Online Ramsey Numbers

Anita Zheng, Angela Song, Dillon Kanne, David Gonzalez

September 1, 2018

Abstract

We explore the online Ramsey game and adjacent problems in the subgraph query game. We focus on the restricted online Ramsey game, where Builder plays on a finite vertex set, and its corresponding online Ramsey number. We introduce novel notions about this game and introduce Builder strategies that make a linear improvement in the number of edges played on the naive strategy when playing the online Ramsey game on $r(n, n)$ vertices.

We also investigate small cases of the subgraph query game where Builder's goal is to build K_3s. In these small cases, we prove bounds on $t(K_3, p, N)$: the maximum number of triangles Builder can create in N turns. Furthermore, we introduce the notion of the restricted subgraph query game and prove precise bounds in the K_3 game.

Finally, we provide a review of related literature in graph theory that provides insight to a special case of the subgraph query game. In particular, it proves bounds relating to the maximum number of copies of a subgraph a graph can contain given that it has a fixed number of edges.

1 Introduction

One of the overarching themes of contemporary combinatorics research is the idea that discrete, finite objects, when large enough, must admit a certain degree of structure. There is perhaps no statement more classically tied with this theme than Ramsey's theorem, which state that monochromatic cliques are guaranteed in sufficiently large 2-colored complete graphs. The following report will explore and prove results related to this quintessential problem. In particular we will focus on the game-theoretic variant known as the online Ramsey numbers. These numbers emerge from the online Ramsey game played between a Builder and a Painter who are trying to build monochromatic cliques and avoid monochromatic cliques respectively. The imperative question is how long this game lasts with ideal play.

We will explore notions related to both upper bounds and lower bounds of the online Ramsey numbers. To understand upper bounds, we will look at strategies for Builder that provably guarantee the game to end early. In particular, we will focus on ideal play in a finite vertex version of the game, a topic of general interest for Builder strategies. Within this section, it is proven that an unbounded amount (linear in vertex count) of edges can be saved in the general diagonal case. To explore lower bounds we look at a variant of the online Ramsey game called the subgraph query game (and its finite vertex counterpart) that approximately simulates a random Painter strategy. To be specific, we will examine the number of triangles that can be built in this game, in expectation, and prove a related result. Lastly, a review of the literature on fitting many copies of a graph in a larger graph is provided due to its apparent similarity of a non-probabilistic version of the subgraph query game.

1.1 Definitions and History

Throughout this paper, we use K_n to denote the complete graph on n vertices. If a graph has a subgraph that is complete on m vertices, we call the subgraph an m-clique. Due to Ramsey, we have the following theorem.

Theorem 1. *For any positive integers m, n, there exists a least positive integer $r(m, n)$ such that any red-blue coloring of the edges of $K_{r(m,n)}$ contains either a red m-clique or blue n-clique.*

We refer to these particular integers $r(m, n)$ as **Ramsey numbers**.

Our project is mainly concerned with a widely-studied variant of Ramsey numbers, called online Ramsey numbers. First we define the online Ramsey game, which is played between two players called Builder and Painter.

Fix positive integers m and n. The game takes place on an infinite vertex set, and starts with no edges built between any two vertices. Each turn, Builder chooses two non-adjacent vertices and builds an edge between them. Painter then paints the edge either red or blue. Builder's goal is to construct a red m-clique or blue n-clique as quickly as possible, and Painter's goal is to prevent Builder's win for as long as possible. Suppose Builder and Painter play optimally. We use $\tilde{r}(m, n)$ to denote the least number of turns Builder must take to ensure victory. These particular $\tilde{r}(m, n)$ are called **online Ramsey numbers**.

Online Ramsey numbers, first introduced by both Beck [B93] and by Kurek and Ruciński [KR05] independently, has been studied in detail already. One can trivially find an exponential bound on the online Ramsey number $\tilde{r}(m, n)$ using the classical Ramsey number as follows.

$$\frac{r(m,n)}{2} \leq \tilde{r}(m,n) \leq \binom{r(m,n)}{2}$$

However, unlike classical Ramsey numbers which have seen no exponential improvements in decades, a paper by Conlon [C09] proved an exponential improvement on the upper bound in the diagonal case for infinitely many n.

$$\tilde{r}(n,n) \leq 1.001^{-n}\binom{r(n,n)}{2}$$

Furthermore, through use of a random painter strategy, a paper by Conlon, Fox, Grinshpun, and He [CFGH18] found an exponential improvement to the lower bound in both the diagonal case

$$\tilde{r}(n,n) \geq 2^{(2-\sqrt{2})n+O(1)}$$

and the off diagonal case, where $m \geq 3$ is fixed and $n \to \infty$.

$$\tilde{r}(m,n) \geq n^{(2-\sqrt{2})m+O(1)}$$

Our project strives to better understand online Ramsey numbers. For reasons that will be elaborated on later, it is helpful to introduce a few variations on the game, which we call the restricted online Ramsey game and the subgraph query game.

Fix positive integers $m, n,$ and v. As before, the restricted online Ramsey game is played between two players called Builder and Painter. Furthermore, the objective of the game is exactly the same, and each turn consists of Builder building an edge and Painter painting it red or blue. However, this time Builder only has access to v vertices, i.e. Builder can only build edges between vertices of a fixed vertex set of size v.

We use $\tilde{r}(m, n; v)$ to denote the number of moves Builder must take to ensure victory in the restricted online Ramsey game (of course, this definition only makes sense when $v \geq r(m, n)$). We call $\tilde{r}(m, n; v)$ the **restricted online Ramsey number**. Many of the builder strategies used throughout [CFGH18] and [C09] use a naive strategy in the restricted online Ramsey game. Improvements made in that domain may allow for corresponding improvements in these bounds, motivating our work on these problems.

Finally, we define the **subgraph query game**, a game that simulates playing the online Ramsey game against a random painter. The game is played by a single player called Builder on an infinite vertex set that starts with no edges built.

Fix a graph G and a probability $p \in (0, 1)$. Each turn, Builder chooses two non-adjacent vertices and an edge is built between the two vertices with probability p. Builder's goal is to build a copy of G in as few moves as possible. Note that if Builder "fails" to build an edge between two vertices, Builder cannot try to build an edge between those two vertices later in the game, i.e. Builder cannot choose the same pair of vertices twice. We define $f(G, p)$ to be the least number of turns Builder must take to ensure victory with probability at least $\frac{1}{2}$.

As may be expected, the subgraph query number has a few variations of note. For a graph G, probability p and positive integer N, we define $t(G, p, N)$ to be the maximum expected number of copies of G that Builder can build in N turns (where, instead of Builder winning when he/she builds a copy of G, Builder has N turns to build as many copies of G as possible).

Finally, we define the **restricted subgraph query game**. Fix graph G, probability p and positive integer v. The restricted subgraph query game is played exactly like the original subgraph query game, except Builder can only build edges between vertices of a fixed vertex set of size v. We define $f(G, p; v)$ to be the least number of moves Builder must take to ensure victory with probability at least $\frac{1}{2}$.

In particular, the subgraph query game is interesting because it inspired the exponential improvement to the lower bound of the online Ramsey game mentioned earlier. The authors of that work conjecture that their bound on the online Ramsey number can be improved through corresponding improvements in the subgraph query game.

1.2 Outline

In this paper, several topics in Ramsey theory will be explored.

In **Section 2**, Builder strategies to guarantee winning the restricted online Ramsey game $\tilde{r}(n, n; r(n, n))$ as early as possible are explored. A proof to save $O(N)$ edges (where $N = r(n, n)$) using the notion of independent pairs is first presented, then a close examination of dependency graphs and their potential to make further improvements on the number of edges saved.

In **Section 3**, insights on how triangles are built in the subgraph query game will be presented. Understandings of $t(K_3, p, N)$ for all N can shed light on the behaviors of bigger cliques in the game and $t(K_m, p, N)$. Further, the subgraph query game on a limited vertex set will also be investigated.

In **Section 4**, a summary of Noga Alon's paper on the number of copies of a graph H in a graph with a given number of edges is presented. This is a special case of the subgraph query problem and gives some insight into how difficult graphs are to build.

2 On how to save edges in a $\tilde{r}(n, n; r(n, n))$ game

2.1 Introduction

In 1947, Erdős and Szekeres gave an upper bound on the diagonal Ramsey numbers, $r(n, n)$ on the order of 4^n [E47]. Since that time, the problem has garnered great interest, including several improvements on this bound. However, this basic upper bound has not been improved upon by an exponential factor. The analogous upper basic bound for the online Ramsey number, $\tilde{r}(n, n)$, is given by $\binom{r(n,n)}{2}$. In contrast to the classical Ramsey numbers, there has been an exponential improvement on this basic bound. In particular, David Conlon [C09] proved for infinitely many n that:

$$\tilde{r}(n, n) \leq 1.001^{-n} \binom{r(n, n)}{2}$$

To prove this upper bound, Conlon describes a strategy that consists of two distinct steps. Broadly speaking, first, Builder creates many high degree vertices, branching within the monochromatic neighborhoods to create a finite vertex set that only requires a small clique to be built in order to win. Play then continues on this finite vertex set designed to be big enough to guarantee the win. The transition between these two steps is optimized to minimize the total time, spending about equal time on each step. Unsurprisingly, a significant shortening of either step would shorten the strategy overall when connected with the new proper stopping point.

Conlon's paper heavily focuses on the first step and largely ignores the second. In fact, the algorithm given for playing the online Ramsey game on a finite amount of vertices is simply filling in all of the edges. For this reason, we have studied playing the online Ramsey game on finite vertex sets, trying to come up with a more efficient way to play as Builder in this case. In fact, we show that if you take $\tilde{r}(n, n; V)$ to be the online Ramsey number on V vertices, we have the following:

$$\binom{r(n,n)}{2} - \tilde{r}(n,n;r(n,n)) \geq \frac{r(n,n)}{2}$$

which is a significant improvement on the naive strategy. We will also explore further methods which may even yield stronger results.

2.2 Independent Pairs

For the remained of this section we let $N = r(n,n)$. For any graph G, we call a pair $(u,v) \in G$ of non-adjacent vertices an *unbuilt pair*, or simply a *pair*.

Definition. If the edges of G are colored red or blue, two vertex-disjoint pairs (u_1,u_2) and (v_1,v_2) in G are *independent* if there exists both a red edge u_iv_j and a blue edge $u_{i'}v_{j'}$ in G.

Lemma 2. *(PIP Lemma) Suppose in the course of the online Ramsey game that Builder can build a graph G which contains P pairwise independent pairs (PIPs) $p_1,\ldots,p_P$. Then, Builder can save at least $P-1$ edges.*

Proof. Builder's strategy from this point forward is to build all the edges other than $p_1,\ldots,p_P$. Once this is done, Builder needs only one more move to win. That is, there is some p_i which forces a monochromatic clique when it is built regardless of which color it is painted.

We know that building *all* the pairs $p_1,\ldots,p_P$ forces a monochromatic clique, because we are playing on $r(m,n)$ vertices. Also, independent pairs cannot lie in a monochromatic clique together, as there are edges of both colors between them. Thus, some single one of the p_i must force a clique when it is built. Builder just builds this one and saves the other $P-1$ moves. □

Lemma 3. *Suppose in the course of the online Ramsey game that Builder can build a graph G which contains s pairs $p_1,\ldots,p_s$ each independent to t pairs $q_1,\ldots,q_t$. Then, Builder can save at least* $\min(s,t)$ *edges.*

Proof. Builder's strategy from this point forward is to build all the edges other than $p_1,\ldots,p_s$ and $q_1,\ldots,q_t$. Once this is done, Builder only needs to build either $p_1,\ldots,p_s$ or $q_1,\ldots,q_t$ to win. This is because we know that building all pairs $p_1,\ldots,p_s$ and $q_1,\ldots,q_t$ forces a monochromatic clique. Also, independent pairs cannot lie in a monochromatic clique together. Thus, either $p_1,\ldots,p_s$ or $q_1,\ldots,q_t$ must force a clique. Builder just builds those and saves the other s or t moves. □

2.3 How to save $\Omega(N)$ edges using independent pairs

Definition. $K_{a,b}$ is a complete bipartite graph with a vertices on one side and b vertices on the other side.

To begin with, Builder builds a K_n within the N vertices. In order for the game not to end, the K_n must not be monochromatic, and thus there must exist a vertex V, among the edges emanating from which there is a red edge VU and a blue edge VW. Builder then completes a $K_{N/2,N/2}$, with V on the left side and U, W on the right side.

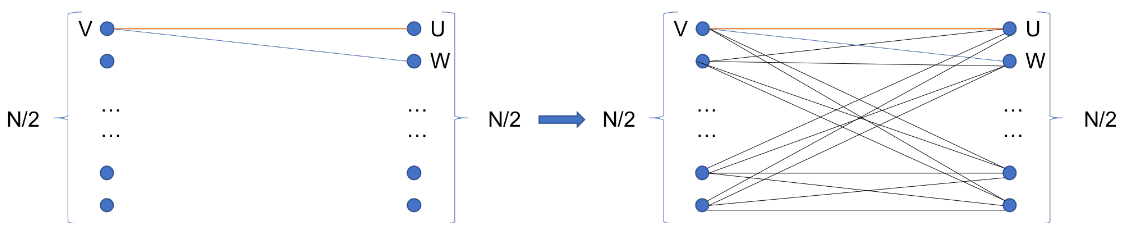


Figure 1: The $K_{N/2,N/2}$ that Builder builds to begin with.

Lemma 4. *In the $K_{N/2,N/2}$ that Builder builds to begin with, either:*
(i) there are $N/2 - n$ unbuilt pairs on the left independent to $N/2 - n$ pairs on the right;
(ii) the $K_{N/2,N/2}$ is monochromatic.

Proof. If the $K_{N/2,N/2}$ is not monochromatic, then there exists a vertex u on the left that has nonempty red neighborhood $R(u)$ and blue neighborhood $B(u)$. For any u' in the left of the $K_{N/2,N/2}$, the pair (u, u') is independent to any pair (v, w) on the right, where $v \in R(u)$ and $w \in B(u)$. Thus, there exists $N/2 - 1$ pairs on the left independent to at least $|R(u)| \times |B(u)|$ pairs on the right. Among the $N/2 - 1$ pairs on the left, edges between at most $n - 1$ pairs might have already been built. Among the $|R(u)| \times |B(u)|$ pairs on the right, by similar reasoning there are at least $N/2 - n$ unbuilt pairs. Hence, there are $N/2 - n$ unbuilt pairs on the left independent to $N/2 - n$ pairs on the right.

Since VU is Red and VW is blue, the $K_{N/2,N/2}$ cannot be monochromatic. Hence, there are $N/2 - n$ pairs on the left independent to $N/2 - n$ pairs on the right. □

Theorem 5. *In an online Ramsey game of $\tilde{r}(n, n; r(n, n))$, Builder is able to leave $\Omega(N)$ edges unbuilt when winning the game.*

Proof. By Lemma 4, Builder can construct $\Omega(N)$ pairs independent to $\Omega(N)$ pairs. And by Lemma 3, Builder will be able to save $\Omega(N)$ edges. □

2.4 Dependency graphs and potential future improvements

In order to understand our results more abstractly and suggest potential future avenues for improvement, we define the independence graph on pairs.

Definition. Given a two edge colored graph G, the **independence graph** is the graph with the vertex set $\{e | e \text{ is an pair of } G\}$ and edges $\{(e, f) | e \text{ is independent of } f\}$.

Our above results can be reinterpreted in the language of independence graphs. For example, we observed that if there is a complete bipartite subgraph in the independence graph, we can save all of the pair on one of the parts in the online Ramsey game. Similarly, if there is a clique in the independence graph we can save all but one of these pair in the online Ramsey game.

One useful feature of this abstraction is that it can be extended to encode even more information. To illustrate this point, we first define a distinction between different sorts of dependent pairs.

Definition. Two pairs are **red dependent** if the induced subgraph on the four vertices within the pair is exactly a monochromatic red $K_{2,2}$.

A similar definition can be created for blue dependency. In some contexts, we may want to take a stricter notion of dependence. For this reason, we can define the notion of deep dependence.

Definition. k pairs are **k-deep red dependent** if drawing all k edges in red creates a red K_n.

A similar definition can be created for k-deep blue dependency. Thankfully, all of these notions can be captured in the independence graph by simply adding colored edges between pairs that exhibit that colors dependence. Remarkably, this extra information may allow us to observe a completely different sort of edge saving structure in the dependency graph.

Lemma 6. *Let H be a non-trivial subgraph of the dependency graph with no blue edges. Builder can always save all but ℓ edges where ℓ is the size of the largest red clique.*

Proof. Builder first draws a pair for each blue edge in the dependency graph, reducing the graph down to a purely red and white subgraph. Let M denote the set of pairs that are in 1-deep blue dependent edges and M' denote the set of pairs that are not. Note that no pair can be in k-deep blue dependent edges for $k > 1$ as there are no blue dependencies left, so M and M' cover all pairs. Consider the coloring of the underlying graph which colors M red and M' blue. We know there are no blue cliques in this graph as none of the edges in M' are 1-deep blue dependent. Therefore, there is a red clique using some of the red edges in M. At most, this clique uses ℓ pairs. Builder can proceed to build these ℓ pairs. If any are blue, builder wins as the are 1-deep blue dependent. If all are red, builder wins as a red clique has been made. □

2.5 Evidence for the existence of further improvements

While a savings for Builder on the order of the diagonal Ramsey number is a promising start, there is some reason to believe that significantly larger savings can be made. In particular, in small cases, it can be demonstrated that a large portion of the edges are saved.

In particular, we have the following:

Theorem 7.

$$\tilde{r}(4,4;18) \leq 41$$

Note that $r(4,4) = 18$ and $\binom{18}{2} = 154$ [GG55]. To prove this theorem we prove an important lemma from which the theorem immediately follows by the Erdős-Szekeres method (see [E47] or [C09] for a description of this strategy):

Lemma 8.

$$\tilde{r}(3,4;9) \leq 24$$

Note that $r(3,4) = 9$ and $\binom{9}{2} = 36$ [GG55]. The following subsections are a description of a Builder strategy in the $(3,4;9)$ game that wins in 24 moves with justification, which constitutes a proof of the lemma. By convention, we take the first color to be blue and the second to be red.

2.5.1 The 3-5 Rule

Lemma 9. *For each vertex, if all 8 edges from this vertex are built, then exactly 3 of the 8 edges are red and exactly 5 of the 8 edges are blue, or Builder wins in 8 moves.*

Proof. If the rule is broken, then the Erdős-Szekeres method can be used to get a red 3-clique (if there are 4 red neighbors) or blue 4-clique (if there are 6 red neighbors). In particular, this will only take at most $\max(\tilde{r}(3,3;8), \tilde{r}(2,4;6)) = \max(8,6) = 8$ more moves. □

Hence, in the following subsections, we can assume that this rule is not broken before $24-8 = 16$ moves are played.

2.5.2 Branching out from two vertices

The strategy starts by branching out all 8 edges from a vertex, and by the 3-5 rule we know that exactly 3 of the 8 edges are red and exactly 5 of the 8 edges are blue. Then we pick one of the blue vertices and branch out the remaining 7 edges from it. By the 3-5 rule we know that exactly 3 of the 7 edges are red and exactly 4 of the 7 edges are blue.

If the red codegree of these vertices is 2 or 3, the blue codegree must be at least 3. If any of these edges are painted blue, Builder wins immediately as a blue K_4 is formed by the two branching vertices and the two newly blue connected vertices. Therefore, completing a K_3 within the blue coneighborhood wins, as to avoid the mentioned pitfall all of these edges must be red, creating a red K_3. This wins in 18 moves.

If the red codegree of these vertices is 0, Builder can simply complete the red neighborhood of one of the branching vertices, v. If any of these edges are painted red, a red K_3 is formed with the vertices in the edge and v. If they are all blue a blue K_4 is formed with the branching vertex that is not v. This also wins in 18 moves.

Therefore, we can conclude that these vertices have red codegree 1.

2.5.3 Timer Method

To proceed further in a productive way, we define the notion of a timer.

Definition. A pair has a blue t-timer if painting red the (built) edge between them results in a Builder win in t moves.

A red t-timer is similarly defined.

The following results about timers in the $(3,4;9)$ game are not difficult to observe and will be used to justify the remainder of the strategy:

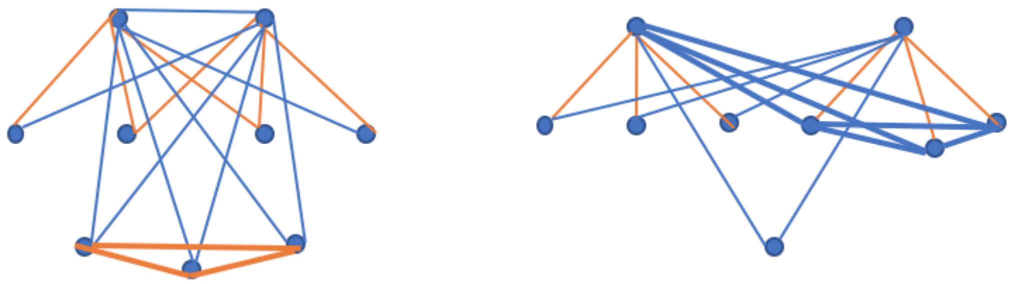
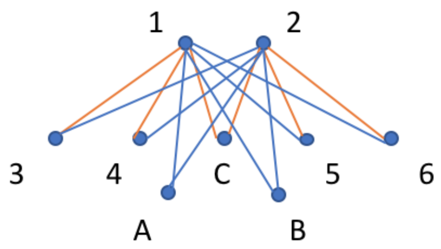


Figure 2: The final positions if the red codegree is too high or too low

1. If two edges of a triangle are both red, then a blue 1-timer can be put on the third edge. Likewise, if 5 edges of a 4-clique are blue, then a red 1-timer can be put on the sixth edge.
2. If the first edge of a triangle is built red, and the second edge is a red t-timer, then we can put a blue $t+1$-timer on the third edge. Likewise, if four edges of a 4-clique are built and painted blue, and one other edge has been put a blue t-timer, then a red $t+1$-timer can be put on the last edge of this 4-clique.
3. 3. If two of the edges of a 3-clique have been put a red a-timer and a red b-timer on them, respectively, then WLOG suppose $a \geq b$. Then a blue $max(a+2, b+1)$-timer can be put on the last edge.
4. If blue timers have been put on all t unbuilt edges of a 4-clique, and the built edges of the clique are all blue, then it takes at most $max(a_1, a_2+1, \cdots, a_t+t-1)$ moves for the Builder to win the game, where $a_1 \geq \cdots \geq a_t$ are the values of the blue timers.

2.5.4 The Final Moves

For clarity, each vertex is given a name. Call the two branching vertices 1 and 2. Call their unique common red neighbor C and their two common blue neighbors A and B. The two red vertices from 1 that are blue from 2 are called 3 and 4 and the two blue vertices from 1 that are red from 2 are called 5 and 6.



After the 15 branching moves, note we can put a red 1-timer on the edge AB, as explained above. Furthermore, we know that blue 1-timers can be put on edges 34, $3C$, $4C$, 56, $5C$, $6C$ due to possible red K_3s with A and B.

The Builder then builds CA and CB. If both of them are red, then AB gains a blue 1-timer due to the possibility of ABC. Of course, building AB now wins immediately. It remains to consider the case in which at least one of these edges is blue. WLOG we suppose that CA is blue.

Now we build $B3$ and $B5$. Consider the following cases:

1. If both of them are red, then a blue 1-timer can be put on 35. A blue 2-timer can be put on $A3$ by checking the triangle $AB3$; likewise, a blue 2-timer can be put on $A5$. Hence now by checking the four clique $A35C$, the Builder can win in at most $max(2, 2+1, 1+2, 1+3, 1+4) = 5$ moves. Hence in total we need $15+2+2+5 = 24$ moves.

2. If exactly one of them is blue, WLOG suppose $B3$ is blue and $B5$ is red. By checking the 4-clique $23B4$, we know that a red 2-timer can be put on $B4$. Then a blue 3-timer can be put on 45. A blue 3-timer can be put on $A4$ by checking the triangle $AB4$; likewise, a blue 2-timer can be put on $A5$. Hence now by checking the four clique $A45C$, the Builder can win in at most $max(3, 3+1, 2+2, 1+3, 1+4) = 5$ moves. Hence in total we need $15+2+2+5 = 24$ moves.

3. If both of them are blue, by checking the 4-clique $23B4$, we know that a red 2-timer can be put on $B4$. Likewise, a red 2-timer can be put on $B6$. Then a blue 4-timer can be put on 46. A blue 3-timer can be put on $A4$ by checking the triangle $AB4$; likewise, a blue 3-timer can be put on $A6$. Hence now by checking the four clique $A46C$, the Builder can win in at most $max(4, 3+1, 3+2, 1+3, 1+4) = 5$ moves. Hence in total we need $15+2+2+5 = 24$ moves.

Now we have shown the required lemma and thus the theorem.

3 Understanding triangles in the subgraph query game

3.1 Motivation for understanding triangles

Recall that $f(G, p)$ is the number of turns needed to create a single copy of G in the subgraph query game. Theorem 13 of [CFGH18] fully describes the behavior of $f(K_3, p)$, stating that $f(K_3, p) = \Theta(p^{-3/2})$. Furthermore, this theorem gives tight bounds on K_4 and K_5 but fails for K_6; their methods bound $f(K_6, p) \leq O(p^{-10/3})$ and $f(K_6, p) \geq \Omega(p^{-3})$. They conjecture that their upper bound is ideal up to a constant factor, implying that their lower bound can be improved dramatically, especially for K_m with $m \gg 6$.

To understand why this discrepancy exists, we need to look at where these bounds on $f(K_m, p)$ come from: bounds on $t(K_m, p, N)$. Recall that $t(K_m, p, N)$ is the number of copies of K_m Builder can create in expectation in N turns. The lower bound $f(K_3, p) \geq \Omega(p^{-3/2})$ comes found the upper bound $t(K_3, p, N) \leq O(N^2p^3)$. This bound can be derived from the methods discussed in Section 5.3 of the same paper. One can see intuitively why the bound on f follows: if Builder can create at most N^2p^3 triangles in N turns, surely Builder needs at least $p^{-3/2}$ turns to make a single one.

However, as N grows much larger, this upper bound on $t(K_3, p, N)$ becomes weaker and weaker. In fact, skipping over a few details of the theorem statement, Theorem 15 of [CFGH18] states that after a certain threshold, $t(K_3, p, N) \approx \Theta(N^{3/2}p^3)$, a whole factor of $\sqrt{N}$ better than the bound for all N. This is emblematic of how our original bounds on K_m for small N do not match the better bounds for larger N. This becomes a problem when upper bounding $t(K_6, p, N)$ because any strategy that makes a even a single K_6 makes many triangles and takes at least $\Omega(p^{-3})$ turns, but we will show that the original upper bound for $t(K_3, p, N)$ is weak for $N \geq \Omega(p^{-3})$.

Our upper bound for $t(K_6, p, N)$ cannot be easily adapted to incorporate this $\Theta(N^{3/2}p^3)$ improvement or other similar improvements on K_3. In order to possibly shed light on how we can improve our upper bounds on $t(K_m, p, N)$ with $m \geq 6$, we look to fully understand $t(K_3, p, N)$ for *all* N, not just small and large N. We explore this idea in the next subsection. Moreover, just as the previous section discusses the online Ramsey game on a limited vertex set, we also briefly investigate $f(K_3, p; V)$: the subgraph query game on a limited vertex set.

3.2 What is Known About Triangles

As discussed previously, $t(K_3, p, N) \leq O(N^2p^3)$. Section 5.3 of the previously cited paper gives a full proof of this statement and a way to generalize a similar statement to all K_m. To quickly see why this is true, consider the largest number of P_2 (paths of length 2) one can create in N moves. Because every vertex in the graph must have degree less than or equal to Np, the maximum number of P_2 one can have is when all edges come out of the same vertex. This is $O(N^2p^2)$ P_2s. Finally, because each P_2 can become at most 1 triangle in exactly 1 way (adding the last edge with probability p), builder cannot make more than $O(N^2p^3)$ triangles.

Moreover, Section 5.4 of the same paper uses a radically different method to show that when $N \geq \Omega(p^{-4}\log^2(p))$, $t(K_3, p, N) \leq O(N^{3/2}p^3)$. The details of this theorem and its proof aren't necessary for this work but indicate how $t(K_3, p, N)$ behaves for large enough N.

Furthermore, each of these upper bounds comes with a paired strategy, giving a tight lower bound as well. For small N, one can branch out of a single vertex Np times and connect all of the pairs of edges to get $\Theta(N^2p^3)$ total triangles. This strategy works until the second half of the strategy, connecting all of the pairs, uses more than N moves. Therefore, N^2p^2 must be less than N, so this strategy only works up until $N = O(p^{-2})$. Likewise, to match the second upper bound, one can always connect all pairs of $\Theta(\sqrt{N})$ vertices. Each triple of vertices has a p^3 chance of becoming a triangle and there are $\Theta(N^{3/2})$ such triples, giving the desired result.

However, there exists a third strategy that bridges the gap between $N = p^{-2}$ and $N = p^{-4}\log^2(p^{-1})$.

Lemma 10. *For $N \geq \Omega(p^{-2})$, $t(K_3, p, N) \geq \Omega(Np)$.*

Proof. Branch out from a single vertex v $\Theta(N)$ times. In expectation, this creates $\Theta(Np)$ edges out of v. Connect as many pairs of vertices in $N(v)$ in the remaining $\Theta(N)$ moves. Note that because $N \geq \Omega(p^{-2})$, the $|N(v)| = \Theta(N^2p^2) \geq \Omega(N)$. Each of these edges results in a triangle, so $\Theta(Np)$ triangles are created, resulting in the upper bound in the theorem statement. □

One may notice that this strategy is morally equivalent to the strategy for $N = \Theta(p^{-3/2})$ that creates a single triangle. The only difference is that when $N \geq \Omega(p^{-2})$, $|\mathcal{N}(v)| \geq N$, so connecting all pairs becomes impossible once $N > \omega(p^{-2})$. To be clear, note that $\mathcal{N}(v)$ means the neighborhood function, not N. Furthermore, this strategy is beaten by the strategy that fills in all pairs between $\sqrt{N}$ vertices when $Np = N^{3/2}p^3$, which occurs when $N = p^{-4}$.

In summary, we know that for $N \leq O(p^{-2})$, $t(K_3, p, N) = \Theta(N^2p^3)$ and for $N \geq \Omega(p^{-4}\log^2(p^{-1}))$, $t(K_3, p, N) = \Theta(N^{3/2}p^3)$. If we ignore the log factors in the cutoff for the second bound, we have seemingly fully understood the behavior of $t(K_3, p, N)$ for all N below p^{-2} and above p^{-4}. We explore an upper bound that matches our $\Theta(Np)$ strategy in this middle regime in the next subsection.

3.3 New Upper Bound on Triangles Created

We will show the following theorem which improves currently known upper bounds on $t(K_3, p, N)$ for larger N.

Theorem 11. *$t(K_3, p, N) = \Theta(Np)$ for $p \geq \Omega(p^{-2})$ and $p \leq O(p^{-3})$*

To get this, we will show the following lemma about $t(K_3, p, N)$ in general.

Lemma 12. *$t(K_3, p, N) \leq O(\max(Np, N^{3/2}p^{5/2}))$ for all N.*

Theorem 11 follows as a corollary, because of $N^{3/2}p^{5/2} \leq Np$ for $N \leq p^{-3}$ and lemma 10. This proves the tight bound we alluded to earlier for half of the interval from p^{-2} to p^{-4}.

Proof. Consider some strategy S in the subgraph query game. For each edge e queried in the strategy, define $P_2(e)$ to be the number of paths of length 2 that begin and end at the end at the endpoints of e. We will look to bound $\sum_{e \in S} \mathbb{E}[P_2(e)]$, where $e \in S$ means every edge queried during the strategy S.

To do this, consider the number of cycles of length 4 created during the strategy. If C is the number of cycles of length 4 created during S, we can see that

$$t(C_4, p, N) \geq \mathbb{E}\left[\sum_{e \in S} \binom{P_2(e)}{2}\right],$$

because each pair of unique P_2 that start and end at the endpoints of a particular edge e contributes a single cycle of length 4. Then, we can use the recursive graph building methods described in Section 5.3 of [CFGH18] to bound $t(C_4, p, N) \leq O(N^2p^3)$. Also, note that our count above can overcount the number of C_4 by up to factor of two (one for each diagonal of the C_4). By clearing out this constant and the 1/2 from the binomial into the big O while using properties of expected value,

$$O(N^2p^3) \geq \sum_{e\in S} \mathbb{E}\left[P_2(e)^2\right] - \sum_{e\in S} \mathbb{E}\left[P_2(e)\right] \geq \sum_{e\in S} \mathbb{E}\left[P_2(e)\right]^2 - \sum_{e\in S} \mathbb{E}\left[P_2(e)\right]$$

Because exactly N edges are attempted in S, we can use Cauchy-Schwarz to find

$$O(N^2p^3) \geq \frac{1}{N}\left(\sum_{e\in S} \mathbb{E}\left[P_2(e)\right]\right)^2 - \sum_{e\in S} \mathbb{E}\left[P_2(e)\right]$$

Now, this is a quadratic in the sum $\sum_{e\in S} \mathbb{E}\left[P_2(e)\right]$. Solving this quadratic reveals

$$\sum_{e\in S} \mathbb{E}\left[P_2(e)\right] \leq O(\sqrt{N^2 + N^3p^3}) \leq O(\max(N, N^{3/2}p^{3/2})).$$

The left hand side is simply the expected number of P_2 that could possibly become triangles under the strategy S. We can then observe that at each turn of the strategy S, we can hope to complete $\mathbb{E}\left[P_2(e)\right]$ total triangles with the placement of our edge e. Because the builder has no knowledge of whether e will succeed or not before they place it, we can expect to create precisely $p \cdot \mathbb{E}\left[P_2(e)\right]$ triangles in that move. So, by factoring the p, over the course of the whole game we expect to create exactly

$$p\sum_{e\in S} \mathbb{E}\left[P_2(e)\right] \leq O(\max(Np, N^{3/2}p^{5/2}))$$

triangles. Because this holds for any strategy S, this gives us our desired result. □

We comment on the proof method used here more in the discussion at the end of the section, but it does present interesting questions that could lead to similar improvements for $t(K_m, p, N)$ in general.

3.4 Triangles on a Restricted Vertex Set

In addition to studying $t(K_3, p, N)$, we also briefly explored the subgraph query game on a limited vertex set. Namely, we explored $f(K_3, p; V)$: the subgraph query game with only V vertices. In this subsection, we prove the following theorem by demonstrating a strategy for the upper bound in Lemma 15 and showing that no strategy can do better in Lemma 14.

Theorem 13. *$f(K_3, p; V) = \Theta(p^{-2+\alpha})$ when $V = \Theta(p^{-1-\alpha})$ for all $\alpha \in [0, \frac{1}{2}]$.*

One can immediately see where the upper bound for the extreme cases ($V = p^{-3/2}$ and $V = p^{-1}$) comes from. In the $\alpha = \frac{1}{2}$ case, Builder can simply perform the regular strategy for K_3 ignoring the vertex restriction. For the $\alpha = 0$ case, the only way Builder can expect to produce a single triangle is if they query all pairs, taking all p^{-2} moves. The theorem guarantees that the best Builder can do is linearly (in the exponent) interpolate between these strategies. This interpolation will become explicitly clear in Lemma 15.

First, we show that no strategy can do better than this linear interpolation.

Lemma 14. *$f(K_3, p; V) = \Omega(p^{-2+\alpha})$ when $V = \Theta(p^{-1-\alpha})$ for all $\alpha \in [0, 0.5]$.*

Proof. The theorem statement reduces to showing that any strategy S that creates a constant number of triangles on $\Theta(p^{-1-\alpha})$ vertices must use $\Omega(p^{-2+\alpha})$ moves.

Consider any strategy S. First, note that because S creates a constant number of triangles, it must create at least $\Omega(p^{-1})$ P_2s, because P_2 is the only graph on three vertices with 2 edges. Furthermore, each P_2 can become at most 1 triangle with probability p.

Now, let us consider how many P_2 any particular successfully built edge participates in at the end of the strategy. Because there are $\Theta(p^{-1-\alpha})$ vertices, if we try to have a particular edge participate in the most P_2 possible by connecting both ends of the edge to as many vertices as possible, we get a total of

$$2 \cdot p \cdot \Theta(p^{-1-\alpha}) = \Theta(p^{-\alpha})$$

possible P_2 that edge can participate in. So, each edge participates in $O(p^{-\alpha})$ P_2.

Now, we will calculate the number of P_2 in the graph. Call E the number of edges the strategy creates, K the average number of paths of length 2 each edge participates in (in the strategy), and P the total number of P_2. Clearly,

$$E \cdot K = P$$

Furthermore, our calculations earlier indicate that $O(p^{-\alpha}) \geq K$ and $P \geq \Omega(p^{-1})$. So,

$$E \cdot O(p^{-\alpha}) \geq E \cdot K = P \geq \Omega(p^{-1}).$$

It follows that $E \geq \Omega(p^{-1+\alpha})$. Because it takes p^{-1} moves to make a single edge, this means that the number of taken in S is at least $p^{-2+\alpha}$, giving the desired result (up to constants). □

Now, we present a strategy that achieves this lower limit and creates a constant number of triangles. Note that if a strategy creates any constant number of triangles, we can simply repeat the entire strategy a constant number of times to create *any* constant number of triangles, which wins the game with high enough probability by the same logic presented in [CFGH18]. Also, note that we gloss over asymptotic notation (namely Θ) when it is cumbersome.

Lemma 15. *$f(K_3, p; V) = O(p^{-2+\alpha})$ when $V = \Theta(p^{-1-\alpha})$ for all $\alpha \in [0, 0.5]$.*

Proof. Choose any $k = p^{2\alpha-1}$ vertices in some canonical order $a_1, \ldots, a_k$. Call this set $A = \{a_1, \ldots, a_k\}$ and put all other vertices in B. It is important to note here that because $|A| \leq \Theta(p^{-1-\alpha})$, most of the vertices in the graph will be in B, which has size $\Theta(p^{-1-\alpha})$. First, connect a_1 to every vertex in B. Because $|B| = \Theta(p^{-1-\alpha})$ and each edge has probability p of being built, a_1 will have $\Theta(p(p^{-1-\alpha})) = \Theta(p^{-\alpha})$ edges out of it to some $\Theta(p^{-\alpha})$ vertices in B. Remove all vertices with successfully constructed edges from B.

Then, repeat this process for a_2 with the new B that have no edge between themselves and a_1 (all the failed vertices). Repeat this process for all vertices $a_i \in A$; attempt connecting them to every vertex in B that failed for all previous a_j coming before a_i.

Finally, try to connect each pair of vertices in $\mathcal{N}(a_i)$ for every $a_i \in A$. Connecting any of these creates a triangle, and finishes the problem.

First, we analyze the number of edges in this strategy. We can see that in each of the "branches" in the first step, we take at most $\Theta(p^{-1-\alpha})$ moves, one for each vertex in B which has the size specified earlier. Because we do this $p^{2\alpha-1}$ times, this is a total of $\Theta((p^{2\alpha-1})(p^{-1-\alpha})) = \Theta(p^{-2+\alpha})$ moves, which is our specified target number of moves. Then, in the second step of the process, we are connecting all $p^{-\alpha}$ vertices together, which takes $p^{-2\alpha}$ moves. Again, repeating this $p^{2\alpha-1}$ times, this step takes p^{-1} moves, which is smaller than $\Theta(p^{-2+\alpha})$, so this strategy works in less than $O(p^{-2-\alpha})$ moves.

Now, we analyze the number of triangles this strategy expects to create. One can check that for reasonably small values of p ($p < 0.5$ perhaps), the size of B never shrinks by more than a constant factor over the course of the strategy. Then, each $\mathcal{N}(a_i)$ has size $\Theta(p^{-\alpha})$, so we create a triangle with probability $p^{1-2\alpha}$ because drawing an edge is probability p and we attempt $\Theta(p^{-2\alpha})$ total edges. Then, repeating this $p^{2\alpha-1}$ times for each $a_i \in A$, we see that this is precisely a constant number of triangles made in expectation. □

3.5 Discussion of Methods

We believe that the methods presented, especially the methods in the proof of theorem 13 give rise to interesting questions that could allow us to better understand $t(K_m, p, N)$ in general. In particular, it gives an upper bound on $t(K_3, p, N)$ using an upper bound on a *larger* graph, namely $t(C_4, p, N)$. The other methods known for upper bounding $t(K_m, p, N)$ as described in Section 5.3 of [CFGH18] all bound graphs using smaller graphs (either in vertices or edges). We believe this could be a promising way to understand the subgraph query game. We are currently looking to directly generalize the proof strategy from theorem 13 to graphs of arbitrary size, not just K_3 and C_4.

We still have a gap in understanding of $t(K_3, p, N)$: N between p^{-3} and p^{-4}. There are ways of adapting the proof method shown here to this problem, but it requires understanding how to bound $\sum \mathbb{E}[P_2(e)]$ when $\mathbb{E}[P_2(e)] \gg \omega(1)$. We are investigating possibly extending the logic used in Section 5.4 of the same paper or bounding $t(C_4, p, N)$ when the C_4 overlap in many vertices. Nevertheless, whatever proof strategy leads to success in this domain could also shed insight on how to understand t in general.

4 The number of copies of a graph H in a graph with a given number of edges

4.1 Introduction

This section is a summary of Noga Alon's *On the number of subgraphs of prescribed type of graphs with a given number of edges* [A81]. As such, all results in this section are due to Alon. The paper, as suggested by the title, bounds the total number of copies of a graph H that can be found in a graph G with ℓ edges.

The paper is relevant to our project because it can be considered as a special case or variation of our subgraph query problem—namely, if Builder builds each edge with probability 1, what is the maximum number of copies of H Builder can build in ℓ turns? In other words, this paper helps us better explore Builder strategies to maximize success in the subgraph query game.

4.2 Definitions

Note that all graphs considered in this paper are finite, simple and contain no isolated vertices. Furthermore, all graphs considered are unlabelled.

If S is a subset of vertices of graph G, let $N_G(S)$ denote the neighborhood of S in G, where $S \cap N_G(S)$ may be nonempty if vertices in S are adjacent to one another.

Definition. Suppose G and H are graphs. Then $N(G, H)$ is defined to be the number of subgraphs of G isomorphic to H, i.e. $N(G, H)$ is the number of copies of H contained in G.

Definition. Suppose H is a graph and ℓ is a positive integer. Then $N(\ell, H)$ is defined to be $\max(N(G, H))$, where the maximum is taken over all graphs G with ℓ edges.

Definition. Suppose H is a graph. Call H **asymptotically extremally complete** (a.e.c. for short) if for all positive integers ℓ, there exists a positive integer n such that

$$\binom{n}{2} \leq \ell$$

and

$$N(\ell, H) = (1 + O(\ell^{-1/2}))N(K(n), H).$$

If graph H is a.e.c., we may think of it as a graph that is "easily found" in complete graphs. Largely, a.e.c. graphs are important in the proofs of the main results because $N(K(n), H)$ can easily be estimated.

Definition. Suppose H is a graph on vertex set V. Then $\delta(H)$ is defined to be $\max(|S| - |N_H(S)|)$, where the maximum is taken over all subsets S of V.

This gives us the vocabulary necessary to describe the interesting results of the paper.

4.3 Main results

Theorem 16. *Suppose H is a graph on v vertices, and suppose $\delta(H) = 0$. Then*

$$N(\ell, H) = c_H \ell^{v/2} + O(\ell^{v/2 - 1/2})$$

where c is a constant dependent on H.

Theorem 17. *Suppose H is a graph on v vertices, and suppose $\delta(H) > 0$. Then*

$$N(\ell, H) = \Theta(\ell^{(v+\delta(H))/2}).$$

As the above theorems suggest, the "difficulty of building" a graph H is determined by the size of H and $\delta(H)$. If we fix the size of the vertex set of H, the larger $\delta(H)$ is, the easier H is to build.

If we consider the star $K(1, n)$, the complete bipartite graph between 1 vertex and n vertices, we can gain a better intuition of the above theorems. As one easily verifies, $\delta(K(1, n)) = n - 1$ is large for a graph on $n + 1$ vertices. By the theorems above, we thus expect $K(1, n)$ to be "easy to build."

One easily checks that $N(\ell, K(1, n)) \geq N(K(1, \ell), K(1, n)) = \binom{\ell}{n}$. But since $K(1, n)$ has n edges, clearly $N(\ell, K(1, n)) \leq \binom{\ell}{n}$. So we have $N(\ell, K(1, n)) = \binom{\ell}{n} = \Theta(\ell^n)$. This matches Theorem 13.

More generally, this example shows that a graph with large δ achieves the rather naive upper bound of $N(\ell, \text{graph with } n \text{ edges}) \leq \binom{\ell}{n}$. So we begin to have the sense that graphs with large δ maximize "buildability." This confirms the general idea of the two theorems.

4.4 Outline of the proofs

The general proofs are as follows. Note that some parts we do not comment on because they can be shown through basic combinatorics.

1. Show H is a.e.c. if and only if $N(\ell, H) = c_H \ell^{v/2} + O(\ell^{(v-1)/2})$ for a constant c dependent on H.

2. Prove that if H' is a spanning subgraph of H and H' is a.e.c., then H is a.e.c.

3. Show that if H is the disjoint union of $H_1, H_2, \ldots, H_n$, where H_i is a.e.c. for all i, then H is a.e.c.

4. Demonstrate $\delta(H) = 0$ if and only if there exists a spanning subgraph H' of H that is a disjoint union of cycles and independent edges.

   This part involves a clever use of Hall's Marriage Theorem, and will be elaborated on in the future.

5. Show that cycles (and independent edges) are a.e.c.

   This is another tricky section, but carefully using induction on length of cycle yields the desired result. Note that completing this part gives us Theorem 12—if $\delta(H) = 0$, then by part 4 H has a spanning subgraph that is a disjoint union of cycles and independent edges. By this part cycles and independent edges are a.e.c., so by part 3 we have H is a.e.c. Applying part 1 gives us exactly Theorem 12.

6. Prove that if $\delta(H) > 0$, $N(\ell, H) \geq c_1 \ell^{(v+\delta(H))/2}$ for some constant c_1.

   Another tricky part. However, for the sake of succinctness, we will omit a detailed discussion of this section.

7. Show that if $\delta(H) > 0$, $N(\ell, H) \leq c_2 \ell^{(v+\delta(H))/2}$ for some constant c_2.

   Note that this part and part 6 are a restatement of Theorem 13. I show a proof of this step in the following section.

4.5 Further proof details

The following is the restatement and proof of part 4.

Lemma 18. *Suppose H is a graph. Then $\delta(H) = 0$ if and only if H has a spanning subgraph H' that is a disjoint union of cycles and independent edges.*

Proof. Note that $\delta(H) = 0$ if and only if for all subsets S of the vertex set of H, $|S| \le |N_H(S)|$.

First suppose that H has a spanning subgraph H' with the desired properties. For any subset S of the vertex set of H, it is easy to check that

$$|S| \le |N_{H'}(S)| \le |N_H(S)|$$

so $\delta(H) = 0$, as desired.

Now, suppose H is such that $\delta(H) = 0$. Suppose the vertex set of H is $\{v_1, v_2, \dots, v_n\}$. Build a bipartite graph G with bipartition (X, Y), where $X = \{x_1, x_2, \dots, x_n\}$, $Y = \{y_1, y_2, \dots y_n\}$, and x_i is connected to y_j if and only if v_i is connected to v_j in H.

We first show that for all $S \subset X$, $|S| \le |N_G(S)|$. Then we may apply Hall's Marriage Theorem, which proves that G has a perfect matching (see [H69]). But because $|S'| \le |N_H(S')|$ for all $S' \subset \{v_1, v_2, \dots, v_n\}$, we immediately obtain $|S| \le |N_G(S)|$ for all $S \subset X$.

Therefore, G has a perfect matching M. Construct a subgraph H' of H where v_i is connected to v_j if and only if (x_i, y_j) is an edge in M. Clearly, H' is a spanning subgraph of H, so if we show that H' is the disjoint union of cycles and independent edges, we are done.

Consider any $v_i \in \{v_1, v_2, \dots, v_n\}$. By our construction of H', either v_i is a vertex of an independent edge (when (x_i, y_j) and (x_j, y_i) are both in M for some $j \ne i$), or v_i has degree 2 (when (x_i, y_j) and (x_k, y_i) are in M for some $j \ne k \ne i$). Then it is easy to see that either v_i is a vertex of an independent edge, or v_i is a vertex of a cycle. So H' is the disjoint union of cycles and independent edges. □

Now we restate and prove part 7. Note that in the following statement and proof, we use n, instead of v, to denote the number of vertices in the graph. This is for ease and clarity of notation.

Lemma 19. *Suppose H is a graph on n vertices with $\delta(H) > 0$. Then*

$$N(\ell, H) \le c_2 \ell^{(n+\delta(H))/2}$$

for some constant c_2 dependent on H.

Proof. Let $V = \{v_1, v_2, \dots, v_n\}$ be the vertex set of H. Suppose $S' \subset V$ achieves $\delta(H)$, i.e. suppose $|S'| - |N_H(S')| = \delta(H)$. Define $S = S' \cap N_H(S')$. We will show that $S \subset V$ is an independent vertex set that also achieves $\delta(H)$.

Clearly, S is an independent vertex set, i.e. H has no edges between vertices of S. Furthermore, $N_H(S) \subset N_H(S') \setminus (S' \cap N_H(S'))$, so we have

$$|S| - |N_H(S)| \ge |S'| - |N_H(S')| = \delta(H).$$

But by the definition of $\delta(H)$ we have $|S| - |N_H(S)| = \delta(H)$, so S is an independent vertex set that achieves $\delta(H)$.

Let $|S| = s$ and $|N_H(S)| = t$. Build a bipartite graph J with bipartition $(S, N_H(S))$, where $v_i \in S$ is connected to $v_j \in N_H(S)$ if and only if v_i and v_j are adjacent in H. We will apply Hall's Marriage Theorem to show that J contains a matching in which every vertex in $N_H(S)$ is in an edge of the matching.

To be able to apply Hall's Marriage Theorem, we must show that for any set $X \subset N_H(S)$, $|X| \le |N_J(X)|$. Suppose otherwise. Then there exists a set $X \subset N_H(S)$ such that $|X| > |N_J(X)|$.

Define $Y = S \setminus N_J(X)$. Note that Y is non-empty, as

$$Y = |S| - |N_J(X)| > |S| - |X| \ge |S| - |N_H(S)| = \delta(H) > 0.$$

Furthermore, we have $Y \subset N_H(S) \setminus X$ so

$$|Y| - |N_H(Y)| = |S| - |N_J(X)| - |N_J(Y)| \ge |S| - |N_J(X)| - |N_H(S)| + |X| \ge |S| - |N_H(S)| = \delta(H).$$

But this contradicts the definition of $\delta(H)$, so $|X| \le |N_J(X)|$ for all $X \subset N_H(S)$, and we may apply Hall's Marriage Theorem. Thus, J has a matching M that "saturates" $N_H(S)$.

Then we can find a spanning subgraph J' of J such that J' contains s edges (M touches t vertices of S. Since H contains no isolated vertices, we can find edges in J to touch the other $s-t$ vertices of S. We know that M touches all t vertices of $N_H(S)$ so we are done). So we have

$$N(\ell, J') \leq \binom{\ell}{s} \leq \frac{1}{s!}\ell^s.$$

Note that J' is also a spanning subgraph of $S \cup N_H(S)$. So through some basic combinatorics we can then show that $N(\ell, S \cup N_H(S)) \leq c_3 \ell^s$ for some constant c_3.

Now define $L = H \setminus (S \cup N_H(S))$. If we show that $N(\ell, L) \leq c_4 \ell^{(n-s-t)/2}$ for some constant c_5, we can easily show, through some more basic combinatorics, that $N(\ell, H) \leq c_5 \ell^{s+(n-s-t)/2} = c_5 \ell^{(n+s-t)/2} = c_5 \ell^{(n+\delta(H))/2}$, and we are done. But L is on $n-s-t$ vertices, so we are done if we prove that L satisfies the conditions of Theorem 12.

Thus, we need only prove that $\delta(L) = 0$, i.e. for all subsets P of the vertex set of L, $|P| \leq |N_L(P)|$. But assuming otherwise gives us a very similar contradiction to the one found above, so L satisfies the conditions of Theorem 12 and we are done.

□

5 Acknowledgements

We would like to thank our mentor, Xiaoyu He, for his continued guidance and patient support. Furthermore, we would like to thank Dr. George Schaeffer for organizing and overlooking all SURIM events, and Stanford University for the opportunity and funding to pursue this project. Finally, we are grateful to Dr. Jacob Fox for his valuable input and suggestions regarding the restricted online Ramsey game.

References

[CFGH18] Conlon, D., Fox, J., Grinshpun, A., & He, X. Online Ramsey Numbers and the Subgraph Query Problem. *arXiv preprint arXiv:1806.09726*, 2018.

[A81] Noga, A.:On the number of subgraphs of prescribed type of graphs with a given number of edges, *Israel Journal of Mathematics*, **38**, 116-130 (1981).

[H69] F. Harary, *Graph Theory*, Addison Wesley, Reading, Menlo Park and London, 1969, pp. 53-54.

[C09] Conlon, D., On-line Ramsey numbers. SIAM J. Discr. Math. 23 (2009/10), 1954–63.

[E47] Erdös, P. Some remarks on the theory of graphs. Bull. Amer. Math. Soc. 53 (1947), no. 4, 292–294.

[GG55] Greenwood, R. E., and Gleason, A.M.. "Combinatorial relations and chromatic graphs." Canad. J. Math 7, no. 1 (1955): 7.

[B93] J. Beck, Achievement games and the probabilistic method, in Combinatorics, Paul Erdős is Eighty, Vol. 1, 51–78, Bolyai Soc. Math. Stud., János Bolyai Math. Soc., Budapest, 1993.

[KR05] A. Kurek and A. Ruciński, Two variants of the size Ramsey number, Discuss. Math. Graph Theory 25 (2005), 141–149.