mu(G)| &\le \max\left\{\sum_{\substack{H \\ c_H>0}} c_H \cdot \mathbb{P}[G_* \cong H], \sum_{\substack{H \\ c_H<0}} -c_H \cdot \mathbb{P}[G_* \cong H]\right\} \\
&\le \max\{M^+(G), M^-(G)\} && \text{(Definition 4.1)} \\
&\le \max\{M^+(K), M^-(K)\} && (K \subseteq G) \\
&= M(K). && \square
\end{aligned}$$

Note that the functions M^+ and M^- have a similar shape to μ, only diverging in the associated set of coefficients. Hence, we can compute values of M in an identical manner to how we computed values of μ in Lemma 3.2.

If we compute $M^+(K)$ and $M^-(K)$ for some K and verify that $M(K) < 1$, this immediately implies $|\mu(G)| < 1$ for all $G \supseteq K$, hence verifying Proposition 2.5 for all such G. Our goal is therefore to choose some *satisfactory set* $\mathcal{S}$ of 3-uniform hypergraphs K that satisfies the following pair of properties:

Definition 4.3. A set $\mathcal{S}$ of 3-uniform hypergraphs is *satisfactory* if the following conditions are true:

- $M(K) < 1$ for all $K \in \mathcal{S}$;

- For *most* hypergraphs G we wish to verify (in this section, $8 \leq v(G) \leq 13$), there exists some $K \in \mathcal{S}$ such that $K \subseteq G$, to the point where directly computing $\mu(G)$ on the remaining G is computationally feasible.

If we had this pair of conditions, then we would immediately show $|\mu(G)| < 1$ for almost all G with between 8 and 13 vertices. Ideally, individually computing $\mu(G)$ for the remaining G would be computationally feasible.

Lemma 4.4. *With c_H chosen as in Table 2, and $\mu(G) = (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_* \cong H]$, then*

1. *$|\mu(G)| \leq 1$ for G with at most 4 edges and between 8 and 13 vertices and*
2. *$|\mu(G)| < 1$ for G with more than 4 edges and between 8 and 13 vertices.*

Proof. All implementations are in C++. We choose the set $\mathcal{S}$ to consist of all 3-uniform hypergraphs K satisfying one of the following:

- $v(K) = 8,\ e(K) = 12$
- $v(K) = 9,\ e(K) = 11$
- $v(K) = 10,\ e(K) = 10$
- $v(K) = 11,\ e(K) = 10$
- $v(K) = 12,\ e(K) = 9$
- $v(K) = 13,\ e(K) = 9$

For each of the approximately 160 million $K \in \mathcal{S}$, we individually calculate $M^+(K)$ and $M^-(K)$ by averaging over all colorings $\varphi \in [3]^{V(K)}$ (the same method as in Lemma 3.2), verifying that $M(K) < 1$. Hence, every G for which there exists a $K \in \mathcal{S}$ such that $K \subseteq G$ satisfies $|\mu(G)| < 1$, thus verifying (1) and (2) for these G.

For the remaining approximately 20 million G, we individually compute $\mu(G)$ and verify conditions (1) and (2), proving the lemma.

Verifying that $M(K) < 1$ for all $K \in \mathcal{S}$ turned out to be the most computationally demanding part of this article. Computing $M^+(K)$ and $M^-(K)$ for a single hypergraph K requires computing about $O(3^{v(K)})$ colorings (in practice, this number is slightly lower as some colorings are functionally identical since they delete the same edges from all hypergraphs).

It took about a week to complete the computation on a desktop computer using a Ryzen 7 5800X CPU. Most of the computational power was used on the hypergraphs with $v(K) = 13$ (which makes sense intuitively given the rate at which $O(3^{v(K)})$ grows). □

5 Verifying 3-uniform hypergraphs G with $v(G) \geq 14$

The base strategy in this section is the same as in Section 4 — choose some set $\mathcal{S}$ of 3-uniform hypergraphs K such that $M(K) < 1$ for all $K \in \mathcal{S}$. The issue with directly applying the method in Section 4 is that computing μ, M^+ or M^- on 3-uniform hypergraphs requires one to iterate through $\sim 3^{v(G)}$ colorings of $V(G)$. As $v(G)$ grows, it becomes computationally infeasible to compute these functions on a large number of graphs.

On the other hand, the largest H for which c_H is non-zero in our construction in Section 3 has only 7 vertices. An isomorphism $G_* \cong H$ for some H with non-zero c_H therefore requires G_* to have at most 7 vertices, which becomes unlikely as $v(G)$ grows (illustrated in Lemma 3.1). In practice, when $v(G) \geq 14$, the probability $\mathbb{P}[G_* \cong H]$ has decayed enough to the point where even crude bounds on $|\mu(G)|$ may be able to verify $|\mu(G)| < 1$ (though not nearly as crude as Lemma 3.1).

Since computing $\mu(G)$ is slow for G of this size, we try to choose the $K \in \mathcal{S}$ to have few edges in order to minimize the number of G for which there does not exist $K \in \mathcal{S}$ satisfying $G \supseteq K$. This motivates the following class of hypergraphs:

Definition 5.1. Define a 3-uniform hypergraph K to be *ℓ-minimal* if $v(K) = \ell$, K has no isolated vertices, and no edges of K can be deleted without isolating a vertex.

In particular, ℓ-minimal 3-uniform hypergraphs are those for which all edges contain at least one vertex of degree 1.

Lemma 5.2. *Let ℓ be an integer with $\ell \not\equiv 1 \bmod 3$. Then, any 3-uniform hypergraph G with $v(G) \geq \ell$ and no isolated vertices contains a ℓ-minimal or $(\ell+1)$-minimal 3-uniform hypergraph.*

Proof. We show that one can delete edges from G to form a ℓ-minimal or $(\ell+1)$-minimal 3-uniform hypergraph with extra isolated vertices.

Iteratively delete edges from G while maintaining that the resulting hypergraph has at least ℓ non-isolated vertices until it is no longer possible to do so. Let G' be the resulting hypergraph when this process terminates, and let β be the number of non-isolated vertices in G'. The termination of the process implies that the deletion of any single further edge in G' would cause G' to have fewer than ℓ non-isolated vertices. Therefore, all remaining edges in G' must contain at least $\beta - \ell + 1$ vertices of degree 1. Note that G' is 3-uniform, hence $\beta - \ell + 1 \leq 3$, so $\beta \leq \ell + 2$.

If $\beta = \ell + 2$, then all remaining edges in G must have all three vertices be degree 1, which is impossible since $3 \nmid \ell+2$. Hence, β is either ℓ or $\ell+1$, so G' is either a ℓ-minimal or a $(\ell+1)$-minimal 3-uniform hypergraph with some extra isolated vertices. Since edges can be deleted from G to form G', this proves the lemma. ☐

Lemma 5.2 implies that all G with $v(G) \geq 14$ contain some 14-minimal or 15-minimal hypergraph. Hence, we may wish for $\mathcal{S}$ to consist primarily of 14-minimal and 15-minimal hypergraphs.

Unfortunately, not all 14-minimal and 15-minimal hypergraphs K satisfy $M^+(K) < 1$ and $M^-(K) < 1$. However, since M^+ and M^- are monotonic, we can iteratively add edges to such K until they do satisfy $M(K) < 1$. Hence, a satisfactory set $\mathcal{S}$ can be generated by the following algorithm:

Algorithm 5.3. *To generate a satisfactory set $\mathcal{S}$ such that for all $K \in \mathcal{S}$, $M(K) < 1$:*

1. *Initialize $\mathcal{U}$ to all 14-minimal and 15-minimal 3-uniform hypergraphs and $\mathcal{S} = \varnothing$.*
2. *While $\mathcal{U}$ is nonempty:*
   (a) *Let K be the first element in $\mathcal{U}$ and let $\mathcal{U} \leftarrow \mathcal{U} \setminus \{K\}$.*
   (b) *Compute $M(K)$. If $M(K) < 1$, $\mathcal{S} \leftarrow \mathcal{S} \cup \{K\}$.*
   (c) *Else, Let $\mathcal{K}'$ be the set of all 3-uniform hypergraphs of the form $K \cup \{1 \text{ more edge}\}$. Let $\mathcal{U} \leftarrow \mathcal{U} \cup \mathcal{K}'$.*

Since all sufficiently large hypergraphs K satisfy $M(K) < 1$, Algorithm 5.3 terminates and thus theoretically achieves a finite satisfactory set. In practice it is still computationally infeasible due to the large size of $\mathcal{S}$. However, it turns out that almost all 14-minimal and 15-minimal K do satisfy $M(K) < 1$. We thus give a method that can verify $M(K) < 1$ for numerous ℓ-minimal K

at the same time, speeding up step 2(b) in Algorithm 5.3 and allowing this algorithm to become computationally feasible.

This final algorithm Algorithm 5.4 is an optimization of Algorithm 5.3 that we actually will implement to certify that $|\mu(G)| < 1$ for all G on 14 or more vertices:

Algorithm 5.4. *To generate a satisfactory set $\mathcal{S}$ satisfying $M(K) < 1$ for all $K \in \mathcal{S}$:*

(1) Initialize $\mathcal{S} = \varnothing$.

(2) Generate all equivalence classes $[F, e_2, e_3]$ on 14-minimal and 15-minimal 3-uniform hypergraphs (see Definition 5.9). For each equivalence class $[F, e_2, e_3]$:

- *(a) Calculate a joint probability mass function $\mathbb{P}(V'_K, E_K, C_K)$ for the entire equivalence class using Algorithm 5.13 (the random variables V'_K, E_K, C_K are defined later).*
- *(b) Using the values of $\mathbb{P}(V'_K, E_K, C_K)$, try to certify $M(K) < 1$ for all K in the equivalence class using Lemma 5.8. If this succeeds, let $\mathcal{S} \leftarrow \mathcal{S} \cup [F, e_2, e_3]$ and continue to the next equivalence class. If this fails, proceed to (c).*
- *(c) Initialize $\mathcal{U} = [F, e_2, e_3]$.*
- *(d) While $\mathcal{U}$ is nonempty:*
  - *i. Let K be the first element in $\mathcal{U}$ and let $\mathcal{U} \leftarrow \mathcal{U} \backslash \{K\}$.*
  - *ii. Compute $M(K)$. If $M(K) < 1$, $\mathcal{S} \leftarrow \mathcal{S} \cup \{K\}$.*
  - *iii. Else, Let $\mathcal{K}'$ be the set of all 3-uniform hypergraphs of the form $K \cup \{1 \text{ more edge}\}$. Let $\mathcal{U} \leftarrow \mathcal{U} \cup \mathcal{K}'$.*

We need to define many of the pieces introduced in Algorithm 5.4 and verify correctness.

Lemma 5.5. *Algorithm 5.4 generates a satisfactory set.*

Proof. Any $K \in \mathcal{S}$ is only added after verifying that $M(K) < 1$, so the first condition in Definition 4.3 is verified. By Lemma 5.2, all hypergraphs on 14 or more vertices contain a 14-minimal or 15-minimal hypergraph. Hence, the only hypergraphs G with $v(G) \geq 14$ and such that there does not exist $K \in \mathcal{S}$ with $K \subseteq G$ are precisely the K in (2.d.iii), which is few enough so that individually computing $\mu(G)$ is feasible. $\square$

Definition 5.6. Let K be a 3-uniform hypergraph. Define V_K, E_K, and C_K to be random variables corresponding to the number of vertices, number of edges, and maximum codegree of the random hypergraph $\varphi(K)$, where $\varphi \in [3]^{V(K)}$ is uniformly randomly chosen and $\varphi(K)$ is as in Definition 2.3.

Suppose we knew the joint probability mass function $\mathbb{P}(V_K, E_K, C_K)$ for some 3-uniform hypergraph K. Then, $M^+(K)$ and $M^-(K)$ could be upper bounded as follows.

Lemma 5.7. *Let K be a 3-uniform hypergraph. Then,*

$$M^+(K) = \sum_{H'} \max_{\substack{H \supseteq H' \\ c_H > 0}} c_H \cdot \mathbb{P}[K_* \cong H'] \leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H')=i \\ e(H')=j \\ \Delta_2(H')=k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H > 0}} -c_H \right\} \cdot \mathbb{P}[V_K = i, E_K = j, C_K = k],$$

and similarly

$$M^-(K) = \sum_{H'} \max_{\substack{H \supseteq H' \\ c_H < 0}} |c_H| \cdot \mathbb{P}[K_* \cong H'] \leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H')=i \\ e(H')=j \\ \Delta_2(H')=k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H < 0}} |c_H| \right\} \cdot \mathbb{P}[V_K = i, E_K = j, C_K = k],$$

where $\Delta_2(H')$ denotes the maximum codegree of any pair of vertices in H'.

Proof. We prove the bound for $M^+(K)$; the bound for $M^-(K)$ follows identically.

We prove these bounds by noting that if $K_* \cong H'$, then necessarily

$$v(K_*) = v(H'),\ e(K_*) = e(H'),\ \Delta_2(K_*) = \Delta_2(H').$$

Hence, we have that

$$\begin{aligned}
M^+(G) &= \sum_{H'} \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \cdot \mathbb{P}[K_* \cong H'] \\
&= \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \sum_{\substack{H' \\ v(H')=i \\ e(H')=j \\ \Delta_2(H')=k}} \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \cdot \mathbb{P}[K_* \cong H'] \\
&\leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H')=i \\ e(H')=j \\ \Delta_2(H')=k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \right\} \sum_{\substack{H' \\ v(H')=i \\ e(H')=j \\ \Delta_2(H')=k}} \mathbb{P}[K_* \cong H'] \\
&\leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H')=i \\ e(H')=j \\ \Delta_2(H')=k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \right\} \mathbb{P}\left[v(K_*) = i, e(K_*) = j, \Delta_2(K_*) = k\right] \\
&= \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H')=i \\ e(H')=j \\ \Delta_2(H')=k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \right\} \mathbb{P}[V_K = i, E_K = j, C_K = k],
\end{aligned}$$

as desired, proving the lemma. $\square$

If we could compute $\mathbb{P}(V_K, E_K, C_K)$ for multiple K simultaneously, then applying Lemma 5.7 would allow us to upper bound $M(K)$ for all such K, potentially verifying step (2) in Algorithm 5.3 for many K at once.

It turns out that for a certain equivalence class of ℓ-minimal K (defined later in Definition 5.9), we are almost able to compute $\mathbb{P}(V_K, E_K, C_K)$ for all K in the equivalence class. More precisely, we are able to compute a joint probability mass function $\mathbb{P}(V'_K, E_K, C_K)$, where V'_K is some integer random variable satisfying

$$\mathbb{P}[V_K \geq c \mid E_K, C_K] \geq \mathbb{P}[V'_K \geq c \mid E_K, C_K]$$

for all values of c and realizations of E_K and C_K (i.e. V_K is either equal to V'_K or statewise dominant over V'_K). Heuristically, the values of the V'_K we are able to compute are very similar to the values of V_K, and so it is still highly useful in bounding $M(K)$. We can upper bound $M(K)$ using V'_K instead of V_K by tweaking Lemma 5.7:

Lemma 5.8. *Let K be a 3-uniform hypergraph. Let V'_K be some integer random variable satisfying $\mathbb{P}[V_K \geq c \mid E_K, C_K] \geq \mathbb{P}[V'_K \geq c \mid E_K, C_K]$ for all values of c and realizations of E_K and C_K. Then,*

$$M^+(K) = \sum_{H'} \max_{\substack{H \supseteq H' \\ c_H > 0}} c_H \cdot \mathbb{P}[K_* \cong H'] \leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H') \geq i \\ e(H') = j \\ \Delta_2(H') = k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \right\} \cdot \mathbb{P}[V'_K = i, E_K = j, C_K = k],$$

and similarly

$$M^-(K) = \sum_{H'} \max_{\substack{H \supseteq H' \\ c_H < 0}} -c_H \cdot \mathbb{P}[K_* \cong H'] \leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H') \geq i \\ e(H') = j \\ \Delta_2(H') = k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H < 0}} |c_H| \right\} \cdot \mathbb{P}[V'_K = i, E_K = j, C_K = k].$$

In particular, in Lemma 5.8 $v(H') = i$ is changed to $v(H') \geq i$ and $(V_K = i)$ is changed to $(V'_K = i)$.

Proof. For clarity, we highlight the changes in each step. By Lemma 5.7,

$$\begin{aligned} M^+(K) &\leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H') = i \\ e(H') = j \\ \Delta_2(H') = k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \right\} \cdot \mathbb{P}[V_K = i, E_K = j, C_K = k] \\ &\leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \max_{\substack{H' \\ v(H') \geq i \\ e(H') = j \\ \Delta_2(H') = k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \right\} \cdot \mathbb{P}[V_K = i, E_K = j, C_K = k] \\ &\leq \sum_{\substack{i \geq 0 \\ j \geq 0 \\ k \geq 0}} \underline{\max_{\substack{H' \\ v(H') \geq i \\ e(H') = j \\ \Delta_2(H') = k}} \left\{ \max_{\substack{H \supseteq H' \\ c_H > 0}} |c_H| \right\}} \cdot \mathbb{P}[V'_K = i, E_K = j, C_K = k], \end{aligned}$$

where the last inequality is because the underlined term is non-increasing as i increases. The proof for $M^-(K)$ follows identically. $\square$

We now present the equivalence classes of ℓ-minimal K and how to compute $\mathbb{P}(V'_K, E_K, C_K)$.

Definition 5.9. For a ℓ-minimal 3-uniform hypergraph K, define its equivalence class $[F, e_2, e_3]$ via the following process:

1. Initialize e_2 and e_3 to 0.

2. While there exists an edge in K that contains exactly two vertices of degree 1, delete the edge and increment e_2 by 1. Then, delete any vertices that are now isolated.

3. While there exists an edge in K that contains three vertices of degree 1, delete the edge and increment e_3 by 1. Then, delete any vertices that are now isolated.

4. Set F to the resulting hypergraph.

Two hypergraphs are equivalent if their values of e_2 and e_3 are the same, and their F are isomorphic to each other.

Additionally, arbitrarily fix some ordering of edge deletion as described in steps (2)-(3). Define a 2-leaf edge of K to be an edge deleted in step (2), and a 3-leaf edge of K to be an edge deleted in step (3).

Note that a 3-leaf edge is a connected component of size 3 at the time of its deletion, but not necessarily a connected component of size 3 in the original hypergraph.

Lemma 5.10. *Let K be an ℓ-minimal 3-uniform hypergraph, and let $e = (v_1, v_2, v_3) \in K$ be a 2-leaf or 3-leaf edge. Then, v_1, v_2, and v_3 are each in distinct connected components in $K \backslash \{e\}$.*

Proof. We show any distinct $u, v \in \{v_1, v_2, v_3\}$ are disconnected in $K \backslash \{e\}$. Let K' be the hypergraph obtained at the moment right before e is deleted when executing the process described in Definition 5.9 on K. Since e is a 2-leaf or 3-leaf edge, at least two of v_1, v_2, v_3 have degree 1 in K'. Hence, we may assume without loss of generality that $\deg_{K'}(v) = 1$. Consider the process described in Definition 5.9 in reverse — starting from K', add back the deleted 2-leaf and 3-leaf edges until we return to K. Each 2-leaf and 3-leaf edge that is added back must also add back at least 2 vertices, and hence cannot connect any two already existing vertices. Note that v has degree 1 in K' hence all paths from v to u in K' go through e. Any returned edges in the reverse process from K' to K cannot connect any two vertices in K', thus all paths from v to u in K still go through e. Thus, v and u are disconnected in $K \backslash \{e\}$. □

Recall that by Observation 2.4, for every edge $e \in K$, the probability that $e \in \varphi(K)$ is 5/9 for a uniformly random coloring $\varphi \in [3]^{V(K)}$. For a 2-leaf or 3-leaf edge $e = (v_1, v_2, v_3) \in K$, we have that v_1, v_2, and v_3 are each in distinct connected components when e is deleted by Lemma 5.10. Hence, the realization of whether $e' \in \varphi(K)$ for all $e' \neq e$ has no effect on the joint distribution of colors $(\varphi(v_1), \varphi(v_2), \varphi(v_3))$, which remains uniform. So, the probability that $e \in \varphi(K)$ is independent from whether or not $e' \in \varphi(K)$ for all other edges e'.

Given their independence, the distribution of 2-leaf and 3-leaf edges of K that remain in K_* is easy to compute. Hence, we focus on computing $\mathbb{P}(V_F, E_F, C_F)$, as F is the hypergraph consisting of all non 2-leaf or 3-leaf edges in K.

It unfortunately turns out that simply iterating all $\sim 3^{v(F)}$ colorings of F in order to compute $\mathbb{P}(V_F, E_F, C_F)$ is still computationally infeasible. However, F has structure which can be exploited. By definition, F is an ℓ-minimal 3-uniform hypergraph with no 2-leaf or 3-leaf edges, hence all edges in F contain exactly one vertex of degree one. Let F_1, F_2 partition $V(F)$ into the sets of vertices with degree 1 and degree at least 2, respectively. Then, every edge $e \in F$ must contain exactly 1 vertex in F_1 and 2 vertices in F_2.

Suppose we fix some coloring $\varphi_2 \in [3]^{F_2}$ of the vertices in F_2 and let $\varphi \in [3]^{V(F)}$ be a uniformly random extension of φ_2 to F. Then by Observation 2.4, the probability an edge $e = (u, v_1, v_2) \in F$ with $u \in F_1$ and $v_1, v_2 \in F_2$ remains in $\varphi(F)$ is $1/3$ if $\varphi_2(v_1) = \varphi_2(v_2)$ and $2/3$ otherwise. Furthermore, when φ_2 is fixed, whether or not $e \in \varphi(F)$ depends only on the color $\varphi(u)$. Since each vertex in F_1 corresponds to exactly one edge, the probability that $e \in \varphi(F)$ is independent of any other edge in F when the coloring φ_2 of F_2 is fixed. Using this independence, we can compute the joint distribution of $e(\varphi(F))$ and $\Delta_2(\varphi(F))$. Then, averaging these distributions over all $\varphi_2 \in [3]^{F_2}$ would result in the joint probability mass function $\mathbb{P}(E_F, C_F)$.

Suppose $\varphi_2 \in [3]^{F_2}$ is a coloring of F_2. In the following algorithm, we write $\mid \varphi_2$ to denote "given that F_2 is colored as in φ_2". For example, the probability mass function $\mathbb{P}(E_F, C_F \mid \varphi_2)$ represents the joint distribution of $e(\varphi(F))$ and $\Delta_2(\varphi(F))$, given that φ is a uniformly random extension of φ_2 to F.

Algorithm 5.11. *To compute the joint probability mass function (p.m.f.) $\mathbb{P}(E_F, C_F)$ of F:*

1. *Repeat steps $(2)-(3)$ over all $\varphi_2 \in [3]^{F_2}$, which calculate the p.m.f. $\mathbb{P}(E_F, C_F \mid \varphi_2)$.*
2. *For every adjacent $v_1, v_2 \in F_2$, compute the p.m.f. $\mathbb{P}\left(codeg_{\varphi(F)}(v_1, v_2) \mid \varphi_2\right)$. This distribution is binomial, and it can be read off from the coefficients of the generating function $(2/3+x/3)^{codeg_F(v_1,v_2)}$ if $\varphi_2(v_1) = \varphi_2(v_2)$, or from those of $(1/3+2x/3)^{codeg_F(v_1,v_2)}$ otherwise.*
3. *Using the values of $\mathbb{P}\left(codeg_{\varphi(F)}(v_1, v_2) \mid \varphi_2\right)$ for all adjacent $v_1, v_2 \in F$ that we computed in the previous step, compute the p.m.f. $\mathbb{P}(E_F, C_F \mid \varphi_2)$ as follows:*
$$\mathbb{P}(E_F, C_F \mid \varphi_2) = \mathbb{P}\left(\sum_{\{v_1,v_2\} \subseteq F_2} codeg_{\varphi(F)}(v_1, v_2), \quad \max_{v_1,v_2 \in F_2} codeg_{\varphi(F)}(v_1, v_2) \;\middle|\; \varphi_2 \right).$$
4. *Average the p.m.f.s $\mathbb{P}(E_F, C_F \mid \varphi_2)$ over all $\varphi_2 \in [3]^{F_2}$ to compute $\mathbb{P}(E_F, C_F)$:*
$$\mathbb{P}(E_F, C_F) = \frac{1}{3^{|F_2|}} \cdot \sum_{\varphi_2 \in [3]^{F_2}} \mathbb{P}(E_F, C_F \mid \varphi_2).$$

We now provide some additional explanation on the correctness of some of the steps in Algorithm 5.11.

- Step 2: The codegree $\mathrm{codeg}_{\varphi(F)}(v_1, v_2)$ considers only the edges that contain both v_1 and v_2 in F. Since $v_1, v_2 \in F_2$, the third vertex of any edge e containing both v_1 and v_2, which we will call v, is in F_1. Since $v \in F_1$ thus e is the only edge containing v, hence the probability that $e \in \varphi(F)$ is independent from any other edge when the coloring of v_1 and v_2 are fixed. By Observation 2.4 this probability is $1/3$ if the coloring of v_1 and v_2 is equal and $2/3$ otherwise. This follows for all edges containing both v_1 and v_2, hence the distribution of $\mathbb{P}\left(\mathrm{codeg}_{\varphi(F)}(v_1, v_2) \mid \varphi_2\right)$ is binomial, and it can be read off of the corresponding generating function.
- Step 3: The number of edges $e(\varphi(F))$ is equal to $\sum \mathrm{codeg}_{\varphi(F)}(v_1, v_2)$. The maximum codegree $\Delta_2(\varphi(F))$ is equal to $\max(\mathrm{codeg}_{\varphi(F)}(v_1, v_2))$. The equation for $\mathbb{P}(E_F, C_F \mid \varphi_2)$ follows.
- Step 4: Since $\varphi_2 : F_2 \to \{1, 2, 3\}$, there are $3^{|F_2|}$ possibilities for φ_2, and each of them are equally likely. The equation for $\mathbb{P}(E_F, C_F)$ follows.

Computing $\mathbb{P}(V_F, E_F, C_F)$ is harder. While the distribution of the number of vertices in F_1 that remain in $\varphi(F)$ is easy to compute (since each vertex in F_1 corresponds to exactly one edge, hence this has the same distribution as E_F), computing the distribution for the vertices in F_2 is not as easy.

Consider some coloring $\varphi \in [3]^{V(F)}$, and suppose we were told whether or not $\operatorname{codeg}_{\varphi(F)}(v_1, v_2) = 0$ for every pair of vertices $v_1, v_2 \in F_2$ which are adjacent in F. Then, we could count the number of vertices $v_1 \in F_2$ that remain in $\varphi(F)$: it is precisely the $v_1 \in F_2$ for which there exist some $v_2 \in F_2$ such that $\operatorname{codeg}_{\varphi(F)}(v_1, v_2) > 0$. Hence, our strategy to compute $\mathbb{P}(V_F, E_F, C_F)$ is to casework on whether or not $\operatorname{codeg}_{\varphi(F)}(v_1, v_2) = 0$ for each pair of vertices $v_1, v_2 \in F_2$ which are adjacent in F — for each case, we can precisely count the number of $v \in F_2$ that remain in $\varphi(F)$.

In the following algorithm, let Γ be the set of all unordered pairs of vertices $\{v_1, v_2\} \subseteq F_2$ which are adjacent in F. Let a subset $Q \subseteq \Gamma$ represent the case for which all $\{v_1, v_2\} \in Q$ satisfy $\operatorname{codeg}_{\varphi(F)}(v_1, v_2) > 0$, and all $\{v_1, v_2\} \notin Q$ satisfy $\operatorname{codeg}_{\varphi(F)}(v_1, v_2) = 0$. We write $\mid Q$ to denote "given that case Q is true". For example, the p.m.f. $\mathbb{P}\left(\operatorname{codeg}_{\varphi(F)}(v_1, v_2) \mid \varphi_2, Q\right)$ represents the distribution of $\operatorname{codeg}_{\varphi(F)}(v_1, v_2)$, given that φ is a uniformly random extension of φ_2 to F that satisfies case Q.

Algorithm 5.12. *To compute the joint probability mass function $\mathbb{P}(V_F, E_F, C_F)$ of F:*

1. *Repeat steps* $(2) - (7)$ *over all $\varphi_2 \in [3]^{F_2}$, which calculate the p.m.f. $\mathbb{P}(V_F, E_F, C_F \mid \varphi_2)$*

2. *Repeat steps* $(3) - (6)$ *over all $Q \subseteq \Gamma$.*

3. *Count the number of $v \in F_2$ that are contained in some pair in Q. For the case Q, these are the vertices in F_2 that remain in $\varphi(F)$. Store this count as α_Q.*

4. *Compute the probability that case Q occurs for a uniformly random extension $\varphi \in [3]^{V(F)}$ of φ_2 to F, which we store as β_Q. Given their independence, this is achieved by taking the product of the probability for each individual condition of Q:*

$$\beta_Q = \prod_{\{v_1,v_2\}\in\Gamma} \begin{cases} 1 - (2/3)^{codeg_F(v_1,v_2)}, & \text{if } \varphi(v_1) = \varphi(v_2) \text{ and } \{v_1, v_2\} \in Q \\ 1 - (1/3)^{codeg_F(v_1,v_2)}, & \text{if } \varphi(v_1) \neq \varphi(v_2) \text{ and } \{v_1, v_2\} \in Q \\ (2/3)^{codeg_F(v_1,v_2)}, & \text{if } \varphi(v_1) = \varphi(v_2) \text{ and } \{v_1, v_2\} \notin Q \\ (1/3)^{codeg_F(v_1,v_2)}, & \text{if } \varphi(v_1) \neq \varphi(v_2) \text{ and } \{v_1, v_2\} \notin Q \end{cases}.$$

5. *For each $\{v_1, v_2\} \in Q$, compute the p.m.f. $\mathbb{P}\left(codeg_{\varphi(F)}(v_1, v_2) \mid \varphi_2, Q\right)$. This distribution is a zero-truncated binomial distribution. If $\varphi_2(v_1) = \varphi_2(v_2)$, it can be computed via the coefficients of the generating function $(2/3 + x/3)^{codeg_F(v_1,v_2)}$, discarding the constant term then normalizing the coefficients to sum to 1. Otherwise, if $\varphi_2(v_1) \neq \varphi_2(v_2)$, the generating function $(1/3 + 2x/3)^{codeg_F(v_1,v_2)}$ is used instead.*

6. *Compute the p.m.f. $\mathbb{P}(V_F, E_F, C_F \mid \varphi_2, Q)$ using the values computed in the previous step:*

$$\begin{aligned}\mathbb{P}(V_F, E_F, C_F \mid \varphi_2, Q) = \mathbb{P}\Bigg(&\alpha_Q + \sum_{\{v_1,v_2\}\in Q} codeg_{\varphi(F)}(v_1, v_2), \\ &\sum_{\{v_1,v_2\}\in Q} codeg_{\varphi(F)}(v_1, v_2), \\ &\max_{\{v_1,v_2\}\in Q} codeg_{\varphi(F)}(v_1, v_2) \;\Bigg|\; \varphi_2, Q\Bigg).\end{aligned}$$

7. *Compute a weighted sum of* $\mathbb{P}(V_F, E_F, C_F \mid \varphi_2, Q)$ *with weight* β_Q *over all cases* $Q \subseteq \Gamma$ *to obtain* $\mathbb{P}(V_F, E_F, C_F \mid \varphi_2)$*:*

$$\mathbb{P}(V_F, E_F, C_F \mid \varphi_2) = \sum_{Q \subseteq \Gamma} \beta_Q \cdot \mathbb{P}(V_F, E_F, C_F \mid \varphi_2, Q).$$

8. *Average* $\mathbb{P}(V_F, E_F, C_F \mid \varphi_2)$ *over all* $\varphi_2 \in [3]^{F_2}$ *to obtain* $\mathbb{P}(V_F, E_F, C_F)$*:*

$$\mathbb{P}(V_F, E_F, C_F) = \frac{1}{3^{|F_2|}} \cdot \sum_{\varphi_2 \in [3]^{F_2}} \mathbb{P}(V_F, E_F, C_F \mid \varphi_2).$$

Note that the description of Algorithm 5.12 provided above is rather high-level. For additional discussion on the actual code implementation of Algorithm 5.12, please refer to Appendix C.

We now provide some additional explanation on the correctness of some of the steps in Algorithm 5.12.

- Step 4: As discussed previously, the probability that $e \in \varphi(F)$ is independent of the realization of any other edge in $\varphi(F)$ when the coloring φ_2 of F_2 is fixed. The probability of each individual condition of Q is therefore independent from all other conditions. If the condition is that $\mathrm{codeg}_{\varphi(F)}(v_1, v_2) = 0$, then it happens with probability $(1-p)^{\mathrm{codeg}_F(v_1,v_2)}$, where p is the probability that some edge containing v_1 and v_2 remains in $\varphi(F)$ (which is $1/3$ if v_1 and v_2 are identically colored and $2/3$ otherwise by Observation 2.4). Otherwise, if the condition is that $\mathrm{codeg}_{\varphi(F)}(v_1, v_2) > 0$, the probability is $1 - (1-p)^{\mathrm{codeg}_F(v_1,v_2)}$. The equation for β_Q follows.

- Step 5: If $\{v_1, v_2\} \in Q$, then case Q gives us that $\mathrm{codeg}_{\varphi(F)}(v_1, v_2) \neq 0$, and so the distribution of $\mathrm{codeg}_{\varphi(F)}(v_1, v_2)$ is a zero-truncated binomial distribution. See the explanation given with Algorithm 5.11 for further discussion.

- Step 6: The number of vertices $v(\varphi(F))$ is equal to the number of vertices that remain in $\varphi(F)$ from F_1 plus those that remain from F_2. We counted the number of vertices that remain in F_2 in Step 3 and stored it as α_Q. Since $\deg_F(v) = 1$ for all $v \in F_1$, the number of vertices that remain from F_1 is equal to the number of edges in $\varphi(F)$, which is $\sum \mathrm{codeg}_{\varphi(F)}(v_1, v_2)$. For E_F and C_F, see the explanation given with Algorithm 5.11. The equation for $\mathbb{P}(V_F, E_F, C_F \mid \varphi_2, Q)$ follows.

- Step 7: Given φ_2, each case Q occurs with probability β_Q, and so we take the weighted average with weight β_Q for each case.

- Step 8: See the explanation given with Algorithm 5.11.

Once we have computed the p.m.f. $\mathbb{P}(V_F, E_F, C_F)$ for F, all that remains is to consider the remaining 2-leaf and 3-leaf edges of K.

Every 3-leaf edge that remains in $\varphi(K)$ adds 3 to the vertex count $v(\varphi(K))$, 1 to the edge count $e(\varphi(K))$, and sets the maximum codegree $\Delta_2(\varphi(K))$ to 1 if it was previously 0. Similarly, every 2-leaf edge that remains in $\varphi(K)$ adds *at least* 2 to the vertex count $v(\varphi(K))$, exactly 1 to the edge count $e(\varphi(K))$, and sets the maximum codegree $\Delta_2(\varphi(K))$ to 1 if it was previously 0. In particular, a 2-leaf edge remaining in $\varphi(K)$ could add 3 instead of 2 to $v(\varphi(K))$, which is why we cannot compute the exact p.m.f. $\mathbb{P}(V_K, E_K, C_K)$ using this method and instead compute $\mathbb{P}(V'_K, E_K, C_K)$.

Algorithm 5.13. *To compute a joint probability mass function* $\mathbb{P}(V_K', E_K, C_K)$ *for all* K *in an equivalence class* $[F, e_2, e_3]$*:*

1. *Using Algorithm 5.12, compute* $\mathbb{P}(V_F, E_F, C_F)$*.*
2. *Let* $X_1, \ldots, X_{e_2}$ *and* $Y_1, \ldots, Y_{e_3}$ *be independent random variables which are* 1 *with probability* $5/9$ *and* 0 *with probability* $4/9$*.* X_i *represents the event that the ith 2-leaf edge remains in* K_**, and similarly* Y_i *represents the event that the ith 3-leaf edge remains in* K_**. Then, a satisfactory p.m.f.* $\mathbb{P}(V_K', E_K, C_K)$ *is*

$$\mathbb{P}(V_K', E_K, C_K) =$$
$$\mathbb{P}\Big(V_F + 2\sum_{i=1}^{e_2} X_i + 3\sum_{i=1}^{e_3} Y_i,\ E_F + \sum_{i=1}^{e_2} X_i + \sum_{i=1}^{e_3} Y_i,\ \max\{C_F, X_1, \ldots, X_{e_2}, Y_1, \ldots, Y_{e_2}\}\Big).$$

Using Algorithm 5.13, we are now able to optimize Algorithm 5.3 to be computationally feasible.

Lemma 5.14. *With* c_H *chosen as in Table 2, and* $\mu(G) = (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_* \cong H]$*, then* $|\mu(G)| < 1$ *for all* G *on at least* 14 *vertices.*

Proof. We execute Algorithm 5.4 (see Appendix C for details) to generate a satisfactory set $\mathcal{S}$. By Lemma 5.2, the only G on 14 or more vertices for which there does not exist $K \in \mathcal{S}$ satisfying $K \subseteq G$ are precisely the K deleted in step (6) of Algorithm 5.4. We individually compute $\mu(G)$ for each such G and verify that $|\mu(G)| < 1$. Otherwise, for K, G with $K \subseteq G$ and $K \in \mathcal{S}$, we have

$$\mu(G) \le M(K) < 1,$$

hence proving the lemma. □

We are now ready to verify our choice of c_H in Table 2 against the conditions in Proposition 2.5:

Lemma 5.15. *With* c_H *chosen as in Table 2, and* $\mu(G) = (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_* \cong H]$*, then there exists* $\delta > 0$ *such that*

1. $|\mu(G)| \le 1$ *whenever* $1 \le e(G) \le 4$
2. $|\mu(G)| \le 1 - \delta$ *whenever* $4 < e(G)$*.*

Proof. Together, Lemma 3.2, Lemma 4.4, and Lemma 5.14 verify Condition 1 and state that $|\mu(G)| < 1$ for all G with $e(G) > 4$. The only remaining issue is to check that such a $\delta > 0$ exists (i.e. there are no issues with an infinite sequence $(G_i)_{i\ge 0}$ with $\lim |\mu(G_i)| = 1$).

We proceed by cases:

- *Case 1:* $v(G) \le 7$. Condition 1 is verified by Lemma 3.2. Additionally, Lemma 3.2 states that all G in this case with $v(G) > 4$ satisfy $|\mu(G)| < 1$. Since this case is finite, such a δ exists, verifying Condition 2.
- *Case 2:* $8 \le v(G) \le 13$. Condition 1 is verified by Lemma 4.4. Additionally, Lemma 4.4 states that all G in this case with $v(G) > 4$ satisfy $|\mu(G)| < 1$. Since this case is finite, such a δ exists, verifying Condition 2.
- *Case 3:* $14 \le v(G)$. Condition 1 does not apply, since a hypergraph with 4 edges has at most $4 \cdot 3 = 12$ vertices. Lemma 5.14 states that all G in this case satisfy $|\mu(G)| < 1$. We split into two sub cases:

– *Case 3.1.* $|\mu(G)| < 1$ was verified in Lemma 5.14 as there existed some $K \in \mathcal{S}$ with $K \subseteq G$. Then,
$$|\mu(G)| \leq \max_{K \in \mathcal{S}} M(K) < 1.$$
Since $\mathcal{S}$ is finite, there exists δ with $\max_{K \in \mathcal{S}} M(K) \leq 1 - \delta$. Hence, there exists δ with $|\mu(G)| \leq 1 - \delta$ for all G in this case, verifying Condition 2.

– *Case 3.2.* $|\mu(G)| < 1$ was verified in Lemma 5.14 via an individual computation of $\mu(G)$. Since this case is finite, such a δ exists, verifying Condition 2.

$\square$

Our choice of coefficients in Table 2 has $c_{\varnothing} = 2^{\binom{4}{3}} - 1 = 15$ and satisfies the constraints of Proposition 2.5. Hence, we have proved the main result, Theorem 1.7.

Acknowledgements

Thank you to the MIT PRIMES program for making this research opportunity possible. NM was supported by the NSF Graduate Research Fellowship and the Hertz Graduate Fellowship. Thank you to Aaron Berger, Shu Ge, and Yufei Zhao for helpful discussions.

References

[AVDB18] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[BZ23] Aaron Berger and Yufei Zhao. K4-intersecting families of graphs. *Journal of Combinatorial Theory, Series B*, 163:112–132, 2023.

[CGFS86] F.R.K Chung, R.L Graham, P Frankl, and J.B Shearer. Some intersection theorems for ordered sets and graphs. *Journal of Combinatorial Theory, Series A*, 43(1):23–37, 1986.

[DB16] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[DEF78] M. Deza, P. Erdös, and P. Frankl. Intersection properties of systems of finite sets. *Proceedings of the London Mathematical Society*, s3-36(2):369–384, 1978.

[EFF12] David Ellis, Yuval Filmus, and Ehud Friedgut. Triangle-intersecting families of graphs, 2012.

[EKR61] Paul Erdős, Chao Ko, and Richard Rado. Intersection theorems for systems of finite sets. *Quart. J. Math. Oxford Ser. (2)*, 12:313–320, 1961.

[Ell22] David Ellis. Intersection problems in extremal combinatorics: theorems, techniques and questions old and new. *Surveys in Combinatorics*, pages 115–173, 2022.

[FR87] Peter Frankl and Vojtech Rodl. Forbidden intersections. *Transactions of the American Mathematical Society*, 300(1):259–286, 1987.

[Fri08] Ehud Friedgut. On the measure of intersecting families, uniqueness and stability. *Combinatorica*, 28(5):503–528, 2008.

[FW81] P. Frankl and R. M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1(4):357–368, 1981.

[FW86] P. Frankl and R.M. Wilson. The erdös-ko-rado theorem for vector spaces. *Journal of Combinatorial Theory, Series A*, 43(2):228–236, 1986.

[JT08] J. Robert Johnson and John M. Talbot. G-intersection theorems for matchings and other graphs. *Combinatorics, Probability and Computing*, 17:559 – 575, 2008.

[Kat64] G. O. H. Katona. Intersection theorems for systems of finite sets. *Acta Mathematica Academiae Scientiarum Hungaricae*, 15:329–337, 1964.

[MP14] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94–112, 2014.

[The25] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 10.6)*, 2025. `https://www.sagemath.org`.

[Wil84] Richard M. Wilson. The exact bound in the erdös-ko-rado theorem. *Combinatorica*, 4(2):247–257, 1984.

A The maximum size of a K_5-intersecting family

In this subsection, we prove Theorem 1.8. All verification computations are scripted in C++, and are available online at `https://github.com/yunowe/tetrahedronintersectingfamilies`.

Berger and Zhao [BZ23] introduced a method to uniformly bound $|\mu(G)| < 1$ for graphs with sufficiently many vertices, which they applied to prove Conjecture 1.4 for $t = 3, 4$. In particular, their method is theoretically sufficient to prove Conjecture 1.4 for any fixed t, but in practice, it becomes computationally infeasible for all $t \geq 5$. We use their method with some optimizations to prove Conjecture 1.4 for $t = 5$, and we highly recommend the reader see [BZ23] for a more detailed explanation of the graph setting. Identically to Section 2, the proof of Conjecture 1.4 can be reduced to the following linear program:

Proposition A.1 (Proposition 2.5 in [BZ23])**.** *There exists a set of unlabeled graphs $\{H\}$, coefficients $\{c_H\}$ and $\delta > 0$ so that for any G on n labeled vertices, we have that*

$$\mu(G) := (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_4 \cong H]$$

satisfies the following conditions.

1. $\mu(\varnothing) = 2^{\binom{5}{2}} - 1 = 1023$.
2. $|\mu(G)| \leq 1$ *for all* $G \neq \varnothing$.
3. $|\mu(G)| \leq 1 - \delta$ *whenever* G *has more than* 10 *edges.*

Recall that as per Definition 2.1, G_4 is defined as follows:

Definition A.2. For a graph G on n labeled vertices, $[4]^{V(G)}$ is the set of maps $\varphi\colon V(G) \to \{0, 1, 2, 3\}$, viewed as 4-colorings of $V(G)$ (not necessarily proper). For each coloring $\varphi : V(G) \to \{0, 1, 2, 3\}$, $\varphi(G)$ is the subgraph of G formed by deleting all monochromatic edges of G, and then deleting all isolated vertices from the result. Then, G_4 is the random graph $\varphi(G)$ given by choosing $\varphi \sim \mathrm{Unif}([4]^{V(G)})$.

We first give our construction for Proposition A.1. To save space, Berger and Zhao chose the coefficients c_H to be equal on a certain equivalence class on H, defined as follows:

Definition A.3 (Definition 4.2 in [BZ23])**.** A *block* of a graph H is a maximal connected subgraph with at least one edge and no cut vertex, which is a vertex whose removal increases the number of connected components of H. The collection of blocks of H partitions $E(H)$. We say two graphs H and H' are equivalent and write $H \sim H'$ if the collection of blocks of H and the collection of blocks of H' are equal as multisets of unlabeled graphs.

Even then, the list of c_H is still very long, found in Table 3. To verify the conditions of Proposition A.1, Berger and Zhao gave the following uniform bound on $\mu(G)$ for all G with sufficiently many vertices:

Proposition A.4 (Proposition 3.5 in [BZ23])**.** *Fix $q > 0$, a list $\{H\}$ of unlabeled graphs on at most n_0 vertices and $\{c_H\}$ a list of coefficients. Then for any G on $n > n_0$ labeled vertices, we have*

$$\sum_H |c_H| \cdot \mathbb{P}[G_4 \cong H] \leq \max_{G' \subseteq K_{n_0}} \left[\frac{1}{q^{k(G')-1}} \sum_H \widetilde{c_H} \cdot \mathbb{P}[G_4' \cong H] \cdot \max_{\substack{\ell \in \mathbb{Z} \\ \ell > n_0}} \frac{\binom{\ell}{n_0}}{q^{\ell - n_0} \binom{\ell - v(H)}{n_0 - v(H)}} \right],$$

where $\widetilde{c_H}$ is the maximum value of $|c_{H'}|$ over all graphs H' which may be transformed to H by repeatedly identifying pairs of disconnected vertices, and $\kappa(G)$ is the number of connected components of G.

We employ Proposition A.4 with a few modifications to prove Proposition A.1. In particular, we upper bound $|\mu(G)| = |\sum_H c_H \cdot \mathbb{P}[G_4 \cong H]|$ instead of $\sum_H |c_H| \cdot \mathbb{P}[G_4 \cong H]$.

Firstly, recall that $\sum c_H \cdot \mathbb{P}[G_4 \cong H]$ can be bounded via its positive and negative terms:

$$|\mu(G)| = \left|\sum_H c_H \cdot \mathbb{P}[G_4 \cong H]\right| \leq \max\left\{\sum_{\substack{H \\ c_H > 0}} c_H \cdot \mathbb{P}[G_4 \cong H], \sum_{\substack{H \\ c_H < 0}} -c_H \cdot \mathbb{P}[G_4 \cong H]\right\}.$$

In particular, both of the sums on the right hand side are of the original form $\sum |c_H| \cdot \mathbb{P}[G_4 \cong H]$, and applying Proposition A.4 to each sum separately yields a tighter bound.

Next, in the proof of Proposition A.4 in [BZ23], Berger and Zhao weaken the constraint $G' \in \binom{G}{n_0}$ to $G' \subseteq K_{n_0}$ when taking the maximum, where $\binom{G}{n_0}$ is the set of labeled subgraphs of G induced by all choices of n_0 vertices from $V(G)$. We strengthen this constraint by additionally requiring that $e(G') \leq e(G)$, which is possible because $G' \in \binom{G}{n_0}$ implies $G' \subseteq G$ and hence $e(G') \leq e(G)$. Note that the resulting bound is now no longer uniform as it now depends on $e(G)$.

Finally, restricting Proposition A.4 to only apply to G with $v(G) > n_1$ for some fixed $n_1 \geq n_0$ allows us to update the $\ell > n_0$ constraint in Proposition A.4 to $\ell > n_1$. These changes yield the following modified version of Proposition A.4:

Proposition A.5. *Fix $q > 0$, a list $\{H\}$ of unlabeled graphs on at most n_0 vertices, $\{c_H\}$ a list of coefficients, and an integer $n_1 \geq n_0$. Then for any G on $n > n_1$ labeled vertices, we have*

$$\left|\sum_H c_H \cdot \mathbb{P}[G_4 \cong H]\right| \leq \max\left\{ \max_{\substack{G' \subseteq K_{n_0} \\ e(G') \leq e(G)}} \left[\frac{1}{q^{k(G')-1}} \sum_H \widetilde{c_{pos_H}} \cdot \mathbb{P}[G_4' \cong H] \cdot \max_{\substack{\ell \in \mathbb{Z} \\ \ell > n_0}} \frac{\binom{\ell}{n_1}}{q^{\ell - n_0}\binom{\ell - v(H)}{n_0 - v(H)}}\right], \right.$$
$$\left. \max_{\substack{G' \subseteq K_{n_0} \\ e(G') \leq e(G)}} \left[\frac{1}{q^{k(G')-1}} \sum_H -\widetilde{c_{neg_H}} \cdot \mathbb{P}[G_4' \cong H] \cdot \max_{\substack{\ell \in \mathbb{Z} \\ \ell > n_0}} \frac{\binom{\ell}{n_1}}{q^{\ell - n_0}\binom{\ell - v(H)}{n_0 - v(H)}}\right] \right\},$$

where $\widetilde{c_{pos_H}}$ is the maximum positive value of $c_{H'}$ over all graphs H' which may be transformed to H by repeatedly identifying pairs of disconnected vertices, $\widetilde{c_{neg_H}}$ is the minimum negative value of $c_{H'}$ over all graphs H' which may be transformed to H by repeatedly identifying pairs of disconnected vertices, and $\kappa(G)$ is the number of connected components of G.

We are now ready to prove the validity of our choice of c_H in Table 3.

Lemma A.6. *With c_H chosen as in Table 3, and $\mu(G) = (-1)^{e(G)} \sum_H c_H \cdot \mathbb{P}[G_4 \cong H]$, one has*

1. *$\mu(\varnothing) = 1023$,*
2. *$|\mu(G)| \leq 1$ whenever G has at most 10 edges, and*
3. *$|\mu(G)| \leq 0.998$ whenever G has more than 10 edges.*

Proof. If $G = \varnothing$ then G_4 is also always the empty graph, hence $\mu(\varnothing) = c_\varnothing = 1023$, verifying (1). Properties (2) and (3) are verified for all 12005168 graphs on up to 10 vertices by individually computing $\mu(G)$ for each such graph G.

For G on 11 or more vertices, we employ Proposition A.5 with $q = 4$, $n_0 = 9$, and $n_1 = 10$. However, there is a slight issue — a singular H in our construction in Table 3 has non-zero c_H and $10 \not\leq 9$ vertices. This graph consists of 5 disjoint edges and has coefficient 0.22. Call this graph $[5e]$. To resolve this issue, we bound $\sum_H c_H \cdot \mathbb{P}[G_4 \cong H]$ by giving separate bounds for $\sum_{H \not\cong [5e]} c_H \cdot \mathbb{P}[G_4 \cong H]$ and $c_{[5e]} \cdot \mathbb{P}[G_4 \cong [5e]]$, the former of which can be done via Proposition A.5. We proceed by cases.

- *Case 1:* $v(G) \geq 11, e(G) \leq 10$. In this case, we compute Proposition A.5 with the constraint $e(G') \leq 10$ (since $e(G') \leq e(G) \leq 10$). Iterating over all graphs on up to 9 vertices gives that

$$\left| \sum_{H \not\cong [5e]} c_H \cdot \mathbb{P}[G_4 \cong H] \right| \leq 0.322.$$

Hence,

$$\begin{aligned} \left| \sum_H c_H \cdot \mathbb{P}[G_4 \cong H] \right| &\leq \left| \sum_{H \not\cong [5e]} c_H \cdot \mathbb{P}[G_4 \cong H] \right| + \left| c_{[5e]} \cdot \mathbb{P}[G_4 \cong [5e]] \right| \\ &\leq 0.322 + 0.22 \cdot 1 \\ &\leq 0.999, \end{aligned}$$

verifying conditions (2) and (3) for this case.

- *Case 2:* $v(G) \geq 11, e(G) \geq 11$. In this case, we compute Proposition A.5 with the constraint $e(G') \leq e(G)$ omitted (since $e(G)$ is not bounded). Iterating over all graphs on up to 9 vertices gives that

$$\left| \sum_{H \not\cong [5e]} c_H \cdot \mathbb{P}[G_4 \cong H] \right| \leq 0.897.$$

Note that every edge in any graph G has a $\frac{3}{4}$ chance of remaining in G_4, hence $\mathbb{E}[e(G_4)] = \frac{3}{4} e(G)$. Since $G_4 \cong [5e]$ implies $e(G_4) = 5$, hence we can write an upper bound on $\mathbb{E}[e(G_4)]$ in terms of $\mathbb{P}[G_4 \cong [5e]]$:

$$\begin{aligned} 5\mathbb{P}[G_4 \cong [5e]] + e(G)(1 - \mathbb{P}[G_4 \cong [5e]]) &\geq \mathbb{E}[e(G_4)] \\ &= \frac{3}{4} e(G). \end{aligned}$$

Solving for $\mathbb{P}[G_4 \cong [5e]]$ gives

$$\mathbb{P}[G_4 \cong [5e]] \leq \frac{e(G)}{4e(G) - 20}.$$

In this case $e(G) \geq 11$, so $\mathbb{P}[G_4 \cong [5e]] \leq \frac{11}{24}$. Now,

$$\left| \sum_H c_H \cdot \mathbb{P}[G_4 \cong H] \right| \leq \left| \sum_{H \not\cong [5e]} c_H \cdot \mathbb{P}[G_4 \cong H] \right| + \left| c_{[5e]} \cdot \mathbb{P}[G_4 \cong [5e]] \right|$$

$$
\begin{aligned}
&\leq 0.897 + 0.22 \cdot \frac{11}{24} \\
&\leq 0.999,
\end{aligned}
$$

verifying conditions (2) and (3) for this case.

$\square$

This verifies our construction in Table 3, proving Proposition A.1 and hence Theorem 1.8.

B Coefficients c_H are uniquely determined for $H \subseteq K_4^{(3)}$

We give this argument in the setting of $K_4^{(3)}$-intersecting families of 3-uniform hypergraphs, but it follows essentially verbatim to show that for any K_t-intersecting family of graphs, c_H is uniquely determined for any $H \subseteq K_t$.

Lemma B.1. *For any 3-uniform hypergraph $R \supseteq K_4^{(3)}$, we have that*

$$\sum_{G \subseteq R} \mu(G) = 0.$$

Proof. By the definition of μ in Proposition 2.5, we wish to show that

$$\begin{aligned}
\sum_{G \subseteq R} (-1)^{e(G)} \sum_{H} c_H \mathbb{P}[G_* \cong H] &= 0 \\
\sum_{H} c_H \sum_{G \subseteq R} (-1)^{e(G)} \mathbb{P}[G_* \cong H] &= 0 \\
\sum_{H} c_H \sum_{H' \cong H} \sum_{G \subseteq R} (-1)^{e(G)} \mathbb{P}[G_* = H'] &= 0.
\end{aligned}$$

It suffices to show the innermost summation is always 0 for a fixed subgraph H'. Expanding $\mathbb{P}[G_* = H']$ gives

$$\begin{aligned}
\sum_{G \subseteq R} (-1)^{e(G)} \frac{|\{\varphi \in [q]^{V(R)} : \varphi(G) = H'\}|}{|\{\varphi \in [q]^{V(R)}\}|} &= 0 \\
\sum_{G \subseteq R} (-1)^{e(G)} \sum_{\substack{\varphi \in [q]^{V(R)} \\ \varphi(G) = H'}} 1 &= 0 \\
\sum_{\varphi \in [q]^{V(R)}} \sum_{\substack{G \subseteq R \\ \varphi(G) = H'}} (-1)^{e(G)} &= 0.
\end{aligned}$$

It suffices to show the inner summation is always 0. Since $R \supseteq K_4^{(3)}$, there exists an edge $e \in R \backslash \varphi(R)$. If $e \in H'$, then no G satisfies the conditions of the summation. Otherwise, note each G contributes either 1 or -1 to the sum based on the parity of $e(G)$. We can pair up graphs of opposite parity by pairing G with $G \triangle \{e\}$, where $\triangle$ denotes symmetric difference, hence the inner summation is always 0, proving the lemma. $\square$

Corollary B.2. *If $\mu(\varnothing) = 2^{\binom{4}{3}} - 1$, then for all $G \subseteq K_4^{(3)}$ with $G \neq \varnothing$, we have that $\mu(G) = -1$.*

Proof. Follows immediately from Lemma B.1 with $R = K_4^{(3)}$ since $\mu(\varnothing) = 2^{\binom{4}{3}} - 1$ and $|\mu(G)| \leq 1$ for all $G \neq \varnothing$. $\square$

Lemma B.3. *For all $H \subset K_4^{(3)}$, the value of c_H is uniquely determined.*

Proof. Let $H_1, \ldots, H_\ell$ be some ordering of the subgraphs $H \subset K_4^{(3)}$ with the property that if $H_i \subset H_j$ then $i < j$. Let $\mathbf{c} = (c_{H_1}, \ldots, c_{H_\ell})$ be the coefficient vector, and let

$$\mathbf{d} := (\mu(H_1), \ldots, \mu(H_\ell))$$

$$= (2^{\binom{4}{3}} - 1, -1, \ldots, -1). \qquad \text{(Corollary B.2)}$$

Note that $\mu(H)$ is a linear combination of c_H and $c_{H'}$ for some set of $H' \subset H$, in other words, that $\mu(H_i)$ is a linear combination of c_{H_j} with $j \leq i$. In particular, the coefficient of c_H in $\mu(H)$ is non-zero for all $H \subset K_4^{(3)}$. It follows that $\mathbf{d} = A\mathbf{c}$ for some lower triangular matrix A with no zeroes on the diagonal. Hence A is invertible, and so $\mathbf{c}$ is uniquely determined. □

C Additional discussion of implementation

In this section, we discuss the steps needed to translate Algorithm 5.12 into efficient C++ code, which can be found as part of `vecbound.cpp` on GitHub at `https://github.com/yunowe/tetrahedronintersectingfamilies`.

To be exact, `vecbound.cpp` implements Steps 2(a) and 2(b) in Algorithm 5.4, which invokes Algorithm 5.12. Few enough equivalence classes fail to be bounded by Step 2(b) such that Steps 2(c) and 2(d) can be done essentially manually.

In `vecbound.cpp`, we represent probability mass functions as fixed-size arrays where each entry holds the probability of a certain realization. Note that the only use of the p.m.f.s $\mathbb{P}(V_F, E_F, C_F)$ in the final algorithm Algorithm 5.4 is to eventually be used in Lemma 5.8 to bound $M(K)$. So, if some realization of V_F, E_F, C_F does not contribute to the sum, then we need not consider it when computing Algorithm 5.12. In particular, since for non-zero c_H in our choice of coefficients we have that $\max v(H) = 7$, $\max e(H) = 6$, and $\max \Delta_2(H) = 4$, we only need to consider realizations with $V_F \leq 7$, $E_F \leq 6$, and $C_F \leq 4$. So, our arrays representing p.m.f.s have size $(7{+}1)(6{+}1)(4{+}1) = 280$, which is small enough that Algorithm 5.12 and Algorithm 5.4 can run efficiently even given the number of p.m.f.s involved in its computation.

Additionally, in `vecbound.cpp`, we exploit the structure of F when representing equivalence classes $[F, e_2, e_3]$ to obtain a more efficient representation. Recall that for any equivalence class $[F, e_2, e_3]$, the vertices of the 3-uniform hypergraph F can be partitioned into two sets, F_1 and F_2, where each vertex in F_1 has degree exactly 1 and each edge contains exactly 1 vertex in F_1. Each vertex in F_1 therefore corresponds to exactly one edge in F and vice versa. In that sense, the structure of F can be determined by only looking at the vertices in F_2 and their pairwise connections.

For some F, consider a weighted graph F' on the vertex set F_2, where vertices $u, v \in F'$ are connected by an edge of weight $\operatorname{codeg}_F(u, v)$ for all $u, v \in F_2$ with $\operatorname{codeg}_F(u, v) \neq 0$. Given the structure of all possible F discussed in the previous paragraph, the resulting weighted graphs F' form a straightforward bijection with the F. Explicitly, the original hypergraph F can be reconstructed from F' (which has vertex set F_2) as follows: For each edge (u, w) in F' of weight $k = \operatorname{weight}_{F'}(u, w)$, add k new distinct vertices $x_1, \ldots, x_k$ to F_1, and add the k edges $\{x_1, u, w\}, \ldots, \{x_k, u, w\}$ to $E(F)$.

Such a representation is highly efficient. Each vertex in F_2 has degree at least 2, and each edge has exactly 2 vertices in F_2, hence $|F_2| \leq e(F)$. Each edge corresponds to a vertex in F_1, so $|F_2| \leq |F_1|$ thus $|F_2| \leq \lfloor v(F)/2 \rfloor$. For the equivalence classes of 14-minimal and 15-minimal hypergraphs, this implies that F' has no more than 7 vertices.

7 is a small enough number where the list of equivalence classes $[F, e_2, e_3]$ of 14-minimal and 15-minimal hypergraphs can be generated by simply iterating over all weighted graphs F' on 7 vertices and checking for validity (if the corresponding hypergraph F satisfies the structural constraint of $\deg_F(v) \geq 2$ for all $v \in F_2$), then generating all possibilities for e_2 and e_3 such that

$$\begin{aligned}
v(K) \text{ for some } K \in [F, e_2, e_3] &= v(F) + 2e_2 + 3e_3 \\
&= |F_2| + |F_1| + 2e_2 + 3e_3 \\
&= v(F') + \sum_{v_1, v_2 \in F'} \operatorname{weight}(v_1, v_2) + 2e_2 + 3e_3 \\
&= 14 \text{ or } 15.
\end{aligned}$$

D Certificates

Set of edges of H	c_H
$\varnothing$	15.0
$\{(1,2,3)\}$	-10.2
$\{(1,2,3),(1,2,4)\}$	0.6
$\{(1,2,3),(1,2,4),(1,3,4)\}$	-6.6
$\{(1,2,3),(2,3,4),(5,6,7)\}$	4.0
$\{(1,2,5),(1,3,4)\}$	3.48
$\{(1,2,3),(1,2,4),(1,2,5)\}$	-0.2
$\{(1,2,3),(1,2,5),(1,3,4)\}$	-0.329
$\{(1,2,5),(1,3,4),(2,3,4)\}$	-0.109
$\{(1,2,3),(1,2,4),(1,2,5),(1,3,4)\}$	0.099
$\{(1,2,4),(1,2,5),(1,3,4),(1,3,5)\}$	-4.249
$\{(1,2,4),(1,2,5),(1,3,5),(2,3,4)\}$	-0.13
$\{(1,2,3),(1,2,4),(1,2,5),(1,3,4),(1,3,5)\}$	-0.181
$\{(1,2,4),(1,2,5),(1,3,4),(1,3,5),(2,3,4)\}$	-0.167
$\{(1,2,3),(1,2,4),(1,2,5),(1,3,4),(1,3,5),(1,4,5)\}$	-0.2
$\{(1,2,3),(1,2,4),(1,2,5),(1,3,5),(1,4,5),(2,3,4)\}$	-0.155
$\{(1,2,4),(1,2,5),(1,3,4),(1,3,5),(2,3,4),(2,3,5)\}$	-0.2
$\{(1,2,3),(1,2,5),(1,3,4),(1,4,5),(2,3,4),(2,3,5)\}$	-0.175
$\{(1,5,6),(2,3,4)\}$	3.48
$\{(1,2,5),(1,2,6),(1,3,4)\}$	1.051
$\{(1,2,6),(1,3,5),(2,3,4)\}$	-0.194
$\{(1,2,3),(1,5,6),(2,3,4)\}$	-0.32
$\{(1,2,3),(1,2,4),(1,2,5),(1,2,6)\}$	-0.107
$\{(1,2,3),(1,2,5),(1,2,6),(1,3,4)\}$	-0.142
$\{(1,2,4),(1,2,6),(1,3,4),(1,3,5)\}$	-0.191
$\{(1,2,6),(1,3,4),(1,3,5),(1,4,5)\}$	1.28
$\{(1,2,4),(1,2,6),(1,3,5),(2,3,4)\}$	-0.151
$\{(1,2,3),(1,2,4),(1,5,6),(2,3,4)\}$	1.395
$\{(1,4,6),(1,5,6),(2,3,4),(2,3,5)\}$	-1.4
$\{(1,3,5),(1,4,6),(2,3,6),(2,4,5)\}$	-0.186
$\{(1,2,3),(1,2,4),(1,2,6),(1,3,4),(1,3,5)\}$	-0.13
$\{(1,2,4),(1,2,5),(1,2,6),(1,3,4),(1,3,5)\}$	-0.152
$\{(1,2,3),(1,2,6),(1,3,4),(1,3,5),(1,4,5)\}$	-0.141
$\{(1,2,5),(1,2,6),(1,3,4),(1,3,6),(1,4,5)\}$	-0.163
$\{(1,2,5),(1,2,6),(1,3,4),(1,3,5),(2,3,4)\}$	-0.14
$\{(1,2,3),(1,2,4),(1,2,5),(1,2,6),(1,3,4),(1,3,5)\}$	-0.139
$\{(1,2,3),(1,2,4),(1,2,6),(1,3,4),(1,3,5),(1,4,5)\}$	-0.14
$\{(1,2,3),(1,2,5),(1,2,6),(1,3,4),(1,3,6),(1,4,5)\}$	-0.145
$\{(1,2,4),(1,2,5),(1,2,6),(1,3,4),(1,3,5),(2,3,4)\}$	-0.134
$\{(1,2,4),(1,2,5),(1,2,6),(1,3,5),(1,3,6),(2,3,4)\}$	-0.106
$\{(1,2,4),(1,2,6),(1,3,4),(1,3,5),(2,3,4),(2,3,5)\}$	-0.104

Table 2: All values of c_H for $K_4^{(3)}$-intersecting families.

Blocks of H	c_H
$\emptyset$	1023
[(1, 2)]	-1019/3
[(1, 2)], [(1, 2)]	111
[(1, 2), (1, 3), (2, 3)]	-1003/3
[(1, 2)], [(1, 2)], [(1, 2)]	-899/27
[(1, 2)], [(1, 2), (1, 3), (2, 3)]	919/9
[(1, 3), (1, 4), (3, 2), (4, 2)]	-2939/27
[(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)]	-6067/27
[(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)]	37717/27
[(1, 2)], [(1, 2)], [(1, 2)], [(1, 2)]	133/27
[(1, 2)], [(1, 2)], [(1, 2), (1, 3), (2, 3)]	-499/27
[(1, 2)], [(1, 3), (1, 4), (3, 2), (4, 2)]	8347/189
[(1, 4), (1, 5), (4, 3), (5, 2), (2, 3)]	-5887/135
[(1, 2)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)]	6611/189
[(1, 3), (1, 4), (1, 5), (3, 2), (4, 2), (5, 2)]	-130685/3213
[(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 3), (2, 3)]	583/9
[(1, 2), (1, 4), (1, 5), (2, 3), (2, 5), (4, 3)]	-98209/945
[(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5)]	129211/3213
[(1, 2)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)]	-39989/189
[(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 5), (3, 4)]	-23363/315
[(1, 3), (1, 4), (1, 5), (3, 2), (3, 4), (4, 2), (5, 2)]	3515111/16065
[(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4)]	391871/16065
[(1, 2), (1, 3), (1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)]	-5050369/16065
[(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5)]	1378795/3213
[(1, 2)], [(1, 2)], [(1, 2)], [(1, 2)], [(1, 2)]	0.219058525
[(1, 2)], [(1, 2)], [(1, 3), (1, 4), (3, 2), (4, 2)]	-7.447032798
[(1, 2)], [(1, 4), (1, 5), (4, 3), (5, 2), (2, 3)]	19.829752462
[(1, 5), (1, 6), (5, 3), (6, 2), (2, 4), (4, 3)]	-20.763588549
[(1, 2)], [(1, 2)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)]	-5.144376306
[(1, 2)], [(1, 3), (1, 4), (1, 5), (3, 2), (4, 2), (5, 2)]	5.365789975
[(1, 2)], [(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 3), (2, 3)]	2.610161629
[(1, 2)], [(1, 2), (1, 4), (1, 5), (2, 3), (2, 5), (4, 3)]	15.359394228
[(1, 2), (1, 3), (2, 3)], [(1, 3), (1, 4), (3, 2), (4, 2)]	28.384304031
[(1, 4), (1, 5), (1, 6), (4, 3), (5, 2), (6, 2), (2, 3)]	-31.633111537
[(1, 2), (1, 5), (1, 6), (2, 4), (2, 6), (5, 3), (4, 3)]	-31.718830366
[(1, 4), (1, 5), (1, 6), (4, 2), (4, 3), (5, 3), (6, 2)]	-34.406458558
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5)]	-3.580410482
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (4, 2), (5, 2), (6, 2)]	59.921749604
[(1, 2)], [(1, 2)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)]	17.290633947
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 5), (3, 4)]	10.242928902
[(1, 2)], [(1, 3), (1, 4), (1, 5), (3, 2), (3, 4), (4, 2), (5, 2)]	-19.933019061
[(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)]	2.557057952
[(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 5), (2, 6), (4, 3)]	15.668654541
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (5, 2), (6, 2)]	-4.735028986
[(1, 2), (1, 3), (1, 5), (1, 6), (2, 4), (2, 6), (3, 4), (3, 5)]	-55.171844856
[(1, 2), (1, 4), (1, 5), (1, 6), (2, 4), (2, 6), (4, 3), (5, 3)]	-35.155678907
[(1, 4), (1, 5), (1, 6), (4, 2), (4, 3), (5, 3), (6, 2), (2, 3)]	53.778763919
[(1, 4), (1, 5), (1, 6), (4, 2), (4, 3), (5, 2), (5, 3), (6, 2)]	10.056923024
[(1, 2), (1, 4), (1, 6), (2, 3), (2, 6), (4, 3), (4, 5), (3, 5)]	-53.829709125
[(1, 4), (1, 5), (1, 6), (4, 3), (4, 5), (5, 3), (6, 2), (2, 3)]	98.671470717
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6)]	89.214193111
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4)]	-14.108776829
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 5), (2, 6), (3, 4)]	-37.3551175
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (4, 2), (5, 2), (6, 2)]	-148.554444755
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)]	39.917238773
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 4), (2, 6), (3, 4), (3, 5)]	-35.36528482
[(1, 2), (1, 3), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5)]	6.740707268
[(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (4, 3), (5, 3)]	20.037857983
[(1, 2), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (4, 3), (5, 3)]	-14.477823517
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 4), (3, 5), (4, 2), (5, 2), (6, 2)]	-52.678728465
[(1, 4), (1, 5), (1, 6), (4, 2), (4, 3), (5, 2), (5, 3), (6, 2), (6, 3)]	40.515739774
[(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)]	-8.873652621
[(1, 2), (1, 3), (1, 4), (1, 6), (2, 3), (2, 6), (3, 4), (3, 5), (4, 5)]	-13.004890238
[(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 6), (4, 3), (4, 5), (5, 3)]	125.084656438
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (4, 5), (6, 2)]	78.085797992
[(1, 3), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (5, 4), (6, 2), (2, 4)]	-22.803555394
[(1, 4), (1, 5), (1, 6), (4, 3), (4, 5), (5, 2), (6, 2), (6, 3), (2, 3)]	-75.224016678
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4)]	-32.164689671
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5)]	-56.212896275
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5)]	60.038307571
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5)]	16.570160688
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (4, 2), (5, 2), (6, 2)]	93.376336459
[(1, 2), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (4, 3), (5, 3), (6, 3)]	122.456530082
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 6), (3, 4), (3, 5), (4, 5)]	63.527494686
[(1, 2), (1, 3), (1, 4), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5), (4, 5)]	-51.417654096
[(1, 2), (1, 3), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5), (5, 4)]	-81.561741995
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (4, 2), (4, 5), (6, 2)]	-66.228028289
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 6), (4, 2), (4, 5), (5, 2), (6, 2)]	247.662082551
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 5), (2, 6), (3, 4), (3, 6), (4, 5)]	272.36405273
[(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 5), (2, 6), (4, 3), (4, 5), (6, 3)]	-33.359264533
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5)]	-107.827301827
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6)]	-35.977491427
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5), (4, 5)]	77.509338127
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (4, 2), (4, 5), (5, 2), (6, 2)]	146.612857852
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 6), (4, 5)]	262.751414017
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 5), (2, 6), (3, 4), (3, 6), (4, 5)]	64.400400071
[(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (3, 6), (4, 2), (4, 5), (5, 2), (6, 2)]	-121.982339169
[(1, 2), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (4, 3), (4, 5), (5, 3), (6, 3)]	-276.311937294
[(1, 2)], [(1, 5), (1, 6), (5, 3), (6, 2), (2, 4), (4, 3)]	3.93446663
[(1, 6), (1, 7), (6, 3), (7, 2), (2, 5), (5, 4), (3, 4)]	-9.329910321
[(1, 2)], [(1, 4), (1, 5), (1, 6), (4, 3), (5, 2), (6, 2), (2, 3)]	-1.471705104
[(1, 2)], [(1, 2), (1, 5), (1, 6), (2, 4), (2, 6), (5, 3), (4, 3)]	-0.586429226
[(1, 2)], [(1, 4), (1, 5), (1, 6), (4, 2), (4, 3), (5, 3), (6, 2)]	-0.023437291
[(1, 3), (1, 4), (3, 2), (4, 2)], [(1, 3), (1, 4), (3, 2), (4, 2)]	5.340637353
[(1, 5), (1, 6), (1, 7), (5, 3), (6, 2), (7, 2), (2, 4), (4, 3)]	-2.197629808
[(1, 2), (1, 3), (2, 3)], [(1, 4), (1, 5), (4, 3), (5, 2), (2, 3)]	4.00873273
[(1, 5), (1, 6), (1, 7), (5, 4), (6, 3), (7, 2), (2, 3), (2, 4)]	-12.749077358
[(1, 2), (1, 6), (1, 7), (2, 5), (2, 7), (6, 3), (5, 4), (3, 4)]	-12.186170092
[(1, 4), (1, 6), (1, 7), (4, 3), (4, 5), (6, 3), (7, 2), (2, 5)]	-12.612371159

[(1, 2)], [(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (4, 2), (5, 2), (6, 2)]	-6.959976017
[(1, 2)], [(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 5), (2, 6), (4, 3)]	7.400013147
[(1, 2)], [(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (5, 2), (6, 2)]	2.68064497
[(1, 2), (1, 3), (2, 3)], [(1, 3), (1, 4), (1, 5), (3, 2), (4, 2), (5, 2)]	-8.52829241
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (5, 2), (6, 2), (7, 2), (2, 3)]	26.670911137
[(1, 2)], [(1, 2), (1, 3), (1, 5), (1, 6), (2, 4), (2, 6), (3, 4), (3, 5)]	-3.197168111
[(1, 2)], [(1, 2), (1, 4), (1, 5), (1, 6), (2, 4), (2, 6), (4, 3), (5, 3)]	-2.612795464
[(1, 2)], [(1, 4), (1, 5), (1, 6), (4, 2), (4, 3), (5, 3), (6, 2), (2, 3)]	4.021187155
[(1, 2)], [(1, 4), (1, 5), (1, 6), (4, 2), (4, 3), (5, 2), (5, 3), (6, 2)]	1.925315836
[(1, 3), (1, 4), (3, 2), (4, 2)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)]	-2.158195753
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (5, 3), (6, 2), (7, 2), (2, 3)]	-6.12182107
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (5, 3), (4, 3)]	5.968372145
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 4), (3, 5), (6, 2), (7, 2), (2, 4)]	-3.39393484
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (5, 3), (6, 2), (7, 2)]	2.35575712
[(1, 4), (1, 6), (1, 7), (4, 2), (4, 3), (6, 3), (7, 2), (2, 5), (5, 3)]	22.779351531
[(1, 2)], [(1, 2), (1, 4), (1, 6), (2, 3), (2, 6), (4, 3), (4, 5), (3, 5)]	-3.483370372
[(1, 2)], [(1, 4), (1, 5), (1, 6), (4, 3), (4, 5), (5, 3), (6, 2), (2, 3)]	3.298282196
[(1, 4), (1, 6), (1, 7), (4, 3), (4, 5), (6, 2), (7, 2), (2, 3), (3, 5)]	1.459289617
[(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 4), (1, 5), (2, 3), (2, 5), (4, 3)]	-2.261472107
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 7), (5, 4), (6, 3)]	0.106793866
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 2), (3, 6), (5, 4), (7, 2), (2, 4)]	-6.078454431
[(1, 5), (1, 6), (1, 7), (5, 4), (6, 3), (7, 2), (2, 3), (2, 4), (3, 4)]	16.086034404
[(1, 2), (1, 3), (1, 6), (1, 7), (2, 5), (2, 7), (3, 4), (3, 6), (5, 4)]	-2.210874381
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 5), (2, 7), (4, 3), (4, 5), (6, 3)]	-6.096302177
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 4), (3, 6), (4, 5), (7, 2), (2, 5)]	-3.83126519
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (4, 5), (5, 2), (6, 3), (7, 2)]	-11.788018449
[(1, 4), (1, 6), (1, 7), (4, 3), (4, 5), (6, 3), (7, 2), (2, 3), (2, 5)]	3.767989672
[(1, 3), (1, 6), (1, 7), (3, 4), (3, 6), (7, 2), (2, 4), (2, 5), (4, 5)]	-0.251972589
[(1, 5), (1, 6), (1, 7), (5, 2), (5, 4), (6, 3), (7, 2), (2, 4), (4, 3)]	13.701287343
[(1, 5), (1, 6), (1, 7), (5, 2), (5, 4), (6, 2), (6, 3), (7, 2), (3, 4)]	5.197228796
[(1, 6), (1, 7), (6, 3), (7, 2), (2, 4), (2, 5), (4, 3), (4, 5), (5, 3)]	28.984031972
[(1, 4), (1, 6), (1, 7), (4, 3), (4, 5), (6, 2), (6, 3), (7, 2), (3, 5)]	-6.391508691
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6)]	-3.633359793
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 2), (4, 2), (5, 2), (6, 2), (7, 2)]	-15.860406814
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 5), (2, 6), (3, 4)]	4.400109383
[(1, 2)], [(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (4, 2), (5, 2), (6, 2)]	-3.716243735
[(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5)]	2.489735347
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 5), (2, 6), (2, 7), (4, 3)]	0.488362482
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 2), (3, 4), (5, 2), (6, 2), (7, 2)]	7.409294491
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 4), (2, 6), (3, 4), (3, 5)]	0.845958786
[(1, 2)], [(1, 2), (1, 3), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5)]	6.737126194
[(1, 2)], [(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (4, 3), (5, 3)]	-1.491092303
[(1, 2)], [(1, 2), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (4, 3), (5, 3)]	-2.39594656
[(1, 2)], [(1, 3), (1, 4), (1, 5), (1, 6), (3, 4), (3, 5), (4, 2), (5, 2), (6, 2)]	0.95852221
[(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)]	1.905006244
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 6), (2, 7), (4, 3), (5, 3)]	-26.893942953
[(1, 2), (1, 3), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (3, 4), (3, 5)]	-4.582241274
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (4, 3), (5, 3)]	0.974410173
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 4), (3, 5), (4, 2), (6, 2), (7, 2)]	4.96602303
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 2), (3, 4), (3, 5), (6, 2), (7, 2), (2, 4)]	7.331176003
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (5, 3), (6, 2), (7, 2), (2, 3)]	-15.480185334
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (5, 2), (5, 3), (6, 2), (7, 2)]	-11.217781744
[(1, 2)], [(1, 4), (1, 5), (1, 6), (4, 2), (4, 3), (5, 2), (5, 3), (6, 2), (6, 3)]	-1.749728945
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (4, 3), (6, 3), (5, 3)]	2.602315827
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (5, 2), (5, 3), (6, 3), (7, 2)]	-2.388127803
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 6), (2, 3), (2, 6), (3, 4), (3, 5), (4, 5)]	-4.25665776
[(1, 2)], [(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 6), (4, 3), (4, 5), (5, 3)]	5.031871473
[(1, 2)], [(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (4, 5), (6, 2)]	4.705671194
[(1, 2)], [(1, 3), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (5, 4), (6, 2), (2, 4)]	4.717895187
[(1, 3), (1, 4), (3, 2), (4, 2)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)]	32.960154572
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 3), (2, 6), (2, 7), (4, 3), (4, 5), (3, 5)]	-0.176514365
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 2), (3, 4), (3, 5), (4, 5), (6, 2), (7, 2)]	19.300201204
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (4, 5), (5, 3), (6, 2), (7, 2), (2, 3)]	29.650634288
[(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 5), (3, 4)]	-1.842293521
[(1, 2), (1, 3), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 7), (3, 6), (5, 4)]	-2.567379833
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 2), (3, 6), (4, 2), (4, 5), (7, 2)]	-13.088409408
[(1, 2), (1, 3), (2, 3)], [(1, 3), (1, 4), (1, 5), (3, 2), (3, 4), (4, 2), (5, 2)]	8.368812768
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 2), (3, 6), (4, 2), (4, 5), (7, 2), (2, 5)]	-13.018912682
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 5), (5, 2), (6, 3), (7, 2), (2, 3)]	-40.884490159
[(1, 2)], [(1, 4), (1, 5), (1, 6), (4, 3), (4, 5), (5, 2), (6, 2), (6, 3), (2, 3)]	2.010837096
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 7), (5, 4), (6, 3), (3, 4)]	-1.745533978
[(1, 2), (1, 3), (1, 4), (1, 6), (1, 7), (2, 5), (2, 7), (3, 4), (3, 6), (4, 5)]	2.3704569
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 5), (2, 7), (4, 3), (4, 5), (6, 3)]	-1.655192538
[(1, 2), (1, 3), (1, 6), (1, 7), (2, 3), (2, 5), (2, 7), (3, 4), (3, 6), (5, 4)]	3.61895855
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 3), (2, 5), (2, 7), (4, 3), (4, 5), (6, 3)]	8.460220116
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 2), (3, 4), (3, 6), (4, 5), (7, 2), (2, 5)]	-5.674810557
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (4, 5), (5, 2), (6, 3), (7, 2), (2, 3)]	-6.211243813
[(1, 2), (1, 3), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (3, 4), (3, 6), (4, 5)]	2.908276006
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 4), (3, 6), (4, 2), (4, 5), (7, 2), (2, 5)]	-1.125346319
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (5, 4), (6, 3), (4, 3)]	-4.177890357
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 4), (3, 6), (5, 2), (5, 4), (7, 2), (2, 4)]	4.289880719
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (4, 5), (5, 2), (6, 3), (7, 2)]	-2.043105841
[(1, 5), (1, 6), (1, 7), (5, 2), (5, 4), (6, 3), (7, 2), (2, 3), (2, 4), (3, 4)]	-3.09560784
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 5), (2, 6), (2, 7), (5, 4), (6, 3), (3, 4)]	-6.394789016
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 4), (3, 6), (5, 2), (5, 4), (6, 2), (7, 2)]	-1.26960095
[(1, 2), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (6, 3), (4, 3), (4, 5), (5, 3)]	4.940706436
[(1, 4), (1, 6), (1, 7), (4, 2), (4, 3), (4, 5), (6, 3), (7, 2), (2, 5), (5, 3)]	1.16373339
[(1, 6), (1, 7), (6, 3), (7, 2), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)]	-16.735165807
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 6), (2, 7), (4, 3), (4, 5), (6, 3), (3, 5)]	3.726513348
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 4), (3, 5), (3, 6), (4, 5), (6, 2), (7, 2)]	5.208685185
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (4, 5), (5, 3), (6, 2), (6, 3), (7, 2)]	19.819743359
[(1, 4), (1, 6), (1, 7), (4, 3), (4, 5), (6, 2), (6, 3), (7, 2), (2, 3), (3, 5)]	7.35350091
[(1, 4), (1, 6), (1, 7), (4, 2), (4, 3), (4, 5), (6, 2), (6, 3), (7, 2), (3, 5)]	-8.678003782
[(1, 5), (1, 6), (1, 7), (5, 3), (5, 4), (6, 2), (6, 3), (7, 2), (2, 4), (4, 3)]	-21.105384688
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (5, 4), (6, 3), (7, 3)]	-7.269732853
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 6), (3, 7), (5, 4), (6, 2), (7, 2), (2, 4)]	-9.191699476
[(1, 5), (1, 6), (1, 7), (5, 4), (6, 2), (6, 3), (7, 2), (7, 3), (2, 4), (4, 3)]	21.890379787
[(1, 4), (1, 5), (1, 7), (4, 5), (4, 6), (5, 2), (5, 3), (7, 2), (2, 3), (3, 6)]	13.390446003
[(1, 4), (1, 6), (1, 7), (4, 5), (4, 6), (6, 3), (7, 2), (2, 3), (2, 5), (3, 5)]	-7.937257869
[(1, 2), (1, 6), (1, 7), (2, 5), (2, 7), (6, 3), (6, 4), (5, 3), (5, 4), (3, 4)]	-4.579823142

[(1, 3), (1, 6), (1, 7), (3, 4), (3, 5), (3, 6), (6, 4), (7, 2), (2, 5), (5, 4)]	-20.862225834
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7)]	-5.295275178
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4)]	6.132325262
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 5), (2, 6), (2, 7), (3, 4)]	-4.157370192
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 2), (3, 4), (4, 2), (5, 2), (6, 2), (7, 2)]	34.457660612
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5)]	-3.14553808
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5)]	-1.743579562
[(1, 2)], [(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (4, 2), (5, 2), (6, 2)]	-2.024741766
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 6), (2, 7), (3, 4), (3, 5)]	-34.582824618
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (3, 4), (3, 5)]	8.027226526
[(1, 2), (1, 3), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 6), (2, 7), (3, 4), (3, 5)]	-3.591018543
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 6), (2, 7), (4, 3), (5, 3)]	-6.074593298
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 2), (3, 4), (3, 5), (4, 2), (6, 2), (7, 2)]	7.557057661
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 4), (2, 5), (2, 6), (2, 7), (4, 3), (5, 3)]	-12.278009904
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 4), (3, 5), (4, 2), (5, 2), (6, 2), (7, 2)]	-8.58231085
[(1, 2)], [(1, 2), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (4, 3), (5, 3), (6, 3)]	-2.531033246
[(1, 2), (1, 3), (1, 4), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (3, 4), (3, 5), (3, 6)]	-16.274902414
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (4, 3), (5, 3), (6, 3)]	-0.977418141
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (5, 2), (5, 3), (6, 3), (7, 2), (2, 3)]	-4.954239332
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (5, 2), (5, 3), (6, 2), (6, 3), (7, 2)]	-19.27251958
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 6), (3, 4), (3, 5), (4, 5)]	1.751643266
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5), (4, 5)]	22.871691682
[(1, 2)], [(1, 2), (1, 3), (1, 5), (1, 6), (2, 3), (2, 4), (2, 6), (3, 4), (3, 5), (5, 4)]	-0.884575764
[(1, 2)], [(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 4), (3, 5), (4, 2), (4, 5), (6, 2)]	-5.679683441
[(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)], [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)]	-8.451611594
[(1, 2), (1, 3), (1, 4), (1, 6), (1, 7), (2, 3), (2, 6), (2, 7), (3, 4), (3, 5), (4, 5)]	4.607137164
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 6), (2, 7), (4, 3), (4, 5), (5, 3)]	44.117695509
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 2), (3, 4), (3, 5), (4, 5), (6, 2), (7, 2)]	9.892552551
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 2), (3, 4), (3, 5), (4, 2), (4, 5), (6, 2), (7, 2)]	-14.232665558
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 2), (3, 4), (3, 5), (5, 4), (6, 2), (7, 2), (2, 4)]	-10.802489588
[(1, 2)], [(1, 3), (1, 4), (1, 5), (1, 6), (3, 2), (3, 6), (4, 2), (4, 5), (5, 2), (6, 2)]	6.424094889
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 7), (3, 6), (4, 5)]	-20.107616665
[(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4)]	-6.83563532
[(1, 2), (1, 3), (1, 4), (1, 6), (1, 7), (2, 3), (2, 4), (2, 5), (2, 7), (3, 6), (4, 5)]	-14.264866154
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 5), (2, 7), (4, 5), (6, 3)]	-13.890509715
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 2), (3, 6), (4, 2), (4, 5), (5, 2), (7, 2)]	-14.430264559
[(1, 2)], [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 5), (2, 6), (3, 4), (3, 6), (4, 5)]	21.442856095
[(1, 2)], [(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 5), (2, 6), (4, 3), (4, 5), (6, 3)]	-4.912223036
[(1, 2), (1, 3), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 7), (3, 4), (3, 6), (5, 4)]	6.826577085
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 5), (2, 7), (3, 4), (3, 6), (4, 5)]	-0.902727802
[(1, 2), (1, 3), (1, 4), (1, 6), (1, 7), (2, 3), (2, 5), (2, 7), (3, 4), (3, 6), (4, 5)]	5.076803039
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 5), (2, 7), (4, 3), (4, 5), (6, 3)]	16.02794721
[(1, 2), (1, 3), (1, 4), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (3, 4), (3, 6), (4, 5)]	2.891259083
[(1, 2), (1, 3), (1, 5), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (3, 4), (3, 6), (5, 4)]	2.177864908
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (4, 3), (4, 5), (6, 3)]	-5.216693345
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 4), (3, 6), (4, 2), (4, 5), (5, 2), (7, 2)]	-2.279776061
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 2), (3, 4), (3, 6), (4, 2), (4, 5), (7, 2), (2, 5)]	13.277213773
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 5), (2, 7), (5, 4), (6, 3), (3, 4)]	-17.365150373
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 2), (3, 4), (3, 6), (5, 2), (5, 4), (7, 2), (2, 4)]	-6.609918649
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (4, 5), (5, 2), (6, 3), (7, 2), (2, 3)]	7.313575499
[(1, 2), (1, 3), (1, 5), (1, 6), (1, 7), (2, 5), (2, 6), (2, 7), (3, 4), (3, 6), (5, 4)]	0.854411722
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 4), (3, 6), (4, 5), (5, 2), (6, 2), (7, 2)]	9.222247371
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 2), (3, 4), (3, 6), (5, 2), (5, 4), (6, 2), (7, 2)]	-4.664441246
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (4, 5), (5, 2), (6, 2), (6, 3), (7, 2), (2, 3)]	0.642441935
[(1, 2), (1, 3), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (3, 4), (3, 5), (3, 6), (4, 5)]	-26.301659702
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 4), (2, 5), (2, 7), (4, 3), (4, 5), (6, 3), (5, 3)]	-4.069212049
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (4, 5), (5, 2), (5, 3), (6, 3), (7, 2)]	23.254129322
[(1, 2), (1, 6), (1, 7), (2, 3), (2, 4), (2, 5), (2, 7), (6, 3), (3, 4), (3, 5), (4, 5)]	4.653489719
[(1, 4), (1, 6), (1, 7), (4, 2), (4, 3), (4, 5), (6, 3), (7, 2), (2, 3), (2, 5), (3, 5)]	-9.251707121
[(1, 2), (1, 3), (1, 4), (1, 6), (1, 7), (2, 6), (2, 7), (3, 4), (3, 5), (3, 6), (4, 5)]	-1.757846699
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 6), (2, 7), (4, 3), (4, 5), (5, 3), (6, 3)]	-7.169334467
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 4), (3, 5), (3, 6), (4, 5), (6, 2), (7, 2)]	-13.342591648
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 3), (2, 6), (2, 7), (4, 3), (4, 5), (6, 3), (3, 5)]	-1.721213112
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 2), (3, 4), (3, 5), (3, 6), (4, 5), (6, 2), (7, 2)]	11.317734312
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (4, 5), (5, 3), (6, 2), (6, 3), (7, 2), (2, 3)]	32.218161334
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (4, 3), (4, 5), (6, 3), (3, 5)]	0.383955182
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 4), (3, 5), (3, 6), (4, 2), (4, 5), (6, 2), (7, 2)]	-25.006393632
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (5, 3), (5, 4), (6, 3), (4, 3)]	9.940971541
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 4), (3, 5), (3, 6), (5, 4), (6, 2), (7, 2), (2, 4)]	-3.905861001
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (4, 5), (5, 3), (6, 2), (6, 3), (7, 2)]	-0.985832772
[(1, 2), (1, 3), (2, 3)], [(1, 2), (1, 3), (1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)]	1.786399627
[(1, 2), (1, 3), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (3, 6), (3, 7), (5, 4)]	9.511790282
[(1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (4, 5), (6, 3), (7, 3)]	1.107207173
[(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (3, 6), (3, 7), (4, 2), (4, 5), (6, 2), (7, 2)]	2.532212926
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 2), (3, 6), (3, 7), (5, 4), (6, 2), (7, 2), (2, 4)]	1.023099262
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 5), (6, 2), (6, 3), (7, 2), (7, 3), (2, 3)]	0.560118735
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 5), (5, 2), (6, 2), (6, 3), (7, 2), (7, 3)]	54.232770888
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 4), (2, 6), (2, 7), (5, 4), (6, 3), (7, 3), (4, 3)]	7.275893416
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 2), (4, 3), (4, 5), (6, 2), (6, 3), (7, 2), (7, 3)]	1.668157579
[(1, 5), (1, 6), (1, 7), (5, 4), (6, 2), (6, 3), (7, 2), (7, 3), (2, 3), (2, 4), (3, 4)]	-3.895552977
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 3), (4, 5), (5, 2), (6, 2), (6, 3), (7, 2), (7, 3)]	-9.035526087
[(1, 2), (1, 3), (1, 5), (1, 7), (2, 3), (2, 5), (2, 7), (3, 5), (3, 6), (5, 4), (6, 4)]	-14.360679634
[(1, 2), (1, 4), (1, 5), (1, 7), (2, 3), (2, 5), (2, 7), (4, 5), (4, 6), (5, 3), (3, 6)]	-17.00351824
[(1, 3), (1, 4), (1, 5), (1, 7), (3, 2), (3, 5), (3, 6), (4, 5), (4, 6), (5, 2), (7, 2)]	2.856552471
[(1, 2), (1, 4), (1, 6), (1, 7), (2, 3), (2, 5), (2, 7), (4, 5), (4, 6), (6, 3), (3, 5)]	-9.92706055
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 2), (3, 5), (3, 6), (4, 5), (4, 6), (7, 2), (2, 5)]	-6.678755687
[(1, 2), (1, 3), (1, 5), (1, 7), (2, 5), (2, 7), (3, 4), (3, 5), (3, 6), (5, 4), (4, 6)]	-1.806083067
[(1, 2), (1, 3), (1, 6), (1, 7), (2, 5), (2, 7), (3, 4), (3, 5), (3, 6), (6, 4), (5, 4)]	-5.291537445
[(1, 3), (1, 4), (1, 6), (1, 7), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (7, 2), (2, 5)]	31.528228871
[(1, 2), (1, 5), (1, 6), (1, 7), (2, 5), (2, 7), (5, 3), (5, 4), (6, 3), (6, 4), (3, 4)]	-19.605870676
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 4), (3, 5), (3, 6), (5, 2), (5, 4), (6, 4), (7, 2)]	-17.779946422
[(1, 3), (1, 6), (1, 7), (3, 2), (3, 4), (3, 5), (3, 6), (6, 4), (7, 2), (2, 5), (5, 4)]	1.160776335
[(1, 4), (1, 6), (1, 7), (4, 3), (4, 5), (4, 6), (6, 3), (7, 2), (2, 3), (2, 5), (3, 5)]	6.814864507
[(1, 5), (1, 6), (1, 7), (5, 2), (5, 3), (5, 4), (6, 3), (6, 4), (7, 2), (2, 3), (3, 4)]	-18.164073663
[(1, 4), (1, 5), (1, 6), (1, 7), (4, 5), (4, 6), (5, 3), (6, 2), (7, 2), (7, 3), (2, 3)]	-1.93399586
[(1, 5), (1, 6), (1, 7), (5, 3), (5, 4), (6, 2), (6, 4), (7, 2), (7, 3), (2, 3), (2, 4)]	33.092504473
[(1, 2), (1, 4), (1, 7), (2, 3), (2, 7), (4, 3), (4, 5), (4, 6), (3, 5), (3, 6), (5, 6)]	-13.82812206
[(1, 3), (1, 5), (1, 6), (1, 7), (3, 2), (3, 7), (5, 4), (5, 6), (6, 4), (7, 2), (2, 4)]	-22.159791554

Table 3: Values of c_H for K_5-intersecting families.