Building Forests in Maker-Breaker Games: Upper and Lower Bounds

SPUR Final Paper, Summer 2017
Justin Lim
Mentor: Frederic Koehler
Project suggested by Asaf Ferber

August 2, 2017

Abstract

We give explicit "linear-time" strategies for building spanning forests in $1:1$ Maker-Breaker games and matching lower bounds in some interesting cases, e.g. for a forest of k-cycles. Here a forest refers to a disjoint union of factors that span the graph. Furthermore, for the purposes of these upper bounds these forests can be arbitrary finite mixtures of different factor graphs.

1 Introduction

1.1 Maker-Breaker games

We are studying Maker-Breaker games on complete graphs. A Maker-Breaker game can be described as a game played on a hypergraph $H = (X, E)$, where X is a vertex set and $E \subset P(X)$, where P is the power set of X. Maker and Breaker take elements of X in turn. If Maker occupies all elements of some $e \in E$, then Maker wins; if after all elements of X have been taken, Maker has not achieved this goal, then Breaker wins. In our case, X is the edge set of the complete graph K_n, and both players take turns claiming edges of K_n.

Some examples of Maker-Breaker games on K_n that have been investigated include the *connectivity* game, where Maker wants to build a spanning tree. In the *Hamiltonicity* game, Maker wants to build a Hamiltonian cycle. In the *perfect matching* game, Maker wants to build $\frac{|V(G)|}{2}$ independent edges.

In the unbiased Maker-Breaker game, where Maker and Breaker each make one move alternately, Maker frequently has an advantage. It is therefore natural to ask how quickly Maker can win, i.e. what is the minimum number of edges Maker has to play in order to win. For instance, it is known that Maker can win the connectivity game in $n-1$ moves, which is clearly the best possible [4]. For sufficiently large n, Maker wins the Hamiltonicity game in $n+2$ moves and the perfect matching game in $\frac{n}{2}+1$ moves (see [3]).

Due to Maker's advantage, it is also natural to consider Maker-Breaker games where Breaker makes b moves for every move of Maker. This is referred to as the $1:b$ Maker-Breaker game, where b is the *bias* of the game. The regular $1:1$ Maker-Breaker game is hence referred to as the *unbiased* game. Another question worth asking is that of the critical value b^* such that Maker wins the $1:b$ game for all $b \leq b^*$ but loses for all other b^*.

We considered two ways to make the game harder for Maker: either to require Maker to win fast, or to allow Breaker to have multiple moves. We observe the following trivial but interesting relationship between these two variants: if Maker wins in a $1:b$ Maker-Breaker game, then Maker wins in at most $n^2/(b+1)$ moves in a $1:1$ Maker-Breaker game, simply by pretending Breaker has played $(b-1)$ additional arbitrary moves at every step and using the $1:b$ Maker-Breaker strategy.

Erdos's "probabilistic intuition" for Maker-Breaker games says that the $1:b$ game should be winnable when b is around $1/p^*$ where p^* is the critical probability for one of Maker's structures appearing in an Erdos-Renyi $G(n,p)$ random graph. It turns out this probabilistic intuition is exactly correct when Maker and Breaker flip a biased $1:b$ coin to determine who takes the next move, instead of strictly alternating 1 Maker move to b Breaker moves (see Appendix A). In the case of deterministic Maker-Breaker games this probabilistic intuition is not always correct, but does correctly predict e.g. that $b = \frac{n}{\log n}$ is the critical threshold for connectivity [2].

A strengthened version of "probabilistic intuition" is the idea that strategies for Maker which just play randomly may achieve the optimal threshold value of b, at least up to constants. This was confirmed in a worst-case sense for the case of embedding sparse subgraphs in the paper of Bednarska and Luczak [1].

We suggest the following heuristic extension of the above idea when it comes to the problem of building structures in $1:1$ Maker-Breaker game: if the structure is sufficiently asymmetrical, then we expect that a randomized strategy will be nearly optimal. Furthermore we guess that the optimal speed can be predicted as n^2p^* where p^* is the critical probability in a Erdos-Renyi random graph. (Possibly this should this be replaced by the critical probability for "robust embedding," where a small number of edges may be removed adversarilly from $G(n,p)$ and we still want to embed our particular subgraph). Unfortunately we are far away from being able to make this statement rigorous in any sense (if it is even possible).

On the other hand, we do *not* expect such heuristics to be accurate in the case where the graph has special structure, in particular a large amount of symmetry. In this case we give a surprising result where a $1:1$ Maker-Breaker game can be won quickly and with a very simple analysis, that of building a spanning forest of disjoint triangles. The "random graph" prediction as suggested above is far off: from [6] the threshold value is $p^* = O(n^{-3/4+o(1)})$, so the predicted building speed is $O(n^{5/4})$; in fact, we show that the optimal building speed is exactly $\Theta(\frac{4}{3}n)$ in this case. We present a general framework for building symmetric graphs which we believe most, if not all, fast explicit strategies for Maker-Breaker games fit into; we believe that showing the non-existence of fast strategies in our restricted framework may be a first step towards making the suggested distinction between easy "symmetrical" graphs and difficult "asymmetrical" graphs into a full mathematical theory.

1.2 Results

Consider the following game:

Definition 1.1. Take some positive integers $k \geq 3$, $n \geq 1$ such that k divides n. Consider the Maker-Breaker game on K_n such that Maker wants to create $\frac{n}{k}$ disjoint k-cycles. Let $T_k(n)$ be the number of moves required for Maker to win. What are good estimates for $T_k(n)$?

In this paper we prove the following, for any $\epsilon > 0$:

$$\frac{k+1}{k}n \leq T_k(n) \leq \frac{k+1}{k}n + O(n^{\frac{1}{2}+\epsilon}).$$

In particular, our method for proving the upper bound also extends naturally to general graphs in the following sense. Let $P(G,a)$ be the Maker-Breaker game where Maker wants to build a copy of G in K_a. If Maker wins $P(G,a)$ within m moves, then Maker can build spanning copies of G in K_n within $\frac{m}{|G|}n + o(n)$ moves. Essentially, if Maker can build a single copy of G, then Maker can make it span K_n with the same number of moves per copy of G.

We also extend our lower bound to a more general family of graphs called **flowers**, which are edge-disjoint cycles all sharing a common vertex, so that building a forest of flowers F on K_n requires at least $\frac{|F|+1}{|F|}n$ moves.

2 Building spanning disjoint cycles

2.1 Lower bound

To build a single k-cycle, it is clear that Maker has to use one extra edge, since otherwise at the very end Breaker will take the unique edge Maker needs to complete the cycle. We prove that this property extends to spanning copies of k-cycles, namely that on average, Maker must waste at least one edge for each k-cycle built.

One difficulty in proving the lower bound is that Maker could potentially be building several different k-cycle forests, so that the edge that Breaker takes to prevent a k-cycle may not be unique. In this case, however, Maker wastes edges in building a double threat. To deal with this, we look at maximal families of disjoint k-cycles and prove that each of them must waste sufficiently many edges. In particular, if we consider the k-cycles as vertices in a new graph A, as long as A is connected and contains a cycle, it has it has at least $|A|$ edges, which is exactly what we need. It is therefore sufficient to prove that A can never be a tree. This argument allows us to consider all possible k-cycle forests simultaneously.

Theorem 2.1. $T_k(n) \geq \frac{k+1}{k}n$.

Proof. Consider a connected component S of Maker's $G = K_n$, and consider all the possible maximal ways to make disjoint k-cycles on S. Take one of them. By maximality, there are no other k-cycles in S disjoint from our k-cycles. Call this set of k-cycles A'.

Make a graph A such that A' is its vertex set. An edge connects X and Y in A if there is a path in S connecting some vertex of X to some vertex of Y that does not pass through any other element of A.

Call a vertex v a **whisker** of A if $v \in S$ and there is no path in S between two k-cycles in A' that passes through v. Call a tree without whiskers a **trunk**. Call A a **pre-trunk** if the addition of one edge in S will make it a trunk. See Figure 1. Note that initially all such A are not pre-trunks or trunks since they're empty. I claim the following.

1. There is a unique edge in a pre-trunk that makes it into a trunk.

   To see this, notice that there are no trunks initially, so the edge that makes the pre-trunk a trunk must be part of a disjoint cycle. This is easily seen to be unique.

2. After any move of Maker that creates pre-trunks, Breaker can prevent them from becoming trunks with one move.

   To see this, suppose Breaker takes e and creates pre-trunks. Take one of them - e lies in one of the cycles, call it C. If there is another pre-trunk using e, then it must be part of the exact same cycle C, otherwise it would have a whisker. But then the cycle C is part of both pre-trunks, so they are the same pre-trunk. By the first result, there is a unique edge that, when taken by Breaker, prevents Maker from making any trunks.

It follows that Maker can never have any trunks. At the end of the game, when Maker has $\frac{n}{k}$ k-cycles, consider some connected component S and suppose it has c disjoint k-cycles. Consider the graph A with these cycles as its vertex set. A cannot have a whisker v, since otherwise v must be part of some disjoint k-cycle. So A is not a tree. It follows that there are at least k edges in A, so S contains at least $kc + c = (k+1)c$ edges. Summing over all connected components, Maker's graph G has at least $\frac{k+1}{k}n$ edges, as desired.

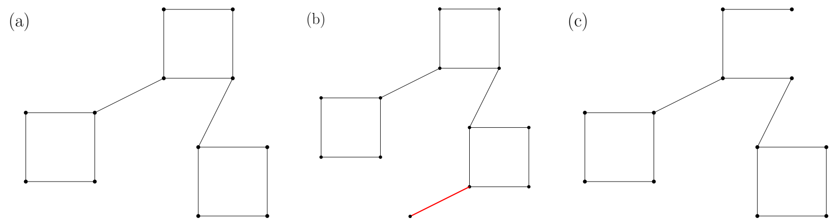

Figure 1: For the case $k = 4$, a trunk, a whisker (in red), and a pre-trunk respectively.

□

2.2 Upper bound

We now show that Maker can win in $\Theta(\frac{k+1}{k}n)$ moves. In [5], Hefetz and Stich proved that for sufficiently large k, Maker wins the Hamiltonicity game on K_k within $k+1$ moves. It follows that by splitting K_n up into $\frac{n}{k}$ sets and playing on each of them separately, Maker can build spanning disjoint k-cycles in exactly $\frac{k+1}{k}n$ moves. For small k, however, our result shows that we can also build spanning k-cycles in the same amount of moves.

As we will show, Maker can build a k-cycle in $k+1$ moves. However, the naive strategy of building a k-cycle whenever possible doesn't obviously work. Consider the situation where we have the final k vertices and Maker has to build a cycle on them. For small k, Maker loses the Hamiltonicity game, so Maker cannot win.

To overcome Breaker's apparent advantage in the end-game, our strategy begins by building sufficiently many, but $o(n)$, copies of K_{2k-1}. Call them C. This is because of the following nice property: if v is an arbitrary vertex outside of some K_{2k-1}, then by claiming any two edges between v and K_{2k-1}, Maker can make two disjoint k-cycles. So now we have a way to resolve any problematic vertices that are created along the way. We then show that Maker can keep making k-cycles outside of the K_{2k-1} until not many vertices are left.

Finally, we want to be able to match the leftover vertices L to C. Since Breaker could have taken edges between L and C, we want to ensure that at the end, there are many unclaimed edges between each element of L and C. To do this, whenever there is a "dangerous" vertex v that has low degree to the elements of C, we immediately match it to some element of C. Since it takes many moves to make a vertex "dangerous", we can show that there are few vertices that can be "dangerous" throughout the entire game. Finally, we are left with a matching problem between L and C, now with the knowledge that they have "high degree" to each other, and then Hall's theorem furnishes a matching that solves our problem.

The bulk of Maker's moves is spent in building k-cycles outside C. There are roughly n/k of these and each of them takes $k+1$ moves, so this takes roughly $\frac{k+1}{k}n$ moves. Since there are $o(n)$ cliques, everything else takes a negligible number of moves to be resolved.

Theorem 2.2. *For any $\epsilon > 0$, for sufficiently large n, we have $T_k(n) \leq \frac{k+1}{k}n + O(n^{\frac{1}{2}+\epsilon})$.*

Proof. We first prove the following Lemma.

Lemma 2.3. *There exists a such that if Maker plays first on an empty a-clique, then Maker can make a k-cycle in $k+1$ moves. Furthermore, if we specify some vertex x in this clique, we can build the k-cycle so that it includes x.*

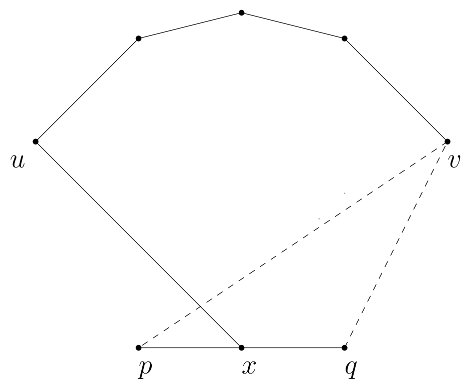

Figure 2: The double threat in Lemma 2.3

Proof. Consider some sufficiently large a. First build a path of length 2, with vertices pxq. Say a vertex is *clean* if Breaker has not taken any edge adjacent to it. Then take an edge uv between two clean vertices. Suppose at step $i = 2, \dots, k-3$ we have a path of length $i-1$ with endpoints u, v after relabelling. Then if Breaker takes some edge adjacent to either u or v, we pick a clean vertex t and take either vt or ut respectively. This guarantees that right after we take t, we have a path of length i with both endpoints clean.

After $k-3$ steps, we have a path of length $k-3$ with clean endpoints u, v. If Breaker does not take any edge involving u or v, then Maker takes ux, and then either vp or vq will create a k-cycle $u \dots vpxu$ or $u \dots vqxu$. Suppose w.l.o.g. that Breaker takes an edge involving u. If it is not up, ux or uq, proceed as above. Otherwise, if Breaker takes ux, Maker takes vx, and then either up or uq wins. If Breaker takes up or uq, Maker takes ux and wins as above.

The total number of steps taken is $2 + k - 3 + 2 = k + 1$. Clearly, if a is large enough then we have sufficient clean vertices for all our operations above, so such an a must exist. □

We begin by building $n^{\frac{1}{2}+\epsilon}$ copies of K_{2k-1} - this can be done within $cn^{\frac{1}{2}+\epsilon}$ moves for some constant c. Let S be the number of unmatched cliques. Define a vertex v to be **dangerous** if for at least $\frac{S}{2}$ of the K_{2k-1}, Breaker has taken an edge between v and the copy of K_{2k-1}. Similarly, say that some K_{2k-1} is dangerous if Breaker has taken at least $\frac{S}{2}$ edges between it and the other vertices. Note that this initial process creates at most $2c$ dangerous vertices and $2c$ dangerous cliques.

Lemma 2.4. *If only $O(n)$ moves have been made, then there can be at most $O(n^{\frac{1}{2}-\epsilon})$ dangerous vertices and cliques, respectively.*

Proof. Suppose there are k dangerous vertices. At worst, the j-th dangerous vertex must be connected by Breaker's edges to at least $\frac{S}{2} \geq \frac{n^{\frac{1}{2}+\epsilon}-j}{2}$ cliques, corresponding to the situation when the previous $j-1$ vertices have been matched. Then we have the inequality

$$\frac{n^{\frac{1}{2}+\epsilon}}{2} + \frac{n^{\frac{1}{2}+\epsilon}-1}{2} + \cdots + \frac{n^{\frac{1}{2}+\epsilon}-k}{2} = O(n)$$

which is easily seen to hold only for k at most $O(n^{\frac{1}{2}-\epsilon})$. □

Let m be the number of vertices that are not in the clique and not part of some disjoint k-cycle.

Lemma 2.5. *As long as $m > n^{\frac{1}{2}+\epsilon}$, we can make a k-cycle among the m vertices.*

Proof. Note that the number of a-cliques among the m remaining vertices is $\binom{m}{a}$. Also, the number of moves played so far outside the K_{2k-1} is at most $\frac{k+1}{k}n + cn^{\frac{1}{2}+\epsilon} \leq Cn$ for some $C > 0$. For each of Breaker's edges, he has destroyed at most $\binom{m}{a-2}$ a-cliques among the m vertices. As long as

$$\binom{m}{a} > Cn\binom{m}{a-2}$$

we can choose an empty a-clique and make a k-cycle in it. Since $m > n^{\frac{1}{2}+\epsilon}$, the above inequality clearly holds. □

And finally we need two more Lemmas.

Lemma 2.6. *Given a bipartite graph of t vertices on either side, if every vertex has degree at least $\frac{t}{2}$, then there is a perfect matching.*

Proof. We use Hall's Theorem. Let the sets be A, B respectively, and consider some subset $S \subset A$. Let N contain the neighbors of S. If $|S| \leq \frac{t}{2}$, then by the degree condition for any $v \in S$ it must have at least $\frac{t}{2} \geq |S|$ neighbors. If $|S| > \frac{t}{2}$, then consider any $v \in B$. It has degree at least $\frac{t}{2}$, and since $\frac{t}{2} + |S| > t$ the neighbors of v cannot be disjoint from S. It follows that v is a neighbor of S, so that in fact $N = B$. Hall's conditions are therefore satisfied. □

Lemma 2.7. *Given a vertex v and a clique K_{2k-1} belonging to Maker, such that Breaker has taken no edge between v and this clique, Maker can make two disjoint k-cycles within 2 moves.*

Proof. Clearly Maker can take two edges between v and this K_{2k-1}, suppose they are va and vb. Then if $a \ldots b$ is some path of length $b - 1$ in the K_{2k-1} then $va \ldots b$ is a k-cycle. The remaining vertices of the K_{2k-1} is the other disjoint k-cycle. □

Now consider the following strategy.

1. Firstly, we resolve the dangerous vertices and cliques created by the initial process of building the K_{2k-1}. There are at most $2c = O(1)$ of each. Match each of the dangerous cliques to a vertex outside of the K_{2k-1}. For each of the dangerous vertices, say v, find an a-clique containing it. Then make a k-cycle containing v. This is possible by Lemma 2.3.

2. While $m > n^{\frac{1}{2}+\epsilon}$, do the following. If there is a dangerous vertex, match it to some clique. If there is a dangerous clique, match it to some vertex. Otherwise, make a k-cycle among the m vertices and decrease m by k. We can do this because of Lemma 2.5.

3. Now that $m = n^{\frac{1}{2}+\epsilon}$, if there are any remaining dangerous cliques or vertices, match them accordingly. By our second Lemma, we definitely have enough cliques and vertices to match the dangerous ones, since we have $O(n^{\frac{1}{2}+\epsilon})$ vertices and cliques, but there are at most $O(n^{\frac{1}{2}-\epsilon})$ dangerous vertices / cliques. For the remaining vertices, find a perfect matching between vertices and cliques. This is possible because there are no more dangerous vertices or cliques, and we can then use Lemma 2.6.

4. Whenever Breaker plays an edge in some vertex / clique pair, Maker plays an edge in the same pair. Otherwise, Maker plays any edge in any pair. It is obvious that eventually every vertex / clique pair will be resolved as stated in Lemma 2.7, and we're done.

□

3 Building a forest of flowers

The lower bound proved in the previous section relies on the fact that to build a single k-cycle, we had to waste a single edge, along with some uniqueness properties. Consider the following family of graphs:

Definition 3.1. A **flower** is a graph with a specified vertex v and n edge-disjoint cycles of posisbly varying lengths, all starting and ending at v.

Theorem 3.2. *Maker can build a spanning forest of a flower F on K_n between $\frac{|F|+1}{|F|}n$ and $\frac{|F|+1}{|F|}n + O(n^{\frac{1}{2}+\epsilon})$ moves.*

The proof of the lower bound relies on the following fact:

Lemma 3.3. *Given some flower F, there exists a such that Maker can build F on K_a with only one wasted edge.*

Proof. Suppose the n cycles are $C_1, C_2, \ldots, C_n$. We build the cycles in order. Firstly, by Lemma 2.3 we can build C_1 with one wasted edge. Suppose the wasted edge is connected to vertex x of C_1, say it is xq. Then as in the proof of Lemma 2.3, take another clean vertex p and take px. Then we have a path of length 2 with vertices pxq, and then we can build C_2. Continuing this way, we build $C_1, \ldots, C_n$ with one wasted edge in the end. □

The proof of Theorem 3.2 follows in exactly the same way as Theorem 2.1, since both the uniqueness properties still hold.

Remark 3.4. Given a finite list of positive integers $(k_1, \ldots, k_r)$, we can build cycles of length $k_1, \ldots, k_r$ in K_n that span the entire graph. Let $b = \prod_{i=1}^{r} k_i - 1$, and build sufficiently many copies of K_b. Then we can resolve a vertex to K_b into cycles of length k_j for any j, since k_j divides $b+1$. This implies that we can build the mixture of cycles in any desired proportion.

4 Building spanning copies of general graphs

Recall from the introduction that $P(G, a)$ is the Maker-Breaker game where Maker wants to build a copy of G in K_a. The following result shows that if it takes m moves to win the game for a single G, then Maker can also build spanning copies of G with the same number of moves per copy.

Theorem 4.1. *For any $\epsilon > 0$, if Maker wins $P(G, a)$ within m moves, then Maker can build spanning copies of G in K_n within $\frac{m}{|G|}n + O(n^{\frac{1}{2}+\epsilon})$ moves.*

Proof. We use the same strategy as that of Theorem 2.2 and show that the calculations still hold.

1. Pick b sufficiently large such that we can resolve a vertex by matching it with a K_b. One example of such a b is as follows: take some vertex of G and suppose it has degree d in G. Then pick b such that $b \geq 2d$ and $b+1$ is divisible by $|G|$. Build $n^{\frac{1}{2}+\epsilon}$ copies of K_b.

2. Resolve the dangerous vertices and cliques created in Step 1.

3. While $m > n^{\frac{1}{2}+\epsilon}$, resolve dangerous vertices and cliques if possible, and otherwise pick an empty K_a and build G on it in m moves. Note that at this point we have made at most $O(m\frac{n}{|G|}) = O(n)$

moves. If there are p dangerous vertices or cliques that we encounter, then by the same reasoning as in Lemma 2.4, p satisfies

$$\frac{n^{\frac{1}{2}+\epsilon}}{2} + \frac{n^{\frac{1}{2}+\epsilon} - 1}{2} + \cdots + \frac{n^{\frac{1}{2}+\epsilon} - p}{2} = O(n)$$

which shows that $p = O(n^{\frac{1}{2}-\epsilon})$. It follows that for sufficiently large n, we always have enough cliques to match dangerous vertices to and vice versa.

4. When $m = n^{\frac{1}{2}+\epsilon}$, find a perfect matching between the remaining vertices and cliques using Lemma 2.6.

5. Whenever Breaker plays an edge in some vertex / clique, pair, Maker plays an edge in the same pair. Suppose this vertex is v and we have taken d edges between v and K_b, then by taking v, its d neighbors and $|G| - d - 1$ other arbitrary vertices in the K_b, we make a copy of G. Since $b + 1$ is divisible by $|G|$, the remaining vertices in the K_b can be split into copies of G.

The total number of moves used in Step 3 is $\frac{m}{|G|}(n - n^{\frac{1}{2}+\epsilon} - O(n^{\frac{1}{2}+\epsilon}))$. We use $O(n^{\frac{1}{2}+\epsilon})$ moves in Steps 1, 2 and 5. The result follows. □

Theorem 4.2. *For any positive integer $k \geq 3$ and for sufficiently large n divisible by k, Maker can build spanning k-cliques on K_n.*

Proof. From the previous results, there is a such that Maker can build a b-clique on an empty a-clique. Suppose this takes c moves. Then the total number of moves made using these moves is at most $\frac{n}{b}c = O(n)$.

We first begin by making a certain number, say $n^{\frac{1}{2}+\epsilon}$ $3b - 1$-cliques. Clearly, given a vertex not connected to one of these cliques, we can resolve the vertex to make 3 b-cliques. We then resolve the dangerous vertices created by this process, then keep on picking empty a-cliques to make b-cliques. This continues until $m = O(n^{\frac{1}{2}+\epsilon})$. Then we take the remaining vertices and match them to the b-cliques. □

Corollary 4.3. *For any graph G, for sufficiently large n divisible by $|G|$, Maker can build spanning copies of G on K_n.*

Remark 4.4. Analogously to Remark 3.4, given a finite list of graphs $G_1, \ldots, G_r$, we can also build copies of G_i that span K_n.

5 Towards building almost-spanning graphs of bounded degree

Consider a graph G such that $|G| = (1 - \epsilon)n$ for some $\epsilon > 0$ and such that it has maximum degree Δ. How many moves does Maker require to build a copy of G in K_n?

We have the following strategy for Maker. Pick some $p > 0$ and let b be roughly equal to $\frac{1}{p}$. In the $1 : 1$ game, in addition to Breaker's move, Maker pretends that $b-1$ other random edges have also been taken by Breaker. On Maker's move, he picks a random edge with the same probability p. The idea is that by doing this, Maker ends up with exactly $G(n, p)$ minus the edges chosen by Breaker. We end up with a "robust" $G(n, p)$ model, where we want to prove that with positive probability, G exists in this "random" graph.

For one of Breaker's edges, the probability that it is in Maker's $G(n, p)$ is p. Since both players have n^2p edges, the expected number of Breaker's edges that are in Maker's graph is n^2p^2. We would therefore like to prove that G exists almost surely in $G(n, p)$ if n^2p^2 edges are taken adversarially.

Suppose $p = cn^{-\frac{1}{\Delta}} \log^{-\frac{1}{\Delta}} n$. Then it is easy to see that $G(n,p)$ must have a copy of G, as follows. Label the vertices of G as $v_1, \ldots, v_t$ (where $t = (1-\epsilon)n$) and suppose we have constructed $v_1, \ldots, v_j$. Since $|G| = (1-\epsilon)n$, there are at least ϵn vertices that are not among the v_i. Let A be these vertices. Suppose v_{j+1} is connected to some subset F of $\{v_1, \ldots, v_j\}$. Then the probability that some vertex $a \in A$ is connected to all of F is p^Δ. So the probability that none of the vertices in A are connected to all of F is $(1-p^\Delta)^{\epsilon n} \leq e^{-p^\Delta \epsilon n}$. This is the probability that we cannot choose a suitable v_{j+1}.

Now we want this process to continue all through to v_t. By the union bound, the probability that we fail at some v_j is at most

$$(1-\epsilon)ne^{-p^\Delta \epsilon n} = (1-\epsilon)e^{-c\epsilon \log n} = (1-\epsilon)n^{1-c\epsilon} < 1$$

The last inequality holds for c sufficiently large such that $1 - c\epsilon < 0$. It follows that for such this p, $G(n,p)$ has a copy of G with high probability.

We have not been able to extend this strategy to prove that G must appear in $G(n,p)$ when Breaker can take away edges from it. There are a few simplifications that we can make, however. Note that on average, Breaker takes np^2 edges from each vertex. By Markov's inequality, if k is such that $\frac{1}{k} < \epsilon$, and X is the number of edges Breaker takes from some vertex, then

$$P(X \geq k\mathbb{E}[X]) \leq \frac{1}{k} < \epsilon$$

It follows that we can simply drop the vertices where Breaker has taken more than $k\mathbb{E}[X] = knp^2$ edges, and work with a smaller value of ϵ instead.

6 Explicit strategies for Maker via symmetry

Consider the problem of building a single copy of some graph G. If G is "sufficiently asymmetric", there should be a unique edge missing from G at the very end of the process, which Breaker can take and prevent Maker from making this particular copy of G. Essentially, the intuition is that for graphs G with great symmetry, such as cycles, it should be possible to find an explicit strategy that builds G. For graphs which are asymmetric, however, we expect the random strategy to work better.

Specifically, graphs such as the k-cycle and Hamiltonian path satisfy the following condition that allows us to state an explicit strategy for Maker such as in Lemma 2.3.

Lemma 6.1. *If there is a sequence of graphs H_i, J_i, $i = 0, \ldots, r$ satisfying the following:*

1. *H_0 is the empty graph on n vertices, and there are edges $m_i \in J_{i-1}$ such that H_{i+1} is isomorphic to $H_i \cup m_i$ for all i.*
2. *For any edge e in J_i, there exists an automorphism ϕ of J_i which fixes H_i and such that $\phi(e)$ is not in $H_{i+1} \cup J_{i+1}$.*
3. *$J_{i+1} \subset J_i$ for all i.*

then Maker can build H_r in any graph containing J_0.

Proof. At each step, Maker plays m_i under some isomorphism of K_n. We show that Maker can always do this while ensuring that no edge in J_i has been taken by either player. We say that J_i is free.

Suppose Maker has built H_i and no edge in J_i has been taken by any player. If Breaker's edge is not in J_{i+1}, just take m_i and we're done. Suppose Breaker takes some edge e in J_{i+1}. By the second condition, there is an automorphism ϕ that sends H_i to $\phi(H_i)$ such that $\phi(e)$ is not in $H_{i+1} \cup J_{i+1}$. Since $J_{i+1} \subset J_i$ and J_i is free, J_{i+1} must be free. Now Maker just takes $\phi(m_i)$ to make H_{i+1}, as desired. Continuing in this manner, we get to H_r and we're done. $\square$

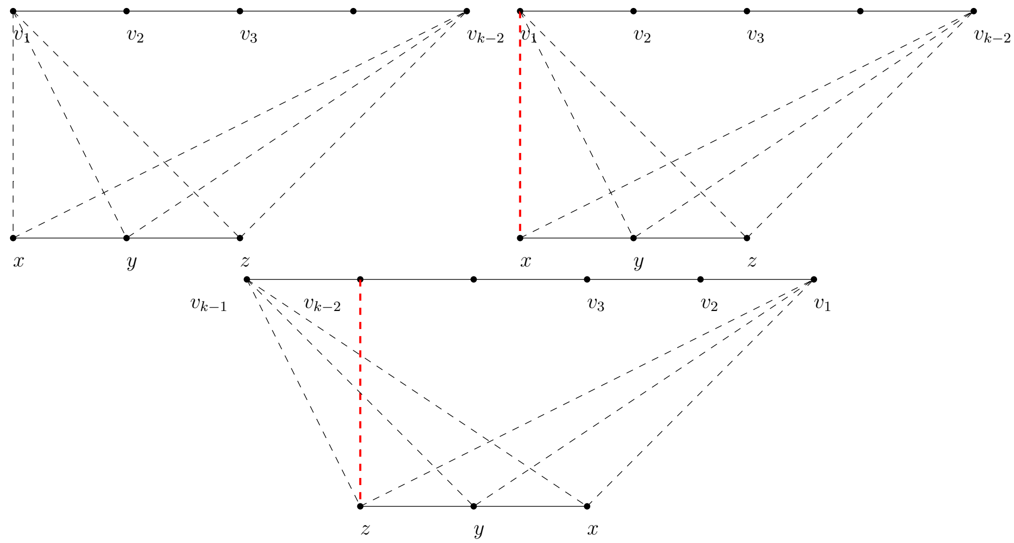

Figure 3: The symmetry of the H_i allows us to disregard Breaker's move

In the case of a k-cycle, the explicit strategy can be stated in the language above.

Remark 6.2. Let A_v denote the set of edges with v as one of its endpoints. Then for $k > 3$, H_k consists of xy, yz and a path $v_1, \ldots, v_{k-2}$. J_k consists of all edges between x, y, z and v_1, v_{k-2}, along with A_t for many other vertices t. Intuitively this means that we want the endpoints of the path v_i to be disconnected from the path xyz, as well as there being many other clean vertices (not connected to any edge taken by either player).

This way, by flipping the two paths in some H_k, we get an isomorphism that sends H_k to H'_k that preserves the paths. If Breaker takes some edge in J_{k+1}, then the corresponding edge in J'_{k+1} is not taken, and we just "extend in the other direction". This is illustrated in Figure 3.

7 Acknowledgements

I would like to thank Asaf Ferber for introducing us to this problem, for extremely valuable insights into Maker-Breaker games, and for suggesting the arguments in Section 5. I am also very grateful to Professors Ankur Moitra and David Jerison for their guidance and advice throughout the program. Finally, I would like to thank my mentor Frederic Koehler for his infinite patience and extensive mentoring; without his guidance this paper would never have been written.

A Random-turn Maker-Breaker games and $G(n, p)$

Consider the Maker-Breaker game on a graph property P where instead of taking turns alternately, Maker and Breaker flip a p-biased coin to decide who goes next. Peres, Schramm, Sheffield, and Wilson [7] showed that the probability of Maker's win in this case is exactly the probability that the desired property P will appear in $G(n, p)$.

To prove this result, first apply the method of conditional expectation to observe that there exists an optimal deterministic strategy for both players; then if Breaker plays the exact same strategy as Maker, we see that each edge is present in the final output with probability p, and considering a similar strategy-stealing argument for Maker proves the result.

B Explicit strategies to build certain graphs

Lemma B.1. *Consider the following graph that looks like a "ladder": take two paths of length k and match the corresponding vertices. Maker can build this graph with $k-2$ wasted edges.*

Proof. We start with a single 4-cycle, which we can build with 1 wasted edge. Suppose at some point Maker has two paths of length j, $a_1, \ldots, a_j$, and $b_1, \ldots, b_j$ such that $a_i b_i$ all belong to Maker. Build a path of length 2 pxq such that p, q are not connected to a_j, b_j. Then as in Lemma 2.3, Maker can resolve this situation to make a 4-cycle with $a_j b_j$ as an edge, thus extending the path by 1. By induction, we can build the entire length k and we're done.

To save the two wasted edges, we use the 1 edge from the first cycle. Suppose the edge connects the cycle to some vertex v; then take vw for some arbitrary w to make a double threat from w to the cycle. □

The original question was to build two k-cycles quickly with the obvious matching.

References

[1] Małgorzata Bednarska and Tomasz Łuczak. Biased positional games for which random strategies are nearly optimal. *Combinatorica*, 20(4):477–488, Apr 2000.

[2] Heidi Gebauer and Tibor Szabó. Asymptotic random graph intuition for the biased connectivity game. *Random Structures & Algorithms*, 35(4):431–443, 2009.

[3] Dan Hefetz, Michael Krivelevich, Miloš Stojaković, and Tibor Szabó. Fast winning strategies in maker–breaker games. *Journal of Combinatorial Theory, Series B*, 99(1):39 – 47, 2009.

[4] Dan Hefetz, Mirjana Rakić, and Miloš Stojaković. Doubly biased maker-breaker connectivity game. *arXiv preprint arXiv:1101.3932*, 2011.

[5] Dan Hefetz and Sebastian Stich. On two problems regarding the hamiltonian cycle game. *the electronic journal of combinatorics*, 16(1):R28, 2009.

[6] Anders Johansson, Jeff Kahn, and Van Vu. Factors in random graphs. *Random Structures & Algorithms*, 33(1):1–28, 2008.

[7] Yuval Peres, Oded Schramm, Scott Sheffield, and David B Wilson. Random-turn hex and other selection games. *American Mathematical Monthly*, 114(5):373–387, 2007.